Empirical Modelling as a new paradigm for educational technology

by

Antony James Harfield

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy in Computer Science

Department of Computer Science

October 2007

THE UNIVERSITY OF WARWICK

Contents

List of Figures

Acknowledgements

I would like to express my deepest gratitude to my supervisor Meurig Beynon for his help and support in realising this thesis, and for making the PhD journey a fulfilling and lively one. Thank you also to Steve Russ and the other past and present members of the Empirical Modelling Research Group, particularly Russell, Charlie, Eric, Karl and Ashley. Our many seminars and informal discussions have been fun and exciting, and also stimulated many of the ideas in this thesis. I feel lucky to have worked with this inspired group.

Next I would like to thank my family, particularly my parents who have supported me and encouraged me through all of life's adventures, and my sister Vicky who has always been one step ahead of me, I am proud to say. I also count myself lucky in being related to a small but inspirational family of grandparents, uncles and aunties.

Lastly I would like to say thank you to all those that have touched my life at Warwick. Special appreciation goes to my nearest and dearest friends Pie, Mark, Hema, Pete, Michael, Jungli, Min, Peng, Jin, Kay and especially Tam—let these great karmic connections continue. My thanks and respect to Venerable Luangpor Khemadhammo for his spiritual guidance and everyone at The Forest Hermitage for your friendship.

May you all be well and happy.

Declaration

This thesis is presented in accordance with the regulations for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated.

The section on the Testing Lossless Join model in Chapter 4 extends a joint paper presented at ICALT 2005 [BH05b] and a related research report [BH05a]. The discussion of human-robot interaction in Chapter 4 elaborates on a joint paper presented at AISB 2005 [BHC05]. The application of EM to lifelong learning discussed in Chapter 5 is based on a joint paper presented at ICALT 2006 [BH06]. The explanation of the Clayton Tunnel railway accident and the accompanying distributed model in Chapter 5 is discussed in a poster prepared for Kaleidoscope Showcase 2005 [HCW05]. The analysis of empirical evidence from the introductory Empirical Modelling module in Chapter 6 draws on material from another joint paper presented at ICALT 2006 [BHB06]. The suitability of Empirical Modelling as a constructivist approach to learning explored in Chapter 7 builds on ideas in the latter part of a paper published in the Journal of Computers [BH07].

The thesis also elaborates other papers, presentations and projects that I have been involved with: a forthcoming paper for the 7th Baltic Sea Conference on Computing Education Research (Koli Calling 2007) [BHV07]; a presentation to the British Computer Society Coventry Branch [Har06]; a joint paper on concretisation presented at the Koli Calling 2005 [BHJ05]; a presentation at the PPIG Unroll Your Ideas Workshop 2004 [Har04]; and an undergraduate project completed in 2003 [Har03].

Abstract

Educational technology has yet to deliver the benefits or successes that were expected in educational practice, especially in relation to issues other than the communication and delivery of teaching materials. Evidence suggests that these difficulties stem from the mismatch between formalised virtual learning environments and everyday sense-making and between the rich potential for enhanced learning afforded by new technology and the constraints of old-style educational practice. In addressing this mismatch, some commentators suggest that the primary need is for a new culture of educational practice—and even that such a culture is already emerging, and others identify the need for a new paradigm for educational technology. The aim of this thesis is to explore the potential for a new paradigm for educational technology based on the principles and tools of Empirical Modelling (see `http://dcs.warwick.ac.uk/modelling`).

The thesis builds upon previous research on Empirical Modelling as a constructionist approach to learning, and in particular Roe's doctoral thesis 'Computers for learning: an Empirical Modelling perspective'. Roe's treatment of Empirical Modelling can be viewed as generalising the use of spreadsheets for learning through applying 'programming by dependency' within the framework of existing educational practice. In contrast, this thesis is concerned at a more fundamental level with the contribution that Empirical Modelling can make to technology enhanced learning that may lead to new educational practices. In particular, it identifies eight significant characteristics of learning that are well-matched to Empirical Modelling activity, and associates these with experimental, flexible and meaningful strands in learning. The credentials of Empirical Modelling as a potential new foundation for educational technology are enhanced by demonstrating that Empirical Modelling is radically different from traditional software development and use. It provides a methodology for modelling with dependency that is more closely related to the use of spreadsheets for learning.

The thesis elaborates on the relationship between Empirical Modelling and learning in a variety of different contexts, ways and applications. Three examples drawn from computer science higher education are explored to emphasise the experimental, flexible and meaningful characteristics of Empirical Modelling. This discussion of Empirical Modelling in a specific educational context is complemented by an investigation of its relevance to learning in a wider context, with reference to a broad range of subjects, to specific issues in language learning, and to the topics of lifelong learning and collaborative learning. Although the application of Empirical Modelling for learning is as yet too immature for large scale empirical studies, its potential is evaluated using informal empirical evidence arising from Empirical Modelling practice at Warwick. The sources for this evaluation are well-established teaching activities relating to Empirical Modelling in Computer Science at the University of Warwick, comprising an introductory module and a number of final year undergraduate projects.

The thesis concludes by considering the extent to which Empirical Modelling can go beyond the support for constructionism envisaged by Roe, to address the broader agenda of supporting constructivist learning using computers. To this end, a close relationship between Empirical Modelling and a vision of constructivism recently set out by Bruno Latour in his paper 'The Promises of Constructivism' is demonstrated.

Abbreviations

AI	Artificial Intelligence
AOP	Agent-Oriented Parser
CALL	Computer Assisted Language Learning
CSCL	Computer Supported Collaborative Learning
DoNaLD	Definitive Notation for Line Drawing
EDEN	Evaluator of DEfinitive Notations
EFL	Experiential Framework for Learning
EM	Empirical Modelling
EMPE	Empirical Modelling Presentation Environment
ET	Educational Technology
GEL	Graphical Environment Language
HRI	Human-Robot Interaction
HTML	HyperText Markup Language
TEL	Technology Enhanced Learning
ODA	Observables, Dependencies and Agency
SASAMI	Solids Animation Simulator And Modelling Interface
SCOUT	SCreen LayOUT notation
STD	Student, Teacher and Developer
TLJ	Testing Lossless Join

Introduction

Empirical Modelling (EM) is a body of research that has been developed at the University of Warwick by Beynon & Russ [EMW] over the last 20 years. The principles and tools for EM enable model construction and manipulation with a unique emphasis on the role of experience. Model-building is seen as intimately connected with sense-making, and models have a high degree of openness encouraging exploration and embellishment. EM has been used by over 100 students and many researchers for model-building activities, and some of the resulting models are featured in the EM Project Archive, which contains over 160 models at present [EMP]. These models explore a wide variety of topics in computer science, such as concurrent systems, computer graphics and artificial intelligence, and in other subject areas including engineering, business, humanities and education. This work has led to over 100 refereed publications and more than 20 graduate theses on topics relating to computing. The principles, tools, history and philosophy are discussed in depth in the theses of Rungrattanaubol [Run02], Ward [War04], and King [Kin07].

The topic of this thesis is EM as an approach to learning and as a support for education. Given the importance assigned by Beynon & Russ [BR07] to personal sense-making in EM, it is natural to assume that EM is already associated with learning in an informal sense. The thesis develops the idea of *EM as learning* together with the use of *EM for learning* in terms of a methodology for technology enhanced learning. The overall aim of the thesis is to answer the following research question: *How*, *where* and *why* can EM benefit learning? In order to answer the 'why' question, I shall be drawing on external influences that have provided the *motivation* for EM. To answer the 'where' question, I analyse and extend research into EM's *principles and tools* in practical applications. And to answer the 'how', I have gathered *evidence* of student EM activity. These three sections are depicted in Figure 0 and elaborated in this introduction.

The earliest work on the connection between EM and learning is by Beynon [Bey97], who highlights that EM has potential for applications in education because EM's principles for model-building are bound up with the learning process. According to Beynon's experiential framework for learning[†], to be discussed further in Chapter 2 (see Figure 2.7 on page 51), "cognition and learning are fundamentally concerned with a process of construing phenomena in terms of agency and dependency" [Bey97]. Such a process is empirical, as well as provisional and tentative in that the function of the model is to prompt more precise understanding of the phenomena. Beynon introduces EM as a computer-based approach to constructing models of phenomena in terms of agency and dependency that is concerned with understanding phenomena prior to precise formalisation [Bey97]. This leads Beynon to recognise EM's potential as an approach to generating software for educational use [Bey97]. Beynon demonstrates that EM can offer support for concerns faced by IT managers, teachers and pupils in developing educational software [Bey97]. This work offers some initial clues for answering the 'why' question.

A detailed account of the potential application of Empirical Modelling (EM) to learning is given by Roe in his PhD thesis entitled *Computers for Learning: An Empirical Modelling Perspective* [Roe03]. Roe takes Papert's constructionism [PH91] (introduced in Chapter 2) as his foundation and argues that technology enhanced learning has only realised a fraction of its full potential because traditional programming practice is not well-suited to the needs of domain learning. He suggests that educational technology based on spreadsheet principles (discussed in Chapter 3) offers a more suitable paradigm to support domain learning [Roe03:p.20]. Elaborating on this notion, Roe introduces EM as a powerful tool for constructionist learning offering the characteristics of a spreadsheet environment (i.e. dependency maintenance) with the power of a procedural programming language [Roe03:p.55]. Work by other members of the EM Research Group has improved the general understanding of EM principles and tools. Rungrattanaubol [Run02] has improved the understanding of the principles of EM with her treatise on Modelling with Definitive Scripts (MWDS). Ward has developed and analysed the EM tools over a number of years in response to the needs of students

[†]Originally referred to as an empiricist perspective on learning.

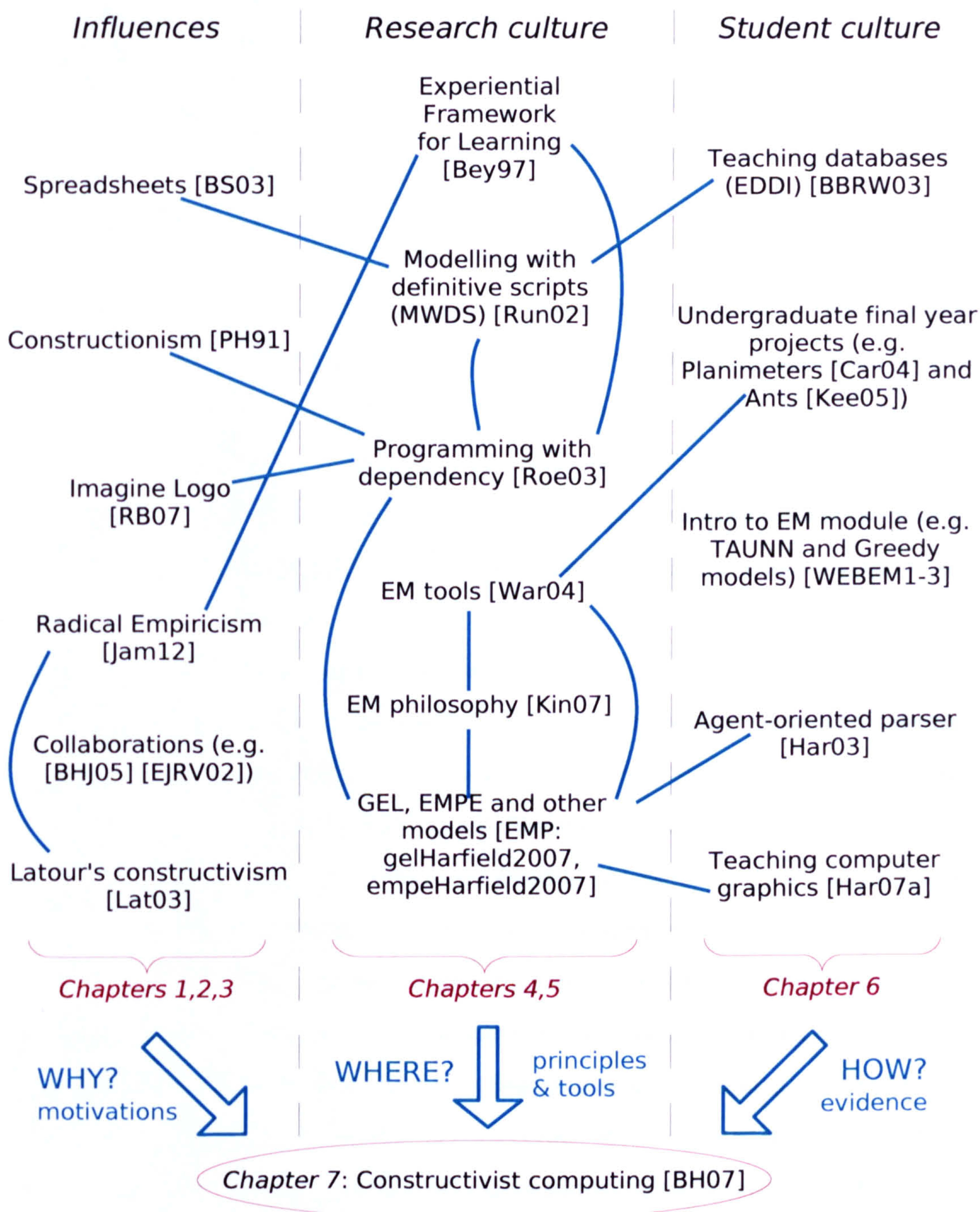


Figure 0: An outline of the thesis drawing together all aspects of EM and learning.

