Chapter 1 – Introduction

The recent explosion in computer usage is transforming every aspect of our society. The rapidly increasing presence of computer technology in the workplace, the home and the school is almost certain to have impacts on the ways in which we learn. In education, the importance placed on using computers in schools is evident from governmental policy and the consequent levels of funding being deployed. For instance, in its latest policy document, the UK government has proposed that, by 2005-6, there should be an annual funding of £700 million on computer-based technology for the education sector [DFE03]. The same policy document states that the number of computers in schools has virtually doubled in the four years from 1998-2002 – an indication of the amount of money that has already been spent on computers for education. This dramatic increase in computer technology in schools is not limited to the UK. In the USA, for instance, funding for computers in schools in the year 2000 was estimated to be in the region of $8 billion [AFC00].

Governmental reports paint a rosy picture of the positive influence of computers on students and the commensurate increases in achievement that the use of computers effects. For instance, in the UK, the government claims that research shows that:

> '[Information and Communication Technology] can have a direct positive relationship to pupil performance – equivalent in some subjects to half a GCSE grade.' [DFE03]

Academic researchers are more ambivalent about whether computers enhance the quality or standard of learning. There are those who champion the use of computers by children (see e.g. [MO94, Tap98, Ben99, BR99]). However, a significant number of researchers claim that there is no positive impact on standards in children using computers, and further that the use of computers may indeed be harmful to their educational development (see e.g. [Tur95, Kay96, Opp97, KC98, Hea99, AFC00, Opp03]).

The increased interest in using 'computers for learning', and the conflicting viewpoints emerging from pedagogical research, give rise to questions that need to be addressed in order to guide future practical developments. To understand how we can use computers fruitfully in learning for both the construction and use of models, we must recognise what learning we wish to support on computers. This thesis addresses this theme with reference to two main categories of computer use for learning: the construction of models by children, and the construction of models by professional adults. The broader context for this research is conveniently framed using a similar bipartite classification in the agenda of the Human-Centric Computing symposium of 2001 [HCC01]. In respect of children realising domain learning through programming:

> 'The first part of the Symposium will focus on educational issues and end-user programming for beginners. How can kids build their own games? How can education be enriched with computational literacy allowing people to express complex ideas with interactive media? What kinds of programming approaches are particularly well suited to computer users with no programming background? What are the cognitive road blocks in programming for beginners?'

In respect of professional adults realising domain learning through programming:

> 'The second part of the Symposium will focus on professional users. How can users gain more control over their high functionality applications, such as word processors, browsers, and spreadsheets, through end-user programming? How well do end-user programming languages scale? How can end-user programming be integrated into high functionality applications? How can the reuse of end-user programs be stimulated? What are the trade offs between domain-specific and generic end-user programming languages?'

This thesis is concerned with investigating an alternative approach to providing computer support for learning that bears on many of the questions raised in the above agenda.

1.1 Construction of computer models by children

In 1980, Seymour Papert published a seminal work on computers for children, called 'Mindstorms' [Pap80]. He described a vision for how children could 'unleash the power of computer programming' to aid their learning in a personally meaningful domain. 'Mindstorms' contains many anecdotes of practical programming undertaken by children together with comments on the perceived effects it had on them. Papert's vehicle for practical programming was the computer language Logo. Logo is a simple procedural programming language originally targeted at the domain of geometry. It has a small number of commands that can be combined to produce complex geometrical patterns. In early versions of Logo, the language could be used to control a floor device called a turtle. There have been many subsequent programming languages inspired by Logo and Papert's vision. Development environments such as Microworlds Project Builder [LCS03] and Imagine [KB00] are examples of Logo-like languages in a more general-purpose context. Logo also spawned research languages aimed at specific contexts, such as massively parallel Logo (as represented in *Logo [Res94] and NetLogo [Wil99]) and Logo interfaced to physical devices such as vehicles and robots [ROP88].