[War04]. King has developed the philosophical basis for EM and prototyped new interfaces for EM [Kin07]. This work provides a basic starting point for answering the 'where' question.

The student-led culture of EM at Warwick has contributed to the application of EM as a support for education. Students have been involved with EM tools through interaction for a module in databases [BBRW03]. Many final year undergraduate projects have engaged with EM (e.g. the projects on planimeters and ant navigation as discussed in Chapter 6) and also developed new EM tools (e.g. the agent-oriented parser [Har03] as used in Chapter 4). An introductory module to EM, discussed in Chapter 6, has provided students with further model-building opportunities—some models themselves being relevant to education (e.g. the TAUNN and greedy algorithm models from WEB-EM-1 [WEBEM1]). The student culture, as well as collaborations external to Warwick (e.g. joint work with University of Joensuu, Finland [EJRV02] [BHJ05] [BHV07]), has inspired new research directions and provided evidence for the potential of EM in education. This work gives an empirical basis for answering the 'how' question.

In answering the 'how', 'where' and 'why' questions, the thesis brings together evidence from student EM activity, research work into principles and tools, and external influences that have motivated EM. The main contribution of the thesis is to redefine EM as a complete approach to learning and education drawing on all of the previous work as shown in Figure 0. Instead of solely viewing EM *as* learning, as introduced by Beynon, or EM as 'programming with dependency' as developed by Roe, the contribution of the thesis is to acknowledge broader characteristics of EM activity that make it appropriate to consider EM as a new paradigm for technology enhanced learning.

This thesis makes a significant contribution to our understanding of the benefits of EM as an approach to learning and education. Where technology is concerned, there has been much progress in EM's principles and tools, supported by evidence and examples, but the evaluation of the work from an education perspective has been limited by current resources. More support is needed from educationalists and practictioners to do justice to the current work in EM for learning. The aim of the thesis is to develop a broad view of EM in support of learning in order to provide a firm base for future work by educational researchers and practictioners in schools, universities and other organisations.

The thesis can be roughly classified into three sections aimed at providing answers to the three research questions.

Chapters 1, 2 & 3 are concerned with *why* EM might be of benefit to learning and education. In Chapter 1, some issues in educational technology are highlighted that have led to a paradigm conflict between learning using computers and learning in an informal everyday sense, and between the potential for enhanced learning afforded by technology and the constraints of old-style educational practice. An attitude to learning is described, characterised as *experimental, flexible* and *meaningful,* to which educational technology might aspire in order to alleviate issues in the paradigm conflict. Chapter 2 introduces EM as a computer-based approach to model-building well-aligned to supporting learning with experimental, flexible and meaningful characteristics—providing an answer for 'why EM?'. Chapter 3 clearly distinguishes EM from programming, explaining 'why' EM offers a greater potential—over traditional software development and use—for supporting learning as characterised, and compares EM closely to model-building using spreadsheets.

Chapters 4 & 5 are concerned with *where* EM can benefit learning. In Chapter 4, applications of EM to teaching and learning in computer science are explored with specific reference to supporting experimental, flexible and meaningful characteristics of learning in topics from databases, computer graphics and artificial intelligence. Chapter 5 identifies examples of EM support for learning in other subject areas, with a specific example from language learning, for teaching through presentations, for lifelong learning and for collaborative learning. These two chapters contribute to demonstrating 'where' EM might support learning.

Chapter 6 contributes to the 'how' section by providing examples and evidence *how* EM can benefit learning and education. Chapter 6 examines empirical evidence from projects and coursework undertaken by students in computer science at Warwick.

The thesis concludes with Chapter 7 which aims to bring together all three sections—the motivations from Chapters 1, 2 & 3, the principles and tools from Chapters 4 & 5, and the evidence from Chapter 6—as depicted in Figure 0. Chapter 7 considers EM's support for learning from a constructivist viewpoint, resulting in all the elements of EM for learning being brought together in a vision for 'constructivist computing'.

Chapter 1

Paradigmatic challenges for educational technology

> "All social movements involve conflicts which are reflected intellectually in controversies. It would not be a sign of health if such an important social interest as education were not also an arena of struggles, practical and theoretical." John Dewey [Dew59:pv]

The aim of this chapter is to highlight that, despite the hype, there are some difficulties with the application of educational technology (ET). I shall concentrate on the need for ET to offer more of the *experimental*, *flexible* and *meaningful* characteristics of everyday learning in the world. Current ET struggles to address this issue because of a 'paradigm conflict' between learning with computers and learning in an everyday sense. The 'conceptual challenges', as opposed to implementation or political challenges, concern the need to bridge the gap between computing activity and learning as a sense-making activity. To overcome these challenges, *eight significant characteristics of learning* are introduced that ET should aspire to support. The remainder of the thesis explores an approach to computing, Empirical Modelling (EM), that aims to support these characteristics.

1.1 Educational technology and paradigmatic conflicts

1.1.1 What is educational technology?

Educational technology (ET), e-learning, Information and Communication Technology (ICT), and Computer Aided Learning (CAL) are all terms that refer to a very wide range of computer-related technologies that support teaching or learning. They include the use of multimedia CD-ROMs, web-based teaching materials, discussion boards, collaborative software, e-mail, blogs, wikis, chat software, computer aided assessment software, educational animation, simulations, games, learning management software, intelligent tutoring systems, integrated learning systems, mobile learning, to name just a few. A description of educational technologies by the Joint Information Systems Committee identifies a wide range of activities in which computers may be of assistance, from blended learning where traditional learning is combined with technology, to learning that is completely delivered through a computer [JISC04]. Some potential benefits of ET are identified as [JISC04]:

- learning material and support can be accessed from anywhere (assuming appropriate technology is available);
- learning material and support can be accessed at any time;
- feedback from learning can be instantaneous;
- collaborative tools allow learning to extend beyond the current physical location (using the Internet for example);
- learning material and support can extend the potential of traditional learning environments;
- learning material and support using technology can be fun.

While it is clear that ET can have a positive effect on learning, the benefits highlighted above paint a picture of technology as providing better delivery and communication mechanisms (with the exception of the benefit of technology being fun).

1.1.2 The reality of educational technology in the educational system

Technology is seen by many as a way of improving education. Figures within the British government have indicated that technology will drive improvements in learning and education. Jim Knight, Minister of State for Schools, in his 2007 BETT Conference speech [Kni07] declared that: "Technology will be a vital part of our drive to securing higher standards and better schools for all." The Gilbert Report [Gil07], presented to the Secretary of State as a vision for learning in 2020, states clearly that "technology influences what, how and why children learn". In a survey on the progress of education in Britain [GdC05], Green, de Waal & Cackett point out that expenditure on education has risen by 5% each year since 2000, with increasingly more of each budget being spent on technology. According to advice presented to the government, ET is supposed to enable us to learn what we want, when we want [Gil07]—and the government is equally enthusiastic with one senior member ambitiously declaring that education with technology has the potential to be "the great liberating force in providing opportunity to all" [BEC07].

Although there is widespread acceptance that technology *can* bring benefits to education, there is some concern whether current technology *is* of actual benefit. In the UK, the common criticism that ET is ineffective due to lack of government funding comes at a time when expenditure on education is increasing and overall expenditure is above the European average [GdC05]. Albirini, in a paper on the crisis in educational technology [Alb07], highlights the growing scepticism surrounding the value of ET: "Despite huge expenditure, wide experimentation and research, and discursive enthusiasm, educational technology has failed to show substantial benefits to the field". Following this, he adds that "efforts to explain and subsequently resolve the crisis of educational technology have centered mainly on the material obstacles to the implementation of educational technology in schools", such as lack of funds, inadequate planning, shortage of computing expertise, insufficient teacher support, and a fear of security and misuse. Albirini, however, is more concerned that "the real causes of the crisis extend beyond these concrete problems to more theoretical issues related to the *'identity' of educational technology*, its *theoretical assumptions*, and its *paradigmatic conflict with education*" (my italics) [Alb07].

1.1.3 Explaining the crisis

Education technology has been promoted as having the potential to transform education, with the same revolutionary and reformist attitude with which the information age transformed the industrial age [Alb07]. Albirini explains that the drivers behind the information age were stand-alone tools that automated the human element to some extent, putting the focus on the computer not the person to do the job. While this has been successful in industry, education has not been able to accommodate this approach and holds on to a more structured dependable industrial system controlled by a human element [Alb07]. Thus, Albirini argues, the 'identity' of technology is questioned in education as it has not yet shown the revolutionary or reformist role with which it is associated. The second issue that Albirini considers is the disparity between the assumptions of ET and those of education [Alb07]. Educational technology aims to remove the structure of the classroom, decentralise access to tuition, increase access to material, and enhance student collaboration and exploration [Alb07]. The conflict arises because education is founded upon a top-down hierarchical structure from the educational authority right down to the classroom, linearly organised activities, the presentation of prescribed material, and teacher-to-student interaction [Alb07]. Albirini views the mismatch between these assumptions as a major obstacle to the integration of ET in classrooms.

Findings by Goodson and Mangan [GM95] suggest that the demands of computer use on existing learning environments (e.g. schools) lead to a 'culture clash' between the computer and many areas of the curriculum. They reason that, due to the pedagogical and organisational changes that computer use dictates, some subjects in the curriculum are unlikely to be compatible with the use of computers. Selwyn [Sel99] discusses the attitudes of 16-19 year old students to computers in the classroom, and his findings show that attitudes vary mainly according to subject areas as well as student access to computers. These findings suggest that ET, and its designers, should be more sensitive to not only the paradigm of learning, but also the differences in culture across a wide range of learning situations and environments.

Albirini says that in the end we have to choose whether we shall stick with our industrial system for education, or if instead we should develop a new paradigm for education that is more aligned to technology and the information age [Alb07]. From Shaffer & Kaput's point of view, a new paradigm is already developing [SK99]. In their evolutionary perspective on technology and mathematics education, they argue that a new cognitive culture, which they term 'virtual culture', is developing based on the use technology for the externalisation of symbolic processing [SK99]. Shaffer & Kaput view 'virtual culture' as a fifth stage that follows on from Merlin Donald's acclaimed analysis of human culture into four distinct cognitive development stages over a period of at least three million years [SK99]. Donald's fourth stage is identified as 'theoretic culture' or culture based on external representations such as written symbols [Don91]. The development of the cognitive ability to use external representations made it possible for humans to keep records, as well as reflect on the interrelationships among recorded ideas, and Donald suggests that modern scientific culture developed from the existence of external notations for thinking [Don91:p320]. Shaffer & Kaput's fifth stage differs from the fourth stage in that human beings have developed the cognitive ability to use external *processing* (as a consequence of technological developments) [SK99]. The simple example given is that of a spell-checker which performs a processing task that would have previously been performed by a person [SK99]. At the present time, we are only at the very beginning of 'virtual culture', given that 'theoretical culture' goes back around 30,000 years [SK99]. Shaffer & Kaput explore the possible changes in mathematics education that may result if this new virtual culture develops [SK99], and if they are correct then we are, as Albirini says [Alb07], on the verge of a new paradigm for education.

The prospects of a new paradigm for education are further emphasised by Riley [Ril07], a researcher in the history of education and pedagogic innovation. Riley explains the use of ET as fitting into three idealised classes: functional substitution, functional delegation, and functional innovation [Ril07]. Functional substitution is associated with the typical use of multimedia, where previous mediums of transmission are replaced by technology for (e.g.) more realistic graphics. Functional delegation is associated with the use of word processors, spreadsheets, and other generic software that can simplify the concerns of the teacher or student as tasks are delegated to the computer. Both of these uses of ET have been relatively well-explored. The third classification of use is functional innovation, which has so far been associated primarily with computer modelling. Models can serve as "a way of thinking, a means of expression, and a subject of investigation" [Ril07]. The functional innovation use of ET is what Riley views as having the potential to significantly change education because it involves for the first time putting 'heads and computers together' [Ril07]. Riley suggests that the full character of functional innovation has yet to be realised and if it does evolve into the new paradigm for education then the change may be measured in generations rather than decades.

1.1.4 Unravelling the paradigm conflict

The above discussion highlights one aspect to the paradigm conflict that there is a mismatch between the rich potential for enhanced learning afforded by new technology and the constraints of old-style educational practice. However, the focus of this thesis is at a more practical or technical level. My interest—and the link with EM—is in how ET can be developed to overcome the conflict and mismatch, or to support a new paradigm for education. Such changes or developments, as noticed by Riley, occur on a macro-level over long periods of time, whereas on a micro-level of interactions among students, change in ET can have an immediate effect [Ril07]. Therefore, the task is to consider the paradigm conflict at a primitive level, in particular at the level of characteristics of learning that ET can aim to achieve, which may eventually lead to new practices and a new paradigm for education.

In some respects Albirini's explanation of the paradigm conflict [Alb07] is too loose to be able to offer any suggestions for improving ET. However the theoretical issues he highlights are echoed in Jonassen's concerns towards educational technologists in his book entitled *Modeling with Technology* [Jon06]. Jonassen states more clearly that a major problem with ET is that educational technologists have assumed that if you create lessons that use technology and show them to students then they will learn [Jon06:pxiii]. Under these theoretical assumptions the purpose of the technology is to communicate ideas to learners, thus replacing the role of the teacher. As a technologist (or computer scientist), the primary challenge is to make the communication as efficient and effective as possible, as can be seen from the benefits that JISC proclaim [JISC04]. The reason that problems arise, as stated by Jonassen, is that "students do not learn from technology; they learn from thinking" [Jon06:pxiii]. Therefore a tension exists between those concerned with the benefits of technology enhanced learning and those interested in how we learn. Jonassen's work is special in this respect as he starts the premise that meaningful learning involves conceptual change and then he goes on to show how modelling with technology can bring about conceptual change.