In some respects, Logo has been a spectacular success, but in others a failure. This is apparent when we contrast Papert's visions for Logo with its practical applications in education. Papert saw Logo as pioneering a new wave of technology aimed at children that would liberate the child from the 'oppressive nature of school-based instruction' [Pap80]. In his vision, children would be able to use computer technology in a free way to learn personally important subjects at a pace and style that suited them. His 'Mindstorms' book was targeted at teachers and imparted his message that teaching would become a radically different profession under the influence of computer technology. Logo has been successful in terms of its widespread use – programming a floor turtle is mentioned in the United Kingdom National Curriculum [UNC03] as a way of teaching aspects of geometry. Due to this, it is almost certainly the programming language that has had the greatest exposure in UK classrooms.

However, Papert would not necessarily interpret this wide adoption of Logo as an indication of success. His second book, 'The Children's Machine', published in 1993, was already expressing concern about the way in which Logo was being used. For him, the classroom use of Logo – and computer technology in general – was moving away from his vision to reinforce traditional school-based instruction [Pap93].

Papert's design rationale for Logo was that of taking what he perceived as the best ideas from computer science and 'child engineering' them (as reported by Kahn [Kah01] in notes from a Logo project meeting in 1977). There have been a number of attempts at providing programming languages that are either designed for children, or are accessible to children. They include:

i) Boxer – a programming language that was spawned from the Logo project [diSA86]. It is a general purpose language that adopted a mixture of programming paradigms.

ii) Agentsheets – a rule-based visual programming language and development environment that is related to spreadsheets (see section 2.3.3) [Rep93].

iii) Toontalk – an animation-based programming language in which programs are created by manipulating animated tools and training robots by demonstration [Too03].

Each of these products takes its inspiration from a particular programming paradigm and attempts to develop a simplified programming language in this idiom. As Kahn mentions in [Kah01], these more recent languages are again attempting to 'child engineer' what are seen by their proponents to be the best ideas of computer science. The motivation for the educational use of the programming languages listed above is a belief that children constructing their own programs will concurrently be engaged in meaningful domain learning. In this thesis, we shall argue that the support that program construction can offer to domain learning is heavily influenced by the choice of programming paradigm – a theme we shall return to in chapter 4.

1.2 Construction of computer models by adults

In 1993, Bonnie Nardi wrote a seminal book on end-user programming called 'A small matter of programming' [Nar93]. Her investigations were centred on how non-programmers can utilise computers to create their own models through programming, or more generally through model construction. Nardi's basic argument [Nar93] is that:

> 'the problem with programming is not programming; it is the languages in which people are asked to program'.

Her research suggested that end-users can build their own models if the conditions are right. She highlighted several successful end-user languages:

i) Logo – widely used in the domain of geometry by a large number of children.

ii) Mathematica – a general software system for mathematical applications [GG00].

iii) Computer-Aided Design languages – used extensively in architecture and product design [Dug99].

iv) Spreadsheets – a ubiquitous desktop application used in many domains by an enormous number of end-users, ranging from children to accountants.

Nardi noted that these languages had been successful in part because of their domain specific primitives and in part because their design characteristics made it easier for non-programmers to learn through model building. She observed that spreadsheet development was a particularly successful end-user programming environment because [Nar93]:

> 'Managing dependency relationships is a particularly good example of the way in which an end user programming system can allow users to focus on their domain-related problems at a very high level'.

Spreadsheets have been successful in the end-user programming domain because, in specific learning domains, there is a close relationship between constructing a model and learning about the domain. In the light of this observation, we use spreadsheets as our starting point for investigating 'computers for learning' (see chapter 2) and go on to connect spreadsheets with an approach to computer-based model construction that we have developed – called *Empirical Modelling*.

1.3 Introducing Empirical Modelling

In this section, we give a brief overview of the fundamental concepts of the Empirical Modelling (EM) approach. Dr Meurig Beynon initiated EM research in 1983 at the University of Warwick with the design of the definitive notation ARCA [Bey83]. Over the subsequent 20 years, the project has encompassed a wide range of interests including definitive notations [Yun90, Yun93, Run02], geometry [Car94, Car99], computer graphics [ABC+98], business [BRR00, CRB00, RRR00], artificial intelligence [Bey99] and educational technology [Bey97].

Empirical Modelling is an approach to constructing models – typically computer-based – that can assist our understanding of a phenomenon. The term 'empirical' is used to reflect the emphasis on experiment, observation and interaction during the construction of a model. In contrast to typical computer models – which are formal mathematical models – the development of an EM model more closely resembles the development and use of a spreadsheet than that of a traditional computer program, in that the model is incrementally constructed through interaction with a partially completed model.