Jonassen's notion of *conceptual change* originates from the work of educational psychologists Strike & Posner [SP85]. The basis for viewing learning as conceptual change begins with the assumption that humans are natural theory builders; from an early age we build "intuitive personal theories" to explain the external world in which we live

[Jon06:p4]. When these theories, or *concepts*, conflict with new experiences or cannot be used to solve problems then *change* can sometimes occur [Jon06:p5]. As Jonassen describes, "Conceptual change occurs when learners change their understanding of the concepts they use and of the conceptual frameworks that encompass them." [Jon06:p4]. For Jonassen, this change occurs best when learners are engaged with building models.

If a conceptual change view of learning is adopted then the focus of education should be on creating experiences that 'prod' at our current concepts and make us reconsider our personal theories in everyday life—what we need is an approach to learning emphasising *everyday sense-making*.

This leads to two tensions in the paradigm conflict: the difficulties of technology enhanced learning stem from the mismatch on a high-level between the rich potential for enhanced learning afforded by new technology and the constraints of old-style educational practice; and also between formalised virtual learning environments and everyday sense-making.

The first tension in the paradigm conflict results in ET being concerned with higher-level issues of transmission, delivery and communication. This leads to the second tension that ET is not appropriate for everyday sense-making or learning in everyday situations which is actually concerned with the primitive activity of conceptual change. The paradigm conflict can be examined in more detail by comparing the nature of 'learning with ET' and 'learning in the world' in an everyday sense. Contrasting Figure 1.2 with Figure 1.1 illustrates the differences between these two paradigms that many educational technologists are attempting to combine.

The issues resulting from attempting to mix 'learning with ET' and 'learning in the world'—as depicted in Figure 1.2 and Figure 1.1—suggest ET lacks attention to the primitive aspects of learning in the world. These aspects are not difficult to find, they are a part of everyday living, and are evident from the things we learn in the world on a day-to-day basis. It is learning that means something to us, or enables us to do something. It is learning that brings about conceptual change [Jon06]. Some of the relevant characteristics of such learning is what I shall attempt to describe in the next section. Carl Rogers, best known for his role in the development of client-centred therapy or counselling, might describe this everyday learning as 'significant learning' referring to its characteristics as a genuine type of learning [Rog61]. By this Rogers

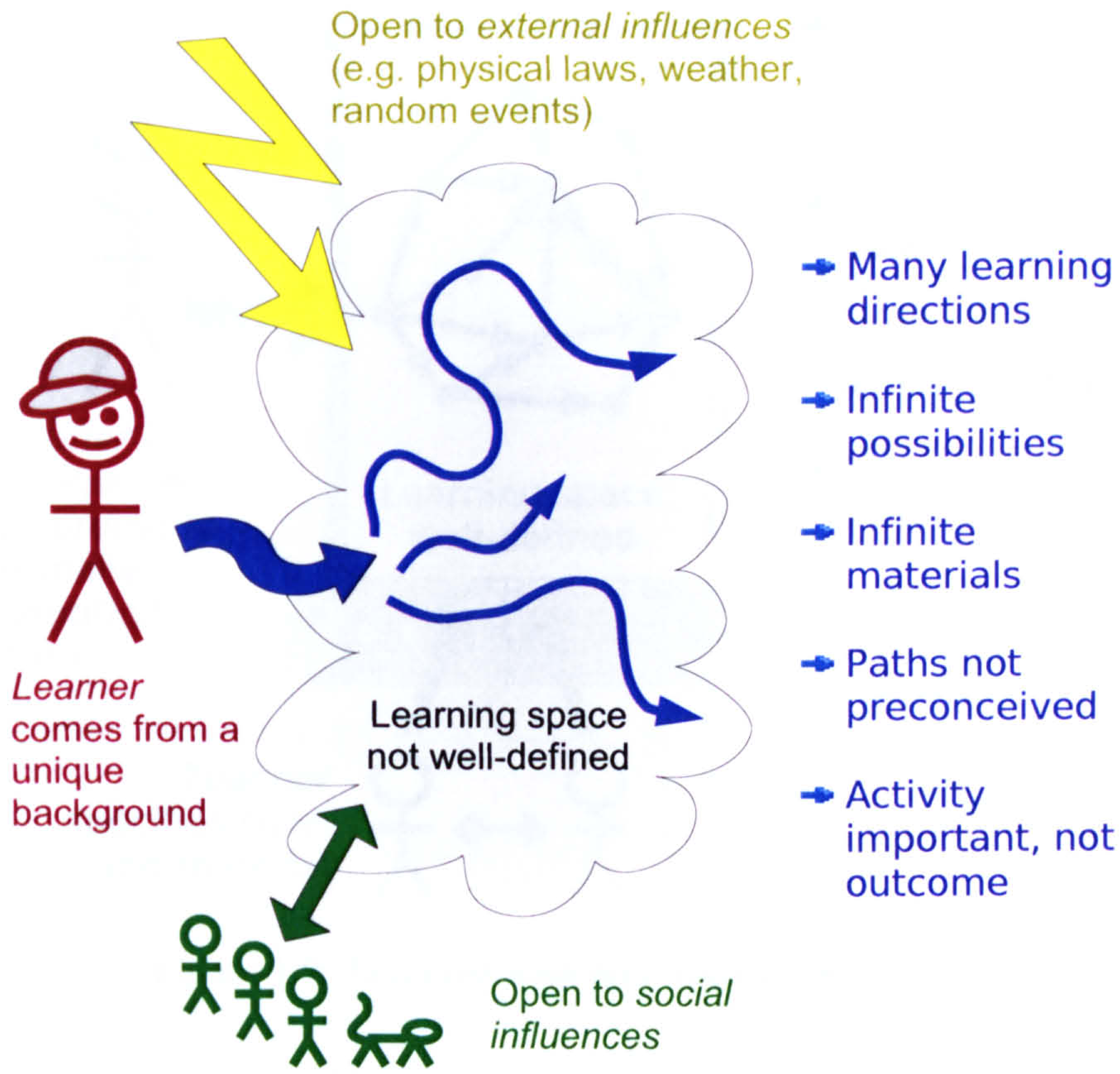


Figure 1.1: Learning in the world in everyday life.

means "learning which is more than an accumulation of facts ... it is learning which makes a difference—in the individual's behaviour, in the course of actions he chooses in the future, in his attitudes and in his personality" [Rog61:p280]. The struggle to construct this thesis is an example of the learning that is apparent in the problems of our everyday lives. We juggle ideas, try to formalise our thoughts, experiment with trials, modify our model based on errors or mistakes—these aspects are familiar to everyday activities. (This thesis represents only the outcome of a learning process which has involved a significant amount of experimentation, trial and error, wrestling with thoughts and formulating new ideas.) The relevant properties of everyday learning are that the learning space is not well-defined, it is open to external and social influences, the learning may take many directions, the possibilities are endless, the learning space contains an infinite amount of material, there are no preconceived paths for the learning, and it is the activity of learning—not the outcome—that is important. Figure 1.1 attempts to illustrate the nature of everyday learning in the world: that the learning ebbs and flows with one's life in response to internal and external influences.

The problems with existing ET surround an attachment to learning as transmission,

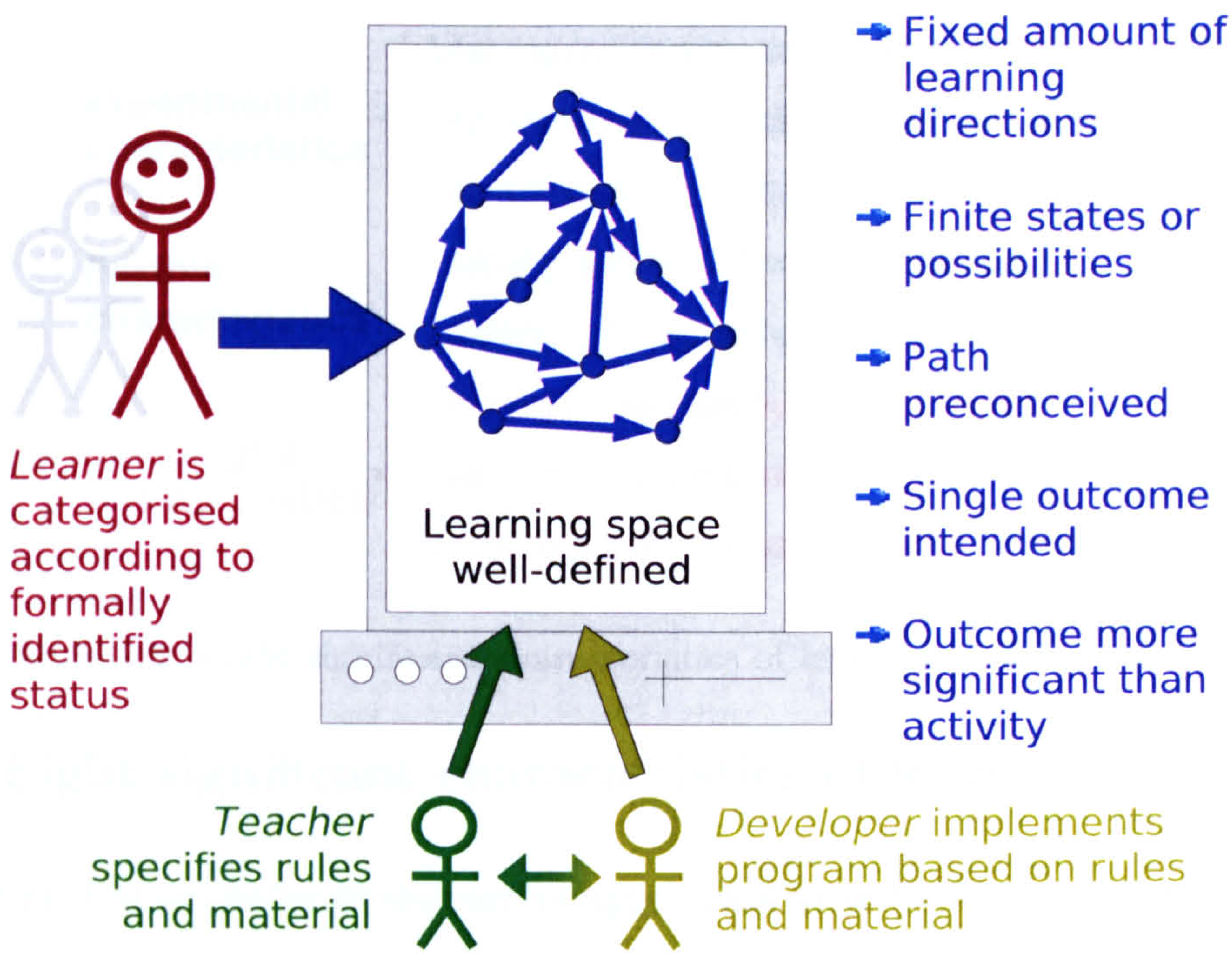


Figure 1.2: Learning with educational technology.

delivery, and communication [Jon06], as well as being built on a foundation of computer science. The properties of learning with ET are that the learner is categorised according to a formally defined status, the learning space is well-defined, the learning space can only be changed by the teacher who specifies the requirements and the developer who implements the requirements. Thus the learner can only explore a fixed number of preconceived learning paths, containing a finite number of possibilities, leading to a specific learning outcome, the result of which is often treated more important than the activity of learning. Figure 1.2 shows the nature of learning with ET. Such learning does not encourage the making of meaning, or 'significant learning' as Rogers describes, in an individual as put bluntly by Jonassen: "Technology-centric approaches to education ignore the sole purpose of technology in classrooms: to support meaningful learning." [Jon06:pxiii].

Jonassen's suggestion, which shall be followed up in this thesis, is that "rather than analyzing how technology can teach better, educators need to consider how students must think to learn most meaningfully." [Jon06:xiii]. Put in the terms used in this thesis, in order to resolve the paradigm conflict, ET must support learning on a more primitive everyday level.