In EM, the primary emphasis is on modelling state-as-experienced rather than behaviour-as-abstracted, as respectively represented by spreadsheets and traditional computer programs. Whereas the construction of a traditional program relies upon the prior specification of its abstract behaviour, the development of a spreadsheet model evolves through the representation of state as currently perceived. The crucial distinction between spreadsheet development and conventional programming is that there is no circumscription of the possible future states that the system may enter. The identification of sensible behaviours is found through experimentation and interaction with the spreadsheet. An EM model exhibits similar qualities to a spreadsheet, in that the modeller has complete discretion over the interactions they perform.

Within the framework of observation-oriented state-based modelling, EM identifies three key concepts: observables, dependency and agency. Each of these concepts has a part to play in understanding and exploring a phenomenon. An observable is a perceived element of the state to which we can ascribe a value. Dependencies are indivisible relationships that exist between observables. Agency is concerned with attribution and realisation of state change.

In this thesis, we discuss and illustrate how the principles of EM can enable computer-based models to be constructed in a way that is intimately linked with domain learning. We shall discuss the concepts and orientation of EM in much greater depth in chapter 3.

1.4 Motivations for the thesis

The research in this thesis exploring connections between EM and learning is motivated by research from fields such as experimental science, education and psychology that has direct relevance to our exploration of computer-based modelling as a learning aid. Here we outline key ideas that have guided this thesis and which are discussed in detail in later chapters:

- **Constructionism**. In [Pap80, Pap93], Papert describes how children can use computers as exploratory tools to further their own private active learning. His theory of constructionism asserts that learning is most beneficial when learners are actively building their own knowledge structures in a domain of personal interest. In this thesis, we explore how

EM as a modelling approach can enable learners to construct computer-based models in a constructionist framework.

- **End-user programming**. In [Nar93], Nardi describes how successful end-user programming environments, such as spreadsheets, have allowed non-specialists to harness the power of the computational medium to construct their own artefacts to help them in solving their personal tasks. In this thesis, we explain the potential of EM as a way in which computer users can construct artefacts that builds on the principles embodied in spreadsheets.
- **Open development**. In [Brö95], Brödner describes two cultures in engineering that he calls 'closed-world' and 'open-development'. In a closed-world approach, the assumption is that all properties and relationships between objects can be stated as objectified, explicit, propositional knowledge. An open-development approach does not contest our ability to form objectified, explicit, propositional knowledge, but assumes the primary existence of practical experience that has been gained through an individual's interaction in the world. In this thesis, we explore how EM can be viewed as an open-development approach that emphasises the primacy of personal experience in constructing computer-based models that embody our emerging understanding of a phenomena.
- **Construals**. In [Goo90], Gooding describes how the physicist Michael Faraday constructed physical artefacts to enable him to understand electromagnetic phenomena. These concrete artefacts are termed 'construals' by Gooding. In this thesis, we argue that EM models should be considered as construals rather than preconceived programs and discuss the impact of this change in interpretation with reference to learning.
- **Bricolage**. In [Lev68], Levi-Strauss describes the idea of bricolage. Bricolage refers to an approach to construction that is hands-on, negotiational, exploratory, interactive and experimental. In domains where knowledge is provisional, a bricolage approach allows learners to construct physical models concurrently with their emerging understanding

of that domain. In this thesis, we explore how EM can be used as a computer-based modelling approach that embodies bricolage and consider the resulting benefits from a learning perspective.

- **Situated learning**. In [Lav88], Lave emphasises the need for learning to be situated in realistic situations, especially when the subject matter concerns human interaction or is hard to grasp in abstract terms. In this thesis, we explore how EM can act as a situated modelling approach (cf. Suchman [Suc87], Goguen [Gog96]) that allows personal viewpoints and conflicting interests to be represented.

Constructionism, originally introduced by Papert, was primarily concerned with children learning through writing computer programs. End-user programming is more generally concerned with domain learning through the construction of computer-based models. In respect of both these agendas, Brödner draws attention to the significance of practical experience explored through open development as a complement to propositional knowledge of the closed world. Gooding considers the role that the construction of artefacts can play in embodying our understanding. Brödner and Gooding's views endorse our use of artefacts in support of experiential learning activities. Levi-Strauss and Lave describe the key qualities that modelling approaches require to support the open-development approach to constructing artefacts.