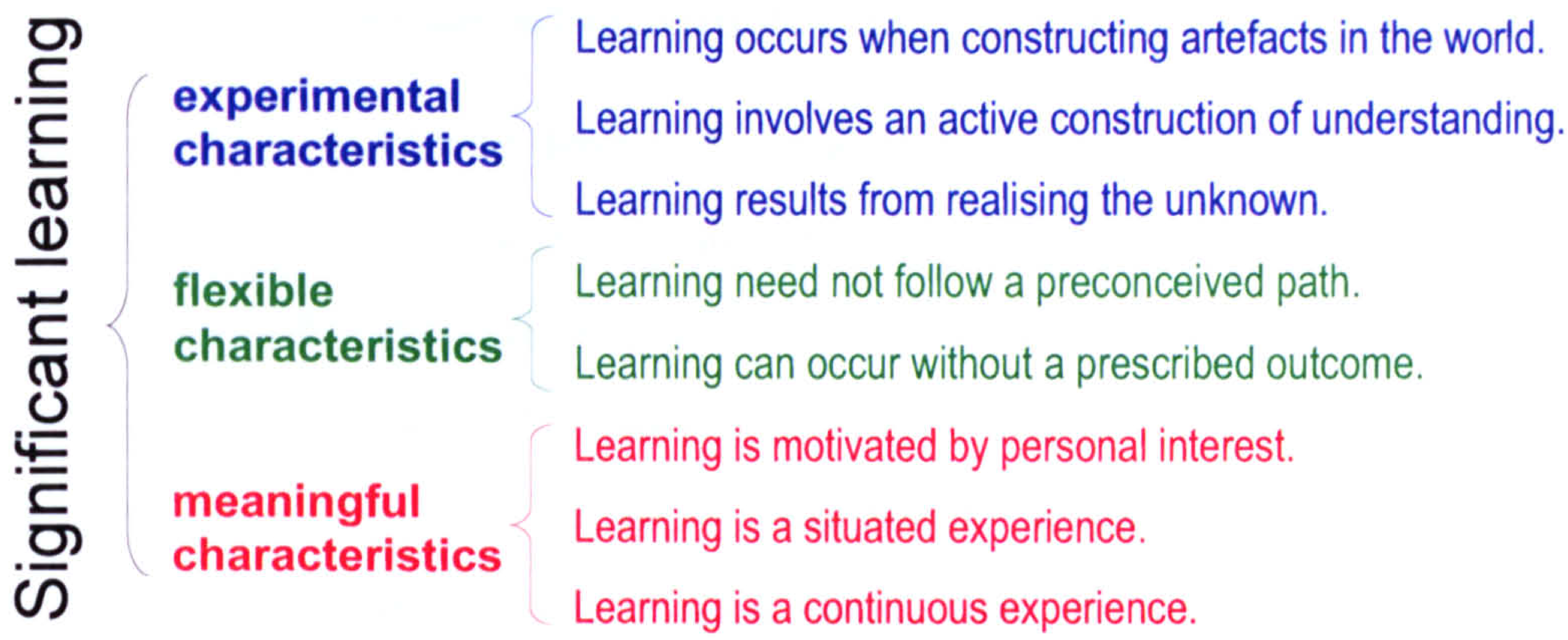


Figure 1.3: Eight significant characteristics of learning in three strands.

1.2 Eight significant characteristics of learning

In order for EM to challenge the current approaches to technology enhanced learning, the above discussion points to the need to focus on supporting the primitive aspects of learning that occur in everyday situations. This section introduces *eight significant characteristics of learning* that feature in everyday learning.

The eight characteristics are the result of the observed need to make ‘learning with ET’ as depicted in Figure 1.2 more like ‘everyday learning’ as depicted in Figure 1.1. These characteristics reflect the kind of learning that ET should promote in order to alleviate the tensions surrounding the paradigm conflict. This section introduces each characteristic in relation to theories of learning originating from a wide range of educational and philosophical thinkers. As introduced in the next chapter, the eight significant characteristics of learning share a close affinity with the characteristics of EM.

The eight characteristics are broken down into three strands. The first of these strands, explained by the first three characteristics, views learning as essentially *experimental*, both on a practical level and in terms of knowledge. The second strand is associated with a view of learning as open-ended or *flexible*, as described by the middle two characteristics. The last strand is associated with learning that is relevant or *meaningful* to the learner, as implied by the last three characteristics. Figure 1.3 illustrates the breakdown of the *eight significant characteristics of learning*.

1.2.1 Learning occurs when constructing artefacts in the world

The first characteristic to be described is one that is particularly relevant to ET. It has its roots in 'learning by making' and was first explained by computer scientist and educator Seymour Papert as constructionism. The idea behind constructionism is that learning occurs "especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe" [PH91]. Papert claims that the constructionist idea is an extension of constructivism, in that the latter is concerned with the construction in the head and the former links this to construction in the world. However, given the many conflicting flavours of constructivism (discussed further in Chapter 7), I shall avoid describing constructivism and simply talk about constructionism as an activity in which the construction of artefacts in the world can lead to the development of understanding. Papert's particular emphasis is on the construction of artefacts using computers, such as the LOGO environment [Pap80].

The concept of 'learning by doing' only captures Papert's idea of constructionism in very general terms. 'Learning by doing' as an idea has a long history—the great Chinese philosopher Confucius is widely attributed as having said "I hear and I forget. I see and I remember. I do and I understand." and Aristotle is quoted as having said "What we have to learn to do, we learn by doing." [Ari12]. Although Papert might be more likely to think of constructionism as 'learning by making', both forms of thinking absolve the teacher somewhat from their traditional teaching role. Such ideas are linked to the great education reformist John Dewey:

> "I believe that much of the time and attention now given to the preparation and presentation of lessons might be more wisely and profitably expended in training the child's power of imagery and in seeing to it that he was continually forming definite, vivid, and growing images of the various subjects with which he comes in contact in his experience." [Dwo59:p29]

Through Dewey we can see that construction can be linked to imagination and the forming of ideas. It is Dewey that argues for the importance of learners being able to investigate things for themselves, and not take the teachers' words as absolute. The role of the teacher (and now, of ET) is not to constrain the thinking of the student, but to prompt in the student's imagination the construction of new ideas in relation to their everyday experience.

Creativity therefore plays an important role in learning when there is an emphasis on constructing. Negus & Pickering, in a book devoted to explaining creativity [NP04:p22], discuss the nature of creative experiences being something often not describable but 'intensely felt' in the same way that Confucius sees true understanding as only arising out of doing. Negus & Pickering quote the controversial American poet John Ashbery as saying, "If I did not write, I would have no idea of what I can write. I suppose that I write so as to find what I have to write." [NP04:c4]. The close connection between creativity, imagination and significant learning is reflected in constructionism. Papert believes strongly that these elements should be imbued in ET as emphasised in a talk relating to educational change: "Wild imagination, passion, being close to nature, and believing in magic—that is what we need. I think these are all the elements that we need to bring into the otherwise cold version of use of computers called 'ICT'." [PS05].

Papert's constructionism and the idea of learning by doing highlight very general practices that may bring about change in education. Riley's idealised use of ET for functional innovation [Ril07] provides a more focussed idea based upon 'learning by building models' as discussed earlier in this chapter. Riley takes up Jonassen's standpoint that model-building brings about 'conceptual change' in the learner, and further argues that model-building offers the potential for cultural change in education [Ril07]. Jonassen demonstrates on a primitive level that modelling environments are tools for 'conceptual change' that can bring about significant learning [Jon06]. In this scheme a wide variety of computer-based tools can be used for learning, such as spreadsheets, databases, and concept maps, so long as there is an element of model construction. Jonassen's main justification for this thesis is that 'if we cannot construct a model then we do not understand it' [Jon06].

A criticism of constructionism and learning by doing is that if a child is allowed the freedom to build whatever they like then there is no guarantee that the learner will engage with the material that is required by the curriculum. This is evidence that constructionism alone is not a solution to bridging the gap between education and educational technology. Thus these eight significant characteristics of learning offer a holistic approach for thinking about education that is well-aligned to computer-based model-building.

1.2.2 Learning involves an active construction of understanding

The idea that a learner *actively constructs their own understanding* is partly represented in the old adage "you can lead a horse to water but you cannot make him drink", meaning that a teacher can provide information and demonstrate skills but the student must be active to develop an understanding so that they can benefit from the teaching. Bruner, an influential cognitive psychologist, suggests that learners who actively engage with the domain are more likely to be able to recall information and apply the understanding in different contexts or to new domains [Bru66]. The following quote from Bruner captures this characteristic of learning:

> "To instruct someone... is not a matter of getting him to commit results to mind. Rather, it is to teach him to participate in the process that makes possible the establishment of knowledge. We teach a subject not to produce little living libraries on that subject, but rather to get a student to think mathematically for himself, to consider matters as a historian does, to take part in the process of knowledge-getting. Knowing is a process not a product." [Bru66:p72]

Swiss biologist and psychologist Piaget first described what later became known as constructivism in his theory of cognitive development, proposing that the world is not full of latent knowledge ready to be gleened, but that learners construct understanding for themselves [Pia71]. Knowledge, or understanding, is built up through experiences and intelligence is shaped by experience. Piaget describes two processes that occur whenever an experience occurs: assimilation and accommodation [Pia71]. Accommodation is the process of accommodating a concept in the mind to an experience in the world [Pia71]. An experience changes previous understanding of things (changing an idea in the mind). Assimilation is the process of assimilating an experience in the world to a concept in the mind [Pia71]. In other words, an experience is 'squeezed' to fit in with previous experiences and understanding of things (thus reinforcing an idea in the mind).

Piaget's later work began to address problems in education, and he was critical of school as a means of leading a child "to resemble the typical adult of his society" [Bri80:p132]. Instead, he suggested that education should be about "making creators... You have to make inventors, innovators, not conformists" [Bri80:p132]. Thus supporting learning as an active process.

Further support for learning as active construction is found in other conceptions of learning such as given by the Austrian philosopher of science Popper [Pop72]. Educationalists Swann and Burgess argue that Popper's learning theory can be of benefit in todays' education systems: "the educator with a prescribed learning agenda must also come to terms with the fact that there is no direct transfer of ideas from her to the would-be learners. Therefore, the would-be learners still need the opportunity to engage in trial and error-elimination, and they must have the will to do so." [SB05:p15] Even with a behaviourist conception of learning, such as given by Skinner [Ski74], active learning is encouraged because the learner must actively engage with a behaviour (or repetitions of a behaviour) in order to strengthen (or weaken) a skill, understanding or behaviour.

1.2.3 Learning results from realising the unknown

This characteristic recognises that learning is often random and can take place when and where it is least expected. As expressed in the quote by A.A. Milne, author of Winnie the Pooh, "One of the advantages of being disorderly is that one is constantly making exciting discoveries." In terms of scientific discoveries, when performing experiments it is not the elements that are understood that are of interest but the phenomena that are surprising and do not fit the hypothesis. However, it is essential to exercise the predictable patterns of agency in order to *realise the unknown*. It is when the unknown aspects of understanding are realised that some significant discovery (or learning) can take place. In a study of model-building in humanities [McC03], Willard McCarty uses the word 'residue' to describe the unresolved—but useful—issues that arise from building models and making formalisations†. It is the bit that is left over, the *residue*, that provides valuable learning experiences. These sentiments are expressed by the American educational theorist Kolb in his account of experiential learning [Kol84]:

> "I move through my daily round of tasks and meetings with a fair sense of what the issues are, of what others are saying and thinking, and with ideas about what actions to take. Yet I am occasionally upended by unforeseen circumstances, miscommunications, and dreadful miscalculations. It is in this interplay between expectation and experience that learning occurs.' [Kol84:p28]

† "modelling treats the ill-fitting residue of formalization as meaningfully problematic and problematizing" [McC03]

William James, a philosopher who spoke widely on the subject of experience and whose work shall become more familiar in the course of this thesis, said that there are two ways in which we take new ideas on-board from others: when we hear a new idea, either it fits in with our previous understanding or it contradicts our previous experience [Jam92]. Sometimes the idea is too far removed from our existing experiences (we have nothing to compare or associate with it) and then, as long as it is from a credible source, it is generally accepted [Jam92]. This type of learning is without first-hand experience of the idea that is being learned (cf. learning by rote). Learning which involves examining objects, phenomena, and situations for one's self—an experiential form of learning in line with activity as described in Kolb's learning cycle [Kol84]—leads to personal understanding because it is experienced experimentally first-hand. Just as Plato said, "knowledge will not come from teaching but from questioning" [Pla55], so too it is an important characteristic of learning that the residue (the unknown) is realised and examined.

The idea that the residue is where the learning occurs further relates to Popper's theory, as explained by Swann & Burgess [SB05]. Popper's account of learning is described in terms of creative imagination—the process of taking a problem, forming a trial solution, and then observing the error in the solution, leading on to another problem [SB05]. Following from this, it is explained that the educators role is "to encourage would-be learners to engage in open-ended trial and error-elimination". This is in order "to identify mismatches between their current expectations and experience" [SB05].

In his book *Learning to Learn*, Novak states that "meaningful learning involves the assimilation of new concepts and propositions into existing cognitive structures." [Nov84:p7]. Novak's work is based on the assimilation theory of Ausubel, a psychologist and follower of Piaget, who stressed the importance of prior knowledge in being able to learn new concepts: "The most important single factor influencing learning is what the learner already knows. Ascertain this and teach accordingly." [Aus68]. Therefore this characteristic is not solely about going in search of the unknown, but also the need to look at what is already known, in order to realise the unknown. Vygotsky's Zone of Proximal Development [Vyg78] is a way to describe the 'current' residue, for it is the perception of 'stuff' on the boundary of the learner's attention that is not yet understood.

1.2.4 Learning need not follow a preconceived path

Learning can be spontaneous in the same way that our experience is. Learning can be altogether unstructured. At any one moment there is no knowing what the next moment will bring. There is no knowing what subject will take our attention, what questions will be raised in our mind, what personal feelings will arise. Neither the teacher, nor the learner himself, can predict what path their learning will take.

Contrast the above statement with the reality of schools and education systems. The curriculum not only sets out what should be learnt, but the order and timing of what is to be learnt. Every student is expected to follow the exact same pattern, the same preconceived path. It seems clear that an experimental approach to learning, as expounded in §1.2.1–§1.2.3, cannot be aligned to a traditional approach to education. The need for open-ended and flexible experimentation is essential.

Kolb, in his introduction to experiential learning, says that human beings are special in their ability to identify and adapt to change [Kol84:p1], just as Darwin claims in his most famous work, *On the Origin of Species*: "It is not the strongest of the species that survive, nor the most intelligent, but the ones most responsive to change" [Dar59]. For Kolb, "learning is *the* major process of human adaptation" [Kol84:p32]. Therefore the aim of traditional education, defined by Dewey as the acquisition of the essentially static knowledge incorporated in books and the heads of elders [Dew59:p5], is contradictory to experiential learning [Kol84:p32]. Kolb's thinking resonates with Rogers' observation of his own education that significant learning occurs when we are most open to change [Rog61].

It should perhaps be made clear that Kolb or Dewey or Rogers are not advocating that the teacher is redundant and that the student should have complete control of their learning. As Dewey points out, "on the contrary, basing education upon personal experience may mean more multiplied and more intimate contacts between the mature and the immature than ever existed in the traditional school, and consequently more, rather than less, guidance by others." [Dew59:p8]

The characteristic of learning drawn from this section is that learning *need not follow a preconceived path*, that a process of adaptation should be respected for significant learning to occur.

1.2.5 Learning can occur without prescribed outcomes

It is Dewey who points out that traditional education is concerned with learning that has a prescribed outcome: "The subject-matter of education consists of bodies of information and of skills that have been worked out in the past; therefore, the chief business of the school is to transmit them to the new generation" [Dew59:p2]. This, in Dewey's opinion, is the wrong view for an educational system that is supposed to be conducive to developing a democratic society. In a traditional system of education the outcomes are prescribed by a higher authority and forced upon the student. Exploration, self-discovery, and personal learning are not encouraged in such a system unless they are within the confines of the prescribed.

Rogers, in his personal account of education [Rog61], takes a strong position against prescribed outcomes: "It seems to me that anything that can be taught to another is relatively inconsequential, and has little or no significant influence on behavior". Rogers' use of the word 'significant' reflects the importance of learning being linked to changing behaviour and thus that 'significant learning' is what is important for education. It is this type of learning that occurs without a prescribed outcome: "I have come to feel that the only learning which significantly influences behavior is self-discovered, self-appropriated learning." [Rog61].

Kolb builds on Dewey's idea for education without prescribed outcomes. Kolb says that the emphasis on the activity not the outcome is what distinguishes experiential learning theory from traditional education and behavioural theories [Kol84:p26]. As discussed in §1.2.4, Kolb talks of learning as a process where the emphasis is on adaptation and not content or outcomes. Thus, this characteristic of learning moves away from a view of knowledge as certain and to-be-received, to a view of learning as open-ended and flexible.

A caution by Dewey on the characteristic of learning as not having prescribed outcomes is that complete ignorance of outcomes may also not be the correct approach (if morality is ignored for example) [Dew59:p17]. Thus, this characteristic is described as 'learning *can* occur without prescribed outcomes'.

1.2.6 Learning is motivated by personal interest

Most teachers would agree that it is easier to teach a subject the student is interested in than one that the student is not interested in. This phenomena is not only found in education, but at a more primitive level, as James describes in *Psychology*: "Consciousness is always interested more in one part of its object than in another, and welcomes and rejects, or chooses, all the while it thinks." [Jam92:p170]. It is part of the human condition that we select to investigate that which interests us most. What interests us is very much a personal preference, no doubt guided by previous experience. Often we choose that which will benefit us or, more often than not, that which gives us pleasure:

> "We dissociate the elements of originally vague totals by attending to them or noticing them alternately, of course. But what determines which element we shall attend to first? There are two immediate and obvious answers: first, our practical or instinctive interests; and second, our aesthetic interests. The dog singles out of any situation its smells, and the horse its sounds, because they may reveal facts of practical moment, and are instinctively exciting to these several creatures. The infant notices the candle-flame or the window, and ignores the rest of the room, because those objects give him a vivid pleasure" [Jam92:p363]

By recognising that we show more enthusiasm for that which interests us (either practically or aesthetically), the implication for education is that either teachers should try to get the students interested in the subject, or teachers should only teach what the student is interested in (what they can relate to their experience). The latter seems more likely to succeed, as suggested by James in a later work aimed at teachers:

> "From all these facts there emerges a very simple abstract program for the teacher to follow in keeping the attention of the child: Begin with the line of his native interests, and offer him objects that have some immediate connection with these" [Jam25:p63]

To appeal to the students' interests it is necessary to look for elements of a subject that relate to the students' previous experience, looking for material that relates to their life. As Dewey points out, this might be something at home or at play:

> "I believe that the school must represent present life—life as real and vital to the child as that which he carries on in the home, in the neighborhood, or on the playground." [Dwo59:p22]

This maxim that learning be motivated by personal interest is not to say that students must only engage with a narrowly defined static set of interests that are relevant to them. Individuality is not static, but is constantly changing, growing and evolving as we encounter new experiences. This type of learning is not only concerned with letting the students learn about what interests them, but also enabling them to learn what it is that interests them—as proposed by Rousseau in his famous account of education through the story of *Emile* [Rou11].

From an ET perspective, Papert acknowledges the need to take on-board the idea that constructionist activities should be personal: "if we can find an honest place for scientific thinking in activities that the child feels are important and personal, we shall open doors to a more coherent, syntonic pattern of learning." [Pap80:p98]. Only when the learner is placed in a position of feeling some identity with scientists, for example, will there be meaningful learning of the scientific material in a curriculum [Pap80].

1.2.7 Learning is a situated experience

This characteristic of learning acknowledges that learning *takes place in a situation, context or culture.* That learning is fundamentally concerned with experience, and experiences occur in a situation, implies learning must be linked to the context in which the experience occurred. Dewey recognises that situations cannot be separated from experiences [Dew59:p42]. The interdependence of a social situation and an individual's experience of it is expressed by Dewey thus:

> "I believe that the individual who is to be educated is a social individual and that society is an organic union of individuals. If we eliminate the social factor from the child we are left only with an abstraction; if we eliminate the individual factor from society, we are left only with an inert and lifeless mass." [Dwo59].

Bruner (introduced in §1.2.2) talks in the same way that learning is always linked to culture: "Learning and thinking are always situated in a cultural setting and always dependent upon the utilization of cultural resources." [Bru96].

Prominent educational theorists Brown, Collins & Duguid [BC89] point out, in their work on situated cognition, that students are often forced to think and to learn about ideas and activities with a context or culture different from where the idea or activity developed: "Unfortunately, students are too often asked to use the tools of a discipline without being able to adopt its culture. To learn to use tools as practitioners use them, a student, like an apprentice, must enter that community and its culture.

Thus, in a significant way, learning is, we believe, a process of enculturation." Brown, Collins & Duguid [BC89] use the idea of 'useful learning being like an apprenticeship' to emphasise that the learning is an activity with a particular context or situation: "the term *apprenticeship* helps to emphasize the centrality of activity in learning and knowledge and highlights the inherently context dependent, situated, and enculturating nature of learning." These ideas relate very closely to Lave and Wenger, best known for their work on situated learning, who see learning as a deepening process of participation in a community of practice [LW91].

Each of these thinkers is developing a metaphor for learning as participation. As discussed by Sfard [Sfa98] whose interests lie in mathematics education, this metaphor can be contrasted to the 'acquisition metaphor' that thinks of learning as acquiring and having knowledge. Such materialistic thinking has been criticised by some authors (e.g. [BC89] [LW91]) as not leading to meaningful learning, whereas the participation metaphor is praised with being linked to the world and our experience of it. Sfard warns that to disregard either of these metaphors completely is a mistake, and in some respects the ideas behind situated learning can be too extreme, just as often the 'acquisition metaphor' is taken too far in traditional education. It has been shown in §1.2.2 that at least one of the significant characteristics of learning takes into account the acquisition of understanding as an active construction process.

1.2.8 Learning is a continuous experience

This, the most primitive of the eight characteristics, acknowledges that learning and education is bound up with the experience. As John Dewey, in the opening chapter of Experience and Education pronounces: "there is an intimate and necessary relation between the processes of actual experience and education" [Dew59:p7]. With this in mind, Dewey recommends the need for a theory of experience in order that "education may be intelligently conducted upon the basis of experience".

The philosopher whose work is perhaps most concerned with a theory of experience, and whom was most influential on Dewey, is William James. At the root of James' principles of psychology is an idea that consciousness arises, and that our experience can be attributed to an awareness of the succession of 'consciousnesses' [Jam92]. James says that this is something that we can confirm by looking at our own experience.

> "The first and foremost concrete fact which every one will affirm to belong to his inner experience is the fact that *consciousness of some sort goes on.* 'States of mind' succeed each other in him." [Jam92:p152]

In his book *Experiencing and the Creation of Meaning*, Gendlin recognises that it is difficult to describe 'experiencing' using language and symbols because it is the fundamental process occurring in our minds at the most basic level [Gen97]. The best he can do is describe situations in which we become aware of our experiencing, and he does this by talking about the concretely present flow of *feeling* or *felt meaning.* Gendlin's informal experiential description serves as a simple illustration of the nature of experience, shared by James' philosophy:

> "First, feel your body. Your body can, of course, be looked at from the outside, but I am asking you to feel it from the inside. There you are. There, as simply put as possible, is your experiencing of this moment, now." [Gen97]

Delving further into our experience, James points out that we can discern that this sequence of 'states of mind' or 'consciousness' is not discrete, it is continuous: neither can it be stopped and started, nor can there be any definite beginning or end to our experiences [Jam92]. Even when we wake up first thing in the morning, the mind is occupied with a continuation of thoughts from the previous day, or possibly a thought resulting from a dreamy state [Jam92]. Hence James likens consciousness or experience to a river that ebbs and flows, always continuously evolving†:

> "Consciousness, then, does not appear to itself chopped up in bits. Such words as 'chain' or 'train' do not describe it fitly as it presents itself in the first instance. It is nothing jointed; it flows. A 'river' or a 'stream' are the metaphors by which it is most naturally described. *In talking of it hereafter, let us call it the stream of thought, of consciousness, or of subjective life.* " [Jam92:p159]

Dewey takes James' principles of psychology into the domain of education by saying that it is the continuous nature of experience that enables us to learn: "the principle of continuity of experience means that every experience both takes up something from those [experiences] which have gone before and modifies in some way the quality of those [experiences] which come after." The continuity of experience also means that learning begins at a very early age. Experiences during the first moments of life have

† Just as if you look at a river, even from the same point, each time the water will be different: it will never be the same twice.

the potential to affect experiences in the future. Vygotsky, famous for his insights into child development, pointed out that by the time a child reaches school age he has already encountered, and potentially learnt from, an exceptionally large array of experiences [Vyg78:p84]. He says, "children begin to study arithmetic in school, but long beforehand they have had some experience with quantity" [Vyg78:p84]. It is wrong to assume that a child attending school for the first time is a 'clean slate'—the sense-making of experience began a long time ago, and those learnings are likely to effect experiences in school.

As Dewey pointed out, the quality of the experience has an important effect on the learning [Dew59:p16]. By the quality of the experience Dewey means the extent to which the experience represents a reliable view of the world. Positive learning occurs when the quality of the experiences is high. But things can be incorrectly learnt when the experience is misleading, of low quality. This leads to the problem in education of providing an environment in which the learners can have quality experiences. Dewey does not say that traditional schooling does not give experiences, but he questions the quality of the experience.

> "Everything depends upon the *quality* of the experience which is had. The quality of any experience has two aspects. There is an immediate aspect of agreeableness or disagreeableness, and there is its influence upon later experiences. ... Hence the central problem of an education based upon experience is to select the kind of present experiences that live fruitfully and creatively in subsequent experiences." [Dew59:p16]

1.3 Implications for technology enhanced learning

The implications for technology enhanced learning that supports these eight significant characteristics can be described in three strands: the *experimental*, the *flexible*, the *meaningful*. Starting from the final characteristic and working backwards, constructing artefacts with ET provides the *experimental* foundation from which to explore *flexible* paths towards *meaningful* learning, as explained below. When these three strands are wound together they form a strong support for ET that can lead to learning in an informal everyday sense.

1.3.1 The experimental strand

The characteristic described in §1.2.1 forms the basis for an experimental approach to learning, whereby ET—in the form of open-ended model-building—can support individual learning. The next characteristic described, §1.2.2, strengthens the experimental aspect by demonstrating that learning involves an active construction process on the part of the individual. Once combined, the active construction of computer-based artefacts leads to an individual considering that which is known in order that the unknown is realised, as described by the characteristic in §1.2.3. These three characteristics taken together form the experimental strand.

The experimental aspect is important in ET as it provides the basic environment from which learners can explore a domain. The experimental basis allows for many of the influences in everyday learning, as in Figure 1.1, to play a part in the learning activity. However, an experimental approach alone is not enough because 'you cannot make something out of nothing'. In order that there are significant implications for learning, it is essential that the approach is also flexible (in terms of the paths and outcomes) and meaningful (as in linked to personal interests, situations and experiences).

1.3.2 The flexible strand

The middle two characteristics of the eight bring an essence of flexibility to the approach to learning. With the characteristic described in §1.2.4 it is stressed that learning need not necessarily follow a certain pre-defined path, but that an individual should be able to find their own path for their own learning. The other characteristic, described in §1.2.5, strengthens the flexible aspect further by relinquishing the idea that learning should have a prescribed outcome.

The implications of a flexible approach are that many of the constraints usually associated with ET (as shown in Figure 1.2) can be forgotten and the learner can enjoy more the freedom of learning in the everyday sense. Learning supported by ET with these two characteristics does not view the outcome as being with the computer or confined to the computer, ET is the support for learning that is flexibly under the control of the learner.