Previous research has considered the relationship between EM and conventional methods in different application domains. Two common themes in this comparative research are: the emphasis placed on the human element in the modelling process; and the distinctive characteristics of the EM approach when compared to conventional programming and software development. Generally, these advantages are concerned with the knowledge that can be gained more easily using EM than with conventional methods in the particular application domain. However, none of them has considered in detail why this should be possible from a learning perspective.

Broadly speaking research has been directed at two main areas: software system development and business applications.

With reference to software system development, Paul Ness [Nes97], Patrick Sun [Sun99] and Allan Wong [Won03] have all compared development in EM with conventional software system development. Ness was concerned with identifying how computer technology could offer support for the construction of *creative artefacts*, which promote exploratory representation of unfamiliar subjects, rather than *analytical artefacts*, which promote methodological representation of familiar subjects. He argued that conventional software development focuses on analytical artefacts, whereas EM focuses on creative artefacts. Ness argued that the support for creative model building in EM stems from properties of creative artefacts that are not to be found in analytical artefacts. These properties, namely novelty, ambiguity, implicit meaningfulness, emergence, incongruity and divergence were identified in the work of Finke [FWS92]. These properties are ideally suited for supporting exploratory model building for learning, where there are misunderstandings, inconsistencies and digressions. Sun was concerned with identifying how computer technology could support distributed modelling. He argued the case for EM as an amethodical approach to software development for distributed applications in which knowledge was 'cultivated' through situated modelling. Situated modelling for learning is a prominent theme in this thesis. Wong identified EM as supplying 'a suitable setting for both the cognitive and collaborative aspects of system development in which the emphasis is on heuristic human problem solving and maintaining conceptual integrity in a system design' [Won03]. These characteristics of EM are significant in respect of learning because they relate to the negotiation of meaning in both the private and the public domains. Wong also considered how EM can be used to construct environments in which the user takes responsibility for the circumscription and customisation of a system. This vision is well matched to the needs of a teacher who needs to develop and customise resources to suit their educational context.

In terms of applications to business, Suwanna Rasmequan [Ras01], Soha Maad [Maa02] and Yih-Chang Chen [Che01] have compared an EM approach with currently used systems in various business areas. Rasmequan was concerned with the integration of human cognitive processes and computing processes in business software development. She argued that EM supports cognitive processes by promoting rich representations of situations that offer direct experience, encourage active engagement and spontaneous involvement. Maad was concerned with identifying a framework within which to conduct software system development in the domain of finance. She argued that EM offers support for an alternative culture in finance through an approach to software development that integrates experiential and situated aspects of finance together with close human involvement. Chen was concerned with how EM could be viewed as an approach to Business Process Re-engineering (BPR) where businesses revise their practices to adapt to new computer technology. He argued that the failures of BPR systems were attributable to the inability to preconceive or predict all the causalities when modelling real-world situations with high levels of human involvement. He argued that these problems could be alleviated by using EM for requirements engineering to gather knowledge prior to the construction of a business software system. All three of these authors stress the role that EM can play in acquiring and applying domain understanding throughout the development of a business system. This suggests that EM is well suited to active knowledge construction in many different domains.

This thesis establishes a connection between EM and learning that accounts for its fitness for active knowledge construction, and that can also offer a unifying perspective on the previous work described above. The potential of EM in relation to learning was first outlined by Beynon in his 1997 paper entitled 'Empirical Modelling for Educational Technology' [Bey97]. In the paper, Beynon outlined issues for technology in education from the perspectives of IT management, teachers and pupils. The relationship between EM and learning is at the core of the EM for the educational technology agenda. Beynon framed his discussion with reference to a perspective on learning that he termed the 'Empiricist Perspective on Learning'. This perspective highlights the role that the experiential activities that inform pre-articulate understanding play in learning.

Beynon's paper did not discuss the Empiricist Perspective on Learning with reference to any received learning theories such as constructionism and situated learning. In 1999, I undertook a taught Masters project investigating the potential of EM for educational technology [Roe99]. This preliminary study comprised a limited literature review that concentrated on the work of Seymour Papert [Pap80, Pap93] on LOGO and the theory of constructionism. The limited scope of my study did not enable a full investigation of the potential of EM with respect to educational technology. For instance, it made no reference to Beynon's Empiricist Perspective on Learning or to other learning theories.