1.3.3 The meaningful strand

The final strand of the eight significant characteristics of learning involves the last three characteristics, and is the most important as it is the foundation for everyday learning. The characteristic described in §1.2.6 is concerned with making learning meaningful by stating that an individual's personal interest plays a part in learning. In the characteristic described in §1.2.7 it is recognised that learning takes place in a particular situation, context and culture which has meaning for the individual. The last characteristic described, §1.2.8, dealt with the basic tenet that learning is a continuous experience which flows with, and has meaning for, the individual. Each of these three characteristics imply learning that is meaningful or relevant for the individual.

ET that supports these three characteristics has significant implications for learning that is meaningful. The meaningful strand supports more of the everyday aspect of learning, shown in Figure 1.1, as it recognises the unique experience, background and interests of the learner, as well as the external and social influences that effect the learners experience.

As highlighted above, ET that respects these eight significant characteristics of learning is appropriate for liberating the experimental, flexible and meaningful aspects of everyday learning. In the next chapter, Empirical Modelling is suggested as a suitable ET for emphasising the eight characteristics and supporting learning that is better aligned to the experimental, flexible and meaningful aspects when compared to traditional ET.

Chapter 2

Empirical Modelling in support of learning

The challenge introduced in Chapter 1 is for educational technology to support more of the everyday aspect of learning that is evident in the *eight significant characteristics of learning*. The following introduction to Empirical Modelling describes the principles, tools and characteristics that are fundamental to Empirical Modelling practice, and which have strong connections with the eight significant characteristics of learning described in Chapter 1.

2.1 Introduction to Empirical Modelling

2.1.1 Modelling state-as-experienced

Empirical Modelling (EM) is a collection of principles and tools that are fundamentally concerned with *modelling state*. In computer science, state is usually associated with the specification of formalised abstract behaviours. This view of state is concerned with procedures for *preconceived* interaction, all of which are *objectively* interpreted with respect to the computer as a 'state machine'. EM is concerned with state in a much broader sense. When referring to modelling state in EM, it means state in its more everyday sense—that is, the condition or status of things as they are *subjectively* and *empirically* apprehended. This type of state, which Beynon refers to as *state-as-experienced* [Bey07a], is open to many kinds of interpretation based on personal observations and experiences.

As an approach to computer-based modelling, EM is not primarily aimed at developing abstract behaviours. EM is concerned with constructing and engaging with concrete situations using computer-based artefacts. To achieve this, EM activity is focussed on creating computer-based artefacts that capture state-as-experienced. These artefacts (or models) typically offer the flexibility of human interaction in the world, in contrast to the rigid tightly-constrained behaviour of a computer program. For this reason, EM artefacts often invoke personal, subjective, particular, provisional and tacit interpretations that reflect the open-ended nature of human interaction.

The construction of computer-based artefacts for modelling state-as-experienced is underpinned by well-established principles defined by Beynon & Russ [EMW]. These principles are predicated on the basis that an artefact is a collection things that can be observed—called *observables*—and that have counterparts in a set of definitions in the computer. Each definition takes the form:

$v \textit{ is } f(x_1, ..., x_n)$

where $x_1, ..., x_n$ correspond to observables, and the value of v is updated instantaneously whenever $x_1, ..., x_n$ change. In this way, a set of definitions can be viewed as representing a *state* together with a family of atomic state-changes. Artefacts are then constructed in a *fluid* activity involving the creation and manipulation of definitions. The act of creating a definition, or making a redefinition, represents a *state-change*. Meanings and relationships develop through interaction with the artefact occurring from state-change. This continuous activity develops a closer and closer correspondence between the artefact and the set of definitions.

To differentiate EM artefacts from more general terms, such as 'models' and 'programs', the term 'construal', as interpreted by David Gooding, has been adopted by Beynon [Bey07a]. Gooding's use of the word 'construal' refers to the artefacts and the interaction with the artefacts that are developed by experimental scientists in the early stages of exploration of phenomena:

> "Construals are a means of interpreting unfamiliar experience and communicating one's trial interpretations. Construals are practical, situational and often concrete. They belong to the pre-verbal context of ostensive practices." [Goo90:p22]; "... a construal cannot be grasped independently of the exploratory behaviour that produces it or the ostensive practices whereby an observer tries to convey it." [Goo90:p88]. (Something *ostensive* is direct or demonstrative [OED:*ostensive*].)

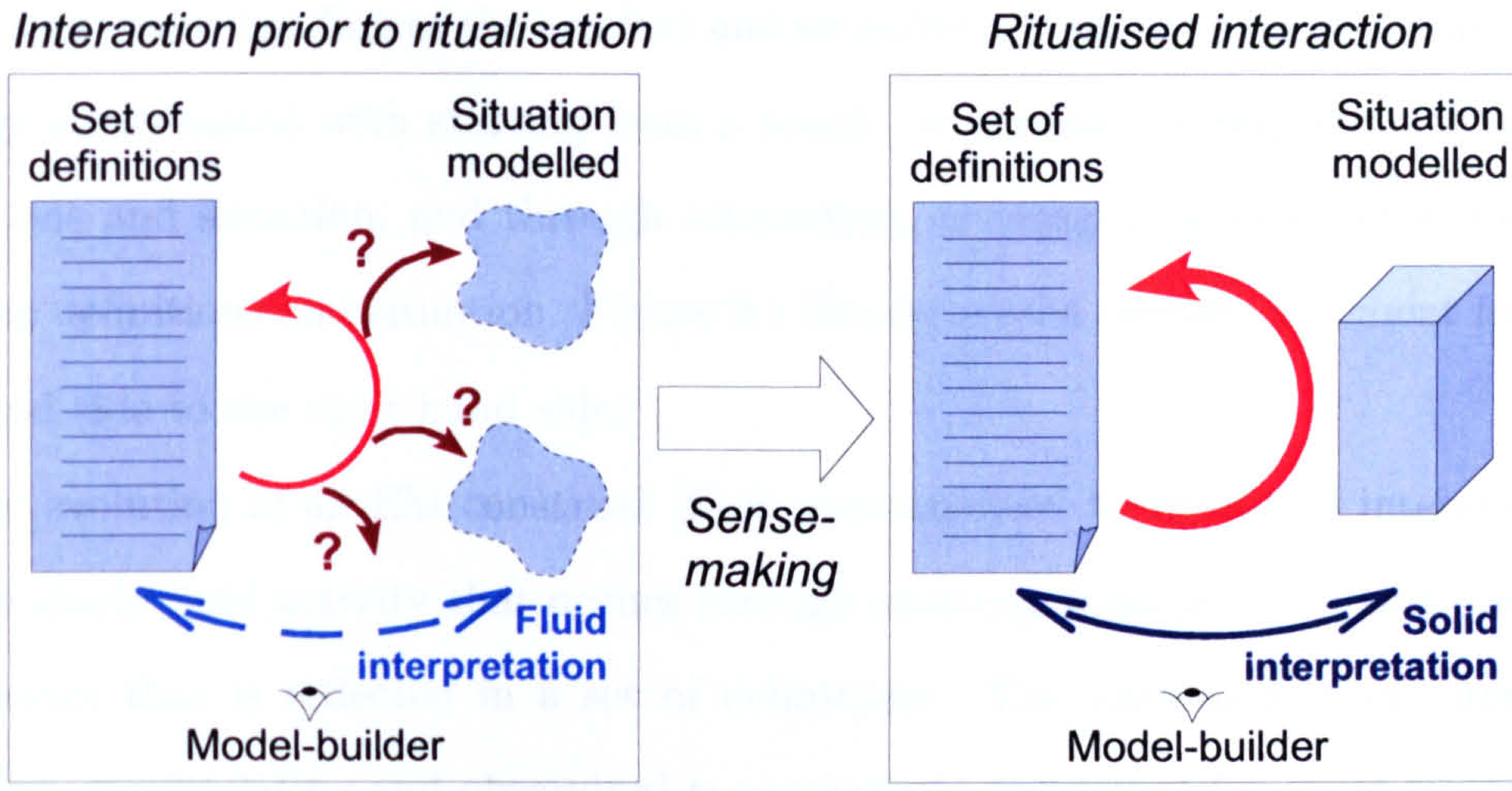


Figure 2.1: The nature of EM activity.

In the same sense, an EM construal can refer to all aspects of an artefact, including the current set of definitions, the history of interactions or redefinitions, and the relationships between experiences with (or expectations of) the artefact and experiences with (or expectations of) the world. Therefore, EM construals are *personal*, *subjective*, *particular to circumstances*, *provisional* and *tacit* [Bey07a].

When construals correspond closely with a situation that is familiar and well understood, we can interpret this to mean that there are well-established patterns of interaction—this will be referred to as *ritualised interaction* because, due to repetition, the interaction with the artefact becomes stereotypical and automatic. Traditional programming is concerned with creating artefacts that support such ritualised interaction. The nature of EM activity means that it is well-suited to interaction that is *prior to ritualisation*. Such interaction is speculative and exploratory, and might involve the negotiating of meaning in situations that are not familiar or well understood. The character of interaction prior to ritualisation is illustrated in the left-hand side of Figure 2.1. In such interaction, it is appropriate for the construal to be personal, subjective, particular to circumstances, provisional and tacit.

The right-hand side of Figure 2.1 illustrates interaction that has been ritualised, where the construal corresponds closely to the situation and this relationship is well understood. This typically means that the key observables in a situation have been clearly identified and have counterparts with fixed interpretations in the construal. Moving from pre-ritualised interaction to ritualised interaction involves becoming familiar and developing understanding of the artefact and situation—it involves sense-making. Such activity is associated with starting from a rough, provisional correspondence between definitions and situation, and through interaction, arriving at a solid correspondence between definitions and situation. Figure 2.1 illustrates the overall movement from the left-hand side to the right-hand side.

The evolution of an EM construal (from pre-ritualised to ritualised interaction) is a particularly fluid activity that occurs through creating, manipulating and observing an artefact that is reflected in a set of definitions. The natural flow of interaction (creating, manipulating and observing) is particularly important for sense-making and learning.

2.1.2 Definition-based notations

The notion of a construal as captured by a set of definitions necessitates methods or notations for creating, manipulating and observing definitions. A number of notations have been developed for a wide range of modelling activities. Beynon refers to these notations as *definitive notations* due to the fact that they are definition-based [Bey07a]. Some notations are general purpose, like DoNaLD (Definitive Notation for Line Drawing) which is used to create line drawings [Yun90]; some notations have a more specific purpose, like the %analog notation created by Charles Care for experimenting with the components of an analogue computer [EMP:analogCare2005].

The primary general-purpose notation is EDEN (Engine for DEfinitive Notations), which was developed by Yun Wai Yung [Yun90]. The EDEN notation is the most primitive and can be used for the definition of base values (e.g. numbers, characters and lists).

The next general-purpose notation is DoNaLD which can be used to create drawings based on points, lines, arcs and other basic shapes [Yun90]. An example of a set of definitions in DoNaLD is shown in Figure 2.2(a). The definitions describe lines and circles that relate to a clock face containing an hour hand, a minute hand, and marks for the quarter positions. The artefact that reflects the set of definitions is shown in Figure 2.2(b).

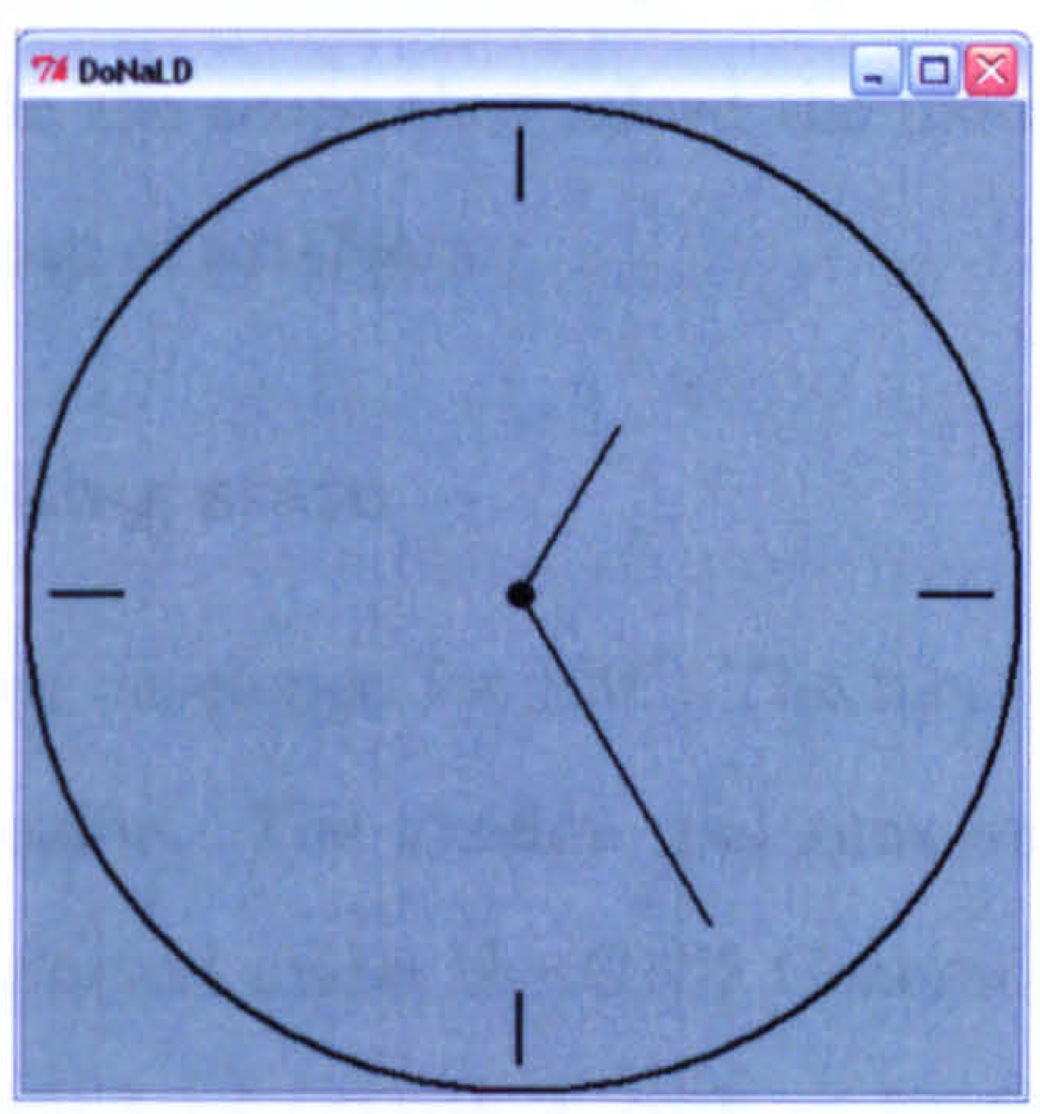
Another general purpose notation is SCOUT (definitive notation for SCreen LayOUT) which was designed by Yun Pai Yung [Yun93] for arranging windows (including

```
int radius
point centre
radius = 400
centre = {400, 600}

line facemark9, facemark6, facemark3, facemark12
circle face, shaft
facemark9 = [centre-{radius-20,0}, centre-{radius-80,0}]
facemark6 = [centre-{0,radius-20}, centre-{0,radius-80}]
facemark3 = [centre+{radius-20,0}, centre+{radius-80,0}]
facemark12 = [centre+{0,radius-20}, centre+{0,radius-80}]
face = circle(centre, radius)
shaft = circle(centre, 10)

int hour, minute
line hourhand, minhand
real hourangle, minangle
hour = 1
minute = 30
hourhand = [centre, centre+{radius div 2.5 @ hourangle}]
hourangle = float(hour) div 12 * -2*pi + 0.5*pi
minangle = float(minute) div 60 *-2*pi + 0.5*pi
minhand = [centre, centre+{radius div 1.3 @ minangle}]
```