The past work on EM and educational technology, my MSc thesis, and the ideas of other authors outlined in this section have together motivated the research in this thesis.

1.5 Research Contributions

The major contention of this thesis is that computer-based model building, as generally practised, does not give adequate support for the experiential aspects of pre-articulate learning. We believe that the principles and practice of Empirical Modelling offer fuller support for pre-articulate learning in respect of both model building and model use. With regard to the construction of models, this thesis develops the 'Empiricist Perspective on Learning' introduced by Beynon in 1997 [Bey97]. Beynon's perspective has been revised in the light of the research in this thesis, and is now referred to as an 'Experiential Framework for Learning' (EFL). This thesis contains the first comprehensive account of the relationships between EM, the EFL and the use of computers for learning. This research has involved a careful investigation, synthesis and analysis of ideas from many learning theories. Relating learning theories and EM has required studying the literature from both fields to identify common ground. With regard to the use of EM models in an educational context, this thesis also gives a full account of principles and techniques that allow the creation of flexible and extensible learning environments.

In the course of this research, I have been the primary author of two papers [RB02, RBF00], and contributed to several other publications [BCH+01, BRW+01, EJR+01, RRR00, BBC+01] that relate to the ideas presented in this thesis. A central claim of the thesis – that EM better supports pre-articulate learning and its integration with formal learning – is discussed in [RB02] and developed in chapter 4. The digital watch artefact described in section 4.7 of the thesis was discussed in [RBF00] in connection with engineering education. The model has also been used as a case study for investigating cognitive aspects of user-artefact interaction [BRW+01] and was also demonstrated at a workshop on cognitive dimensions (see e.g. [Gre89]) to which I contributed (see [BBC+01]). The extent to which EM allows the computer to be used as an instrument rather than a tool is discussed in [BCH+01]. The restaurant case study, discussed in chapter 3, featured as an illustrative example in a paper on Strategic Decision Support Systems [RRR00] and was also adopted as a case study by Rasmequan [Ras02]. The application of EM principles to simulate LEGO Mindstorms robots (developed in conjunction with the researchers at a children's technology club in Finland) was discussed in [EJR+01]. This is described in section 6.4 of the thesis.

For this thesis, I have also constructed several example models of differing levels of sophistication. These include:

i) the spreadsheet model [EMRep, spreadsheetRoe2002], discussed in section 2.5.

ii) the restaurant manager model [EMRep, restaurantRoe2000], discussed in section 3.5.

iii) the digital watch model [EMRep, digitalwatchRoe2001], discussed in section 4.7.

iv) the variations on the OXO model, discussed in section 5.3.3.

v) the clown-and-maze model [EMRep, krustyRoe2002], discussed in section 5.4.2.

vi) the relational algebra tutor [EMRep, ratRoe2003], discussed in section 5.4.3.

vii) the robotic simulation environment [EMRep, rseRoe2003], discussed in section 6.4.

1.6 Contents of the thesis

This thesis is organised into seven chapters, of which this is the introductory one.

In Chapter 2, we consider the potential of exploratory modelling for learning. We first consider the spreadsheet as a tool for exploratory modelling. Three considerations motivate our choice of the spreadsheet as a starting point:

- it is a popular programming paradigm for the end-user.
- it is widely used in education for creating models and exploring phenomena through the use of 'what-if?' style queries.
- it is a ubiquitous application.

We use the spreadsheet concept to draw out two key aspects of exploratory modelling: namely, the *negotiation* and *elaboration* of the relationship between a computer model and its referent (designated as 'the semantic relation β' by Cantwell-Smith [Smi97]). We will argue that spreadsheets are well suited to negotiation but are limited in respect of elaboration. We identify research products based on spreadsheet ideas that have attempted to overcome some of the limitations of spreadsheets and consider their qualities in respect of negotiation and elaboration. In the final part of the chapter, we introduce practical Empirical Modelling and conclude that it offers support for negotiating and elaborating the semantic relation.