(a) DoNaLD definitions describing a clock face.



(b) An EM construal of a clock face.

Figure 2.2: An example of using the DoNaLD definitive notation to create an EM construal.

viewports for DoNaLD drawings) on a screen. A complete list of standard notations can be found in the EDEN documentation [EMD]. It is important to note that these notations are still at an experimental prototype level. There is still a lot to understand about the design of definitive notations, for which there is little precedent in traditional programming language design.

One of the major contributions to the development of definitive notations is a framework for creating and manipulating notations within EDEN—enabling model-builders to develop their own definitive notations (in principle) 'in the flow of modelling'. This is based on my work on an *agent-oriented parser* (AOP) [Har03], that was originally prototyped by Chris Brown [Bro00]. The AOP has led to many innovative models (e.g. the Wumpus model [EMP:wumpusCole2005]) and a number of new definitive notations (e.g. the HTML notation described in Chapter 4). The GEL (Graphical Environment Language) notation [EMP:gelHarfield2006], for creating graphical user interfaces, is one of the notations that has been developed as part of this thesis using the AOP, and that has been used to create and explore many of the models described in this thesis. An introduction to the GEL notation can be found in Figure 4.5 on page 95 and more information is available in the documentation [EMD]. The importance of such auxiliary definitive notations is in the scope they afford the model-builder to exploit richer metaphors in the construction of artefacts.

2.1.3 Tools for modelling state

A number of tools have been developed for EM†. The most widely used and the most extensively developed is `tkeden`. The `tkeden` tool runs on Linux, Mac OS, Unix & Windows and is freely distributed under the GNU General Public License‡. The tool incorporates a number of standard notations, such as EDEN, DoNaLD and SCOUT, as well as other domain-specific notations including those created using the AOP, such as GEL. A significant role for the notations is providing visualisation of state that promotes the experiential rather than the symbolic aspect of the artefact. As illustrated in Figure 2.2(b), visualisations can be expressive without being highly realistic [Bey05b].

Figure 2.3 shows the `tkeden` tool with a simple model of a clock. The bottom left window is the input box where definitions can be entered in a particular definitive nota-

†For a complete discussion of the history of EM tools, see Ward's PhD thesis [War04].
‡The `tkeden` tool is available from: `www.dcs.warwick.ac.uk/modelling/tools`.

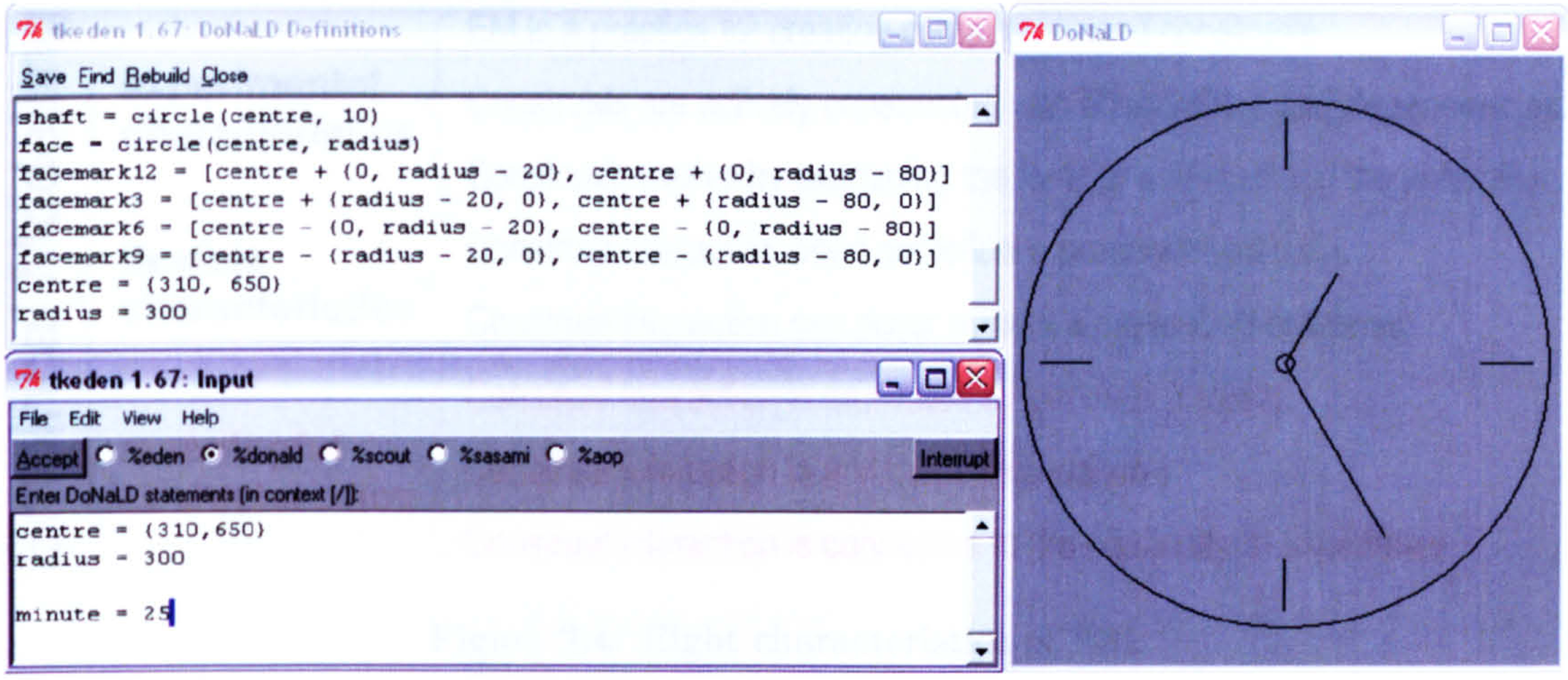


Figure 2.3: The **tkeden** tool for creating EM construals with definitive notations.

tion. In Figure 2.3, the input box currently contains buttons for the EDEN, DoNaLD, SCOUT, SASAMI and AOP notations, but there is potential for new notations, such as GEL, to be introduced on-the-fly. The top left window contains a full list of the current definitions in the environment. The DoNaLD window on the right is the artefact that is described by the DoNaLD definitions. As discussed above, EM construals are created and manipulated through definitive notations and therefore the `tkeden` environment is conceptually relatively simple, requiring only the input box for making definitions and redefinitions.

The key features of the `tkeden` tool are that the interaction between model-builder and artefact is continuous and unconstrained. The fluid nature of EM activity is well accommodated in `tkeden`. Model-building proceeds from a rough, provisional set of definitions that have a loose correspondence to a situation to solid, well-understood set of definitions that have a strong correspondence to a situation. When the definitions of an artefact are recorded or saved, it represents only a snapshot of the EM activity. The `tkeden` tool records the history of redefinitions which affords the possibility of returning to previous significant states.

The `tkeden` tool described above is the most common environment for model-builders. However, there are other variants of the tool. These include an extended version of `tkeden`, called `dtkeden`, with distributed communication features, enabling models to be created across multiple machines. Chapter 5 explores the use of `dtkeden` for collaborative model-building.

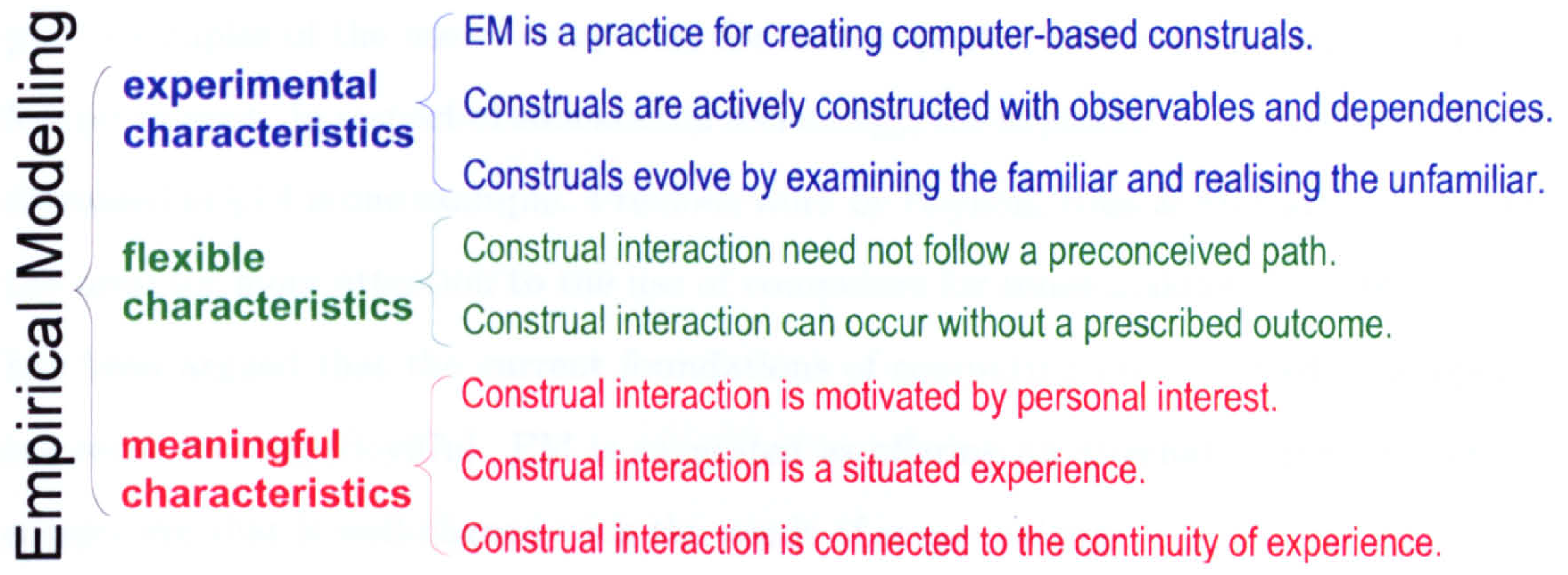


Figure 2.4: Eight characteristics of EM.

2.1.4 More background on EM

EM is the result of 20 years of computer science research led by Beynon & Russ at the University of Warwick [EMW]. The motivations for EM are discussed, for example, in Beynon's lecture notes on modelling for concurrent systems [Bey07a] and in King's thesis [Kin07]. Rungrattanaubol offers a detailed exposition of the principles of EM in its relation to conventional approaches to computing in her thesis entitled, *A treatise on Modelling with definitive scripts* [Run02]. An alternative introduction to EM from an educational perspective is given by Roe [Roe03:p6]. A detailed account of the design and implementation of `tkeden` and other EM tools is given by Ward [War04].

2.2 Eight characteristics of EM

The following discussion introduces EM with particular reference to eight characteristics, summarised in Figure 2.4, that are relevant to the eight significant characteristics of learning set out in Chapter 1. The correspondence between the eight characteristics of EM and the eight significant characteristics of learning can be observed by comparing Figure 2.4 to Figure 1.3 on page 15. The table in Figure 2.8 on page 54 clearly illustrates the connection with references to the relevant section where each characteristic is discussed.

2.2.1 EM is a practice for creating computer-based construals

In a modern world where it is common for people to use computers to create documents, presentations, graphics, websites, and programs—*to produce some readily useful out-*

put—examples of the use of computers for understanding and sense-making are much less prominent. In respect of educational technology, the exploratory use of spreadsheets discussed in §3.4 is one example. Previous work by Beynon, Russ & McCarty has shown the need for more attention to the use of computers for sense-making [BRM06], and it has been argued that the current foundations of computing are ill-suited to supporting sense-making [Bey07a]. EM is suggested as offering an alternative perspective on computing that is well-aligned with the needs of sense-making.

Empirical Modelling, as the name suggests, is concerned with *creating* and *using* computer models that are empirically developed. The word *empirical* can have several meanings, but here it is taken to mean that the modelling activity is *guided by practical experience, not theory.* EM was originally developed as a way of representing concurrent systems. Not the abstract formal models of concurrency associated with, for example, Hoare's CSP [Hoa85], but concurrent systems in a broader sense as found in everyday experience that include people, nature, and constructed artefacts together [Bey07a]. Beynon refers to this as 'common-sense concurrency' and views EM as a means for representing an "external observer's conception of a concurrent system, as it evolves, typically incrementally, through experience of the system" [Bey07a]. Constructing models that reflect common-sense concurrency can be beneficial where sense-making is important, such as: determining the requirements for a piece of software, reconstructing historic events, analysing a archaeological dig, designing a socially-aware robot, or learning to speak a foreign language.

Models or artefacts that are constructed using EM can have a number of characteristics (as explained in this chapter) that differentiate them from uses of models in computer science and, more generally, computing. The word *construal* is used to describe the computer-based artefact that the EM model-builder constructs to avoid confusion with the more general term model. A construal is a computer-based artefact created or used by a person engaging in EM activity. An EM activity is one where the emphasis is on using the construal for *understanding* and *sense-making*, as opposed to necessarily producing a useable artefact. That which the construal relates to in the world is called the *referent* (coming from 'that which is referred to') as depicted in Figure 2.5. The construal is built with some experience to be explored and better understood in mind. The use of the word construal emphasises that the model is

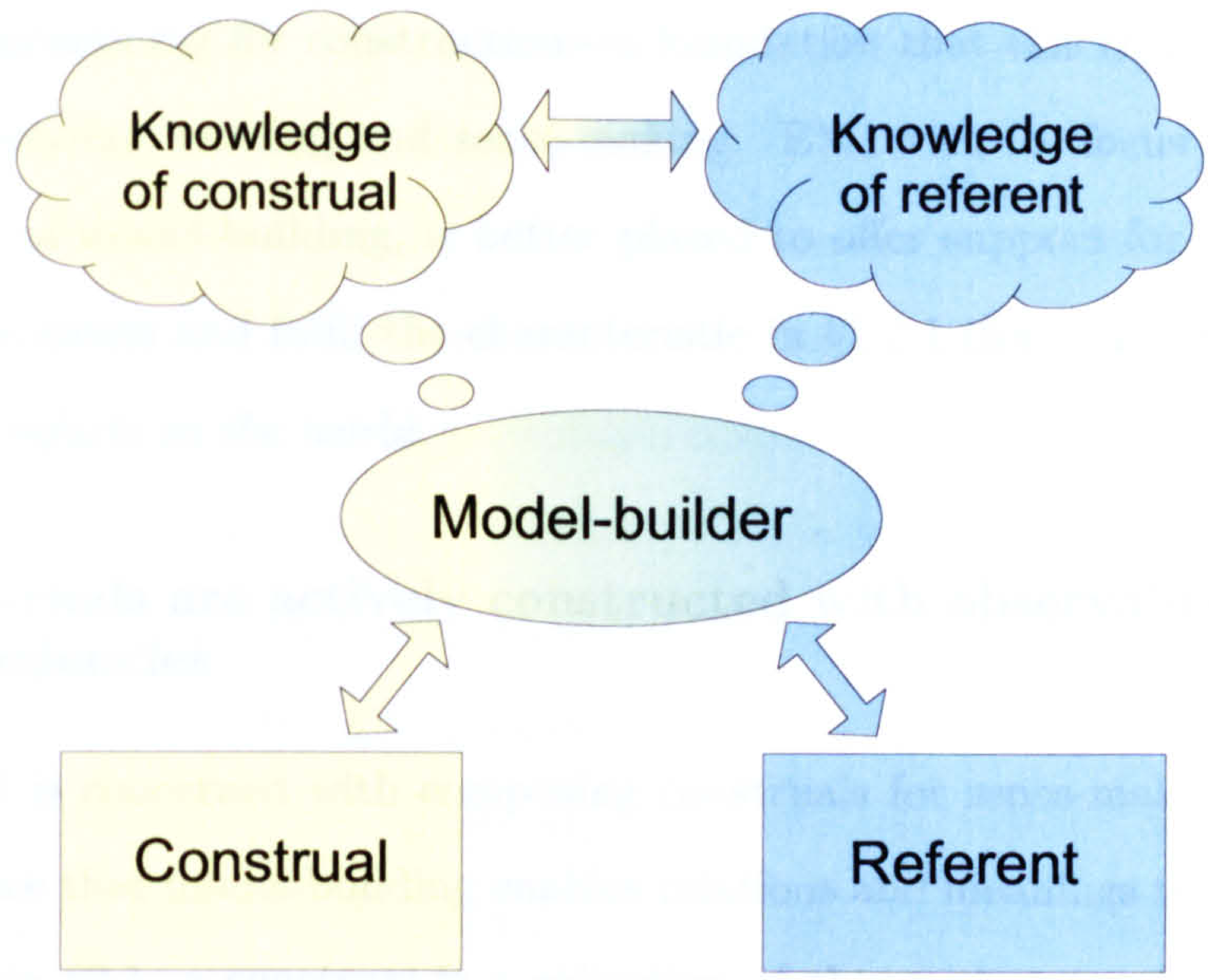


Figure 2.5: A learner constructing an construal relating to a referent.

a personal interpretation of the experience by the model-builder. The experience to which the construal itself refers may relate to a situation, to an abstract procedure, or to a phenomenon. Therefore the referent could be something physical, or the referent could be an emotion or idea to be conveyed in an construal. The learner develops tacit knowledge of the construal and referent through exploratory interaction motivated by establishing a close correspondence between experience of the construal and experience of the referent. The bottom half of Figure 2.5 highlights the essential elements of an EM activity where a model-builder is interacting with a construal that corresponds to a referent. In this activity the model-builder's understanding and experience play a crucial role. Experiences of the interplay between the referent in the world and the construal in the computer, at the top of Figure 2.5, can inform the model-builder's interaction with and construction of the construal.

EM's approach to learning is constructionist in spirit [BR04] [BH05b]. In contrast to traditional ET that exploits a constructionist idiom which is primarily concerned with artefact construction, ET based on EM principles promotes constructionist activity that is essentially concerned with negotiating meaning and sense-making. Well-known examples of constructionist learning environments include Logo, Agentsheets and Toontalk. These are discussed in contrast to EM by Roe [Roe03]. In particular, Roe points out that Toontalk and Agentsheets rely too heavily on the computation metaphor [Roe03]. The limitation of these environments is that they use methods based on traditional procedural programming for construction—a foundation that this thesis has argued is ill-suited to everyday learning and sense-making. EM, with its focus on the experimental aspects of model-building, is better placed to offer support for constructionist learning environments and fulfil the characteristic in §1.2.1 that *learning occurs when constructing artefacts in the world.*

2.2.2 Construals are actively constructed with observables and dependencies

Given that EM is concerned with composing construals for sense-making, it is of primary importance that model-building enables relations and meanings to be negotiated. As introduced in §2.1, a construal is a collection of things that can be observed and manipulated—called *observables.* These observables can be represented by a set of definitions which describe the relationships between observables using *dependency.* The observation and manipulation of definitions is occurs through *agency.* Observables, dependencies and agency reflected in a construal can capture the scope of an external observer's interpretation of a concurrent system in an everyday sense [Bey07a].

From the viewpoint of an external observer, observables are features of the environment that can be ascribed an identity [Bey07a]. The identification of observables arises from interaction with a referent, as in Figure 2.5. An observable could represent a value (e.g. 360), a property (e.g. has wheels), a quantity (e.g. £125,000), a description (e.g. sleek), a colour (e.g. red), a relative measurement (e.g. fast), a perceived feeling (e.g. scary), or an event (e.g. I saw a Ferrari). Observables are often subjective as they reflect a personal meaning for the model-builder that has developed over a series of interactions. Current EM tools are implemented on a digital computer and therefore the representation of observables are rather primitive, however the intended meanings of such observables can reflect quite detailed ideas in the mind of the observer. The level of detail of an observable depends very much on the observer and the motivation for studying a given referent. Depending on the interests and skills of the observer, observations may be made on different levels (e.g. "the meteorite is the source of the light, and the meteor is just what we see").

The external observer, during the course of many interactions with an everyday concurrent system, is likely to develop expectations with respect to observations. For example, I am accustomed to associate the swiping of my university card at the door of the department with the opening of the door. These indivisibly perceived changes, where the observer would be surprised if expectations were unrealised, can be described as *dependencies* [Bey07a]. Such expectations, or dependencies, reflect reliable patterns of state-changes arising from a series of interactions. Thus, a dependency is a description of how changes to observables are linked to one another. In everyday concurrency, observations are related to each other in that a change in one observable often leads to a change in other observables. The perception of dark clouds overhead often coincides with the subsequent falling of rain (event), which often coincides with cars switching their headlamps on (description) and, if driving, a sense of caution on the road (feeling). Each of these dependencies may be reliable observations that have been made according to history of interactions. A point to be discussed later is that these dependencies are not 'set in stone'—at any time during the model-building, maybe in response to new observations, the relationships may be changed.

In EM, *agency* is associated with the attribution of state-change [Bey07a]. Whereas concurrent specification languages often ignore the notion of an agent, in an everyday view of concurrency it is natural to attribute the change of state to an object or entity [Bey07a] (e.g. "candy weighing both of my pockets down"). Changes to observables can occur in any number of ways. For this reason, EM takes a very liberal view of what constitutes *an agent*, because an agent is anything that has *the capacity to change state.* In everyday living, people have the potential to (and regularly do) change the state of the world. People regularly change state as they move about their daily business, they can choose to switch the heating on or off, to place rubbish in a litter bin, to speak out against something or someone, or to keep quiet and do nothing—every action has an effect on state[†]. Animals too have this capacity to affect the world, as do weather systems, plants, bacteria, diseases and many other things. In many cases we do not understand why these effects take place (e.g. freak weather like the Tsunami caused by the Indian Ocean earthquake in 2004). It may turn out that through subsequent iterations with the referent that acts of agency could be observed as complicated dependencies at a more primitive (e.g. atomic) level. However, from

[†]As reflected in the Buddha's description of Kamma: "When this is, that is. From the arising of this comes the arising of that. When this isn't, that isn't. From the cessation of this comes the cessation of that." AN 10.92 (Pali Canon)

the point of view of an external observer, there will often be acts of state-change whose origin are unknown or not important at the current time. For example, my grandmother might perform perceived random acts of gratitude, such as giving me chocolates, where the causes are not relevant to me. Take the weather as another example of agency, and its forecasting: studied since Robert Fitzroy headed the first meteorology department in the British government in 1854 [Bur86], still today, it is often not even possible to explain the causes of freak weather, let alone predict the weather accurately on a daily basis. It is important to acknowledged that common-sense concurrency is not a closed system but is open to agency from the outside (e.g. from people or the weather).

Observables, dependencies and agency (ODA) are the three concepts in EM that can be used to represent concurrent systems in an everyday sense. An EM tool, such as `tkeden`, provides an environment in which a model-builder can experiment with patterns of ODA. The construction of a construal is achieved by creating and manipulating sets of definitions that correspond to ODA. In `tkeden`, observables can be identified as values, strings, or lists (for example)—like the cells in a spreadsheet. Dependencies, being the relationship between observables, can be identified as definitions that relate two or more observables. A dependency definition indicates an expectation, that when one observable changes, a change also occurs in the other observable—like functions between spreadsheet cells. When considering agency, it may come from outside the computer-artefact in the form of mouse movement, mouse button clicks, keyboard presses, or input from other devices. These actions could be automated or semi-automated by the model-builder in such a way that they become *agent actions* within the model thus creating internal agency. Agent actions can be defined in a model as a sequence of redefinitions that are triggered when a condition occurs (i.e. an internal observation is made). If we take the example of modelling the driving of a car, then we might start by controlling the car with external agency by letting the model-builder control the inputs to the car driving activity (i.e. steering, accelerating and braking). The model-builder could use the patterns of redefinitions that she makes to automate or semi-automate the driving activity. Initially these patterns may be quite simple, for example: "when approaching a corner, apply the brake". Further refining of the model might take into account many more factors that could affect the agent's control of the car (e.g. weather conditions, the position of other cars on the road, the driver's knowledge of the car, or the driver's mood)†.

Numerous notations exist for constructing construals based on ODA. As introduced earlier, EDEN, SCOUT and DoNaLD [EMD] are three basic notations that are used in the tkeden tool. Detailed explanations of these and other notations used for creating sets of definitions corresponding to ODA are covered by Ward [War04]. Some of the notations developed by the author are discussed in Chapter 4.

The ODA that are reflected in a construal correspond to ODA in the concurrent system through the model-builder's continual interaction with the construal. ODA are essential ingredients for model-building because of the close correspondence between the ODA reflected in the constual and the ODA in the referent. In Figure 2.5, the construal is linked to the referent by this correspondence through ODA