In Chapter 3, we outline the major challenges for the use of computers for learning. These challenges are concerned with bridging the gap between how the computer scientist and the educationalist view the use of computers for learning. A computer scientist is primarily interested in issues of usability, requirements specification and the choice of programming paradigm. The educationalist focuses on the qualities of the learning that is taking place, the actual computer implementation being of only secondary concern. Marrying these two viewpoints requires an approach to computer-based model construction which is such that:

- there is a close connection between domain learning and model construction.
- educational software that is developed can be easily and flexibly adapted in response to different learning situations and competencies.

With reference to many examples of learning activities, we describe a view of learning that presumes that knowledge is rooted in our personal experience. These examples motivate our 'Experiential Framework for Learning' (EFL). The EFL comprises many learning activities ranging from the private to the public, from the empirical to the theoretical, and from the concrete to the abstract. We introduce the key principles of EM: the development of construals; the primary emphasis on representing state-as-experienced; and the concepts of observable, dependency and agency. We argue that EM model construction that is based on these principles respects the relationships between activities in the EFL, so that EM can support the integration of the empirical and the theoretical within a single modelling environment. We illustrate these ideas with reference to the construction of a restaurant manager model.

In Chapter 4, we consider computers for learning from an educational perspective. We begin by discussing the educational theories of instructionism and constructionism. We adopt a broad perspective on constructionism that encompasses both bricolage and situated learning. Bricolage is a style of construction that puts emphasis on close personal engagement with a task where the evolving construction goes hand in hand with increasing comprehension of the task. Situated learning holds that the surrounding domain context of a problem is not incidental to its solution but provides the necessary social handles for learners to grapple successfully with the problem. We argue that principles and techniques for computer-based model construction must support bricolage and situated learning if they are not to inhibit domain learning. In the remainder of the chapter, we consider three practical techniques that can be applied in domain learning: concept mapping, conventional programming and EM. We discuss the extent to which each of these techniques supports the broad perspective on constructionism and how it is related to the EFL. We argue that concept mapping is only useful in the very early stages of learning and that knowledge gained using it is typically set aside when constructing models. We argue that conventional programming is not well oriented to the constructionist agenda because it emphasises planning, abstraction and circumscription and this detracts from its usefulness as an approach to model construction that promotes domain learning. Finally, we argue that an EM approach to model construction can support our broad perspective on constructionism and learning activities across the EFL, enabling effective domain learning to proceed in tandem with model construction. We illustrate this claim with reference to the construction of a digital watch model.

In Chapter 5, we consider the advantages of using EM to construct learning environments that support many different types of learning objective. We identify three types of learning that can be scaffolded in EM: comprehension of a fixed referent; exploration of possibilities and invention; and learning languages. We illustrate each of these types of learning by case studies in the form of EM models in which learning is scaffolded through gradual embellishment of the model. A different style of presentation to the learner is characteristic of each type of learning. In the Racing Cars model, the referent is fixed from the outset and each layer of the model adds a greater subset of the functionality to the interface so that the learner can explore more complex ideas. In the OXO case study, the model is built up incrementally, and – although a specific learning path is mapped out – the model is flexibly adaptable to different teaching requirements. We illustrate this adaptability by creating a family of games related to noughts-and-crosses. In respect of learning languages, we introduce an EM parsing utility that can allow languages to be incrementally extended or refined as a learner is interacting with a model. We use case studies based on a simple LOGO-like language and a more complex database query language to illustrate language learning in conjunction with emerging domain understanding.

In Chapter 6, we discuss and illustrate the links between EM and the EFL with reference to more elaborate case studies. Conventionally, there are two ways in which computers support learning: through personal model building and through the use of pre-constructed models that cannot be revised by the user. We argue that it is possible to have a third category, namely models that are partially built and that can be extended by a learner in response to their particular learning needs. We present three EM case studies that exhibit different degrees of model building and model use: the Free Distributive Lattice model; the Heapsort model; and the Robotic Simulation Environment. Each model places different demands on the learner and this is reflected in the specific learning activities that it supports within the EFL. These case studies give practical evidence in justification of our claim that EM can support learning activities from across the whole of the EFL.

Chapter 7 summarises the research undertaken for the thesis, drawing some conclusions, considering its limitations and outlining possible future work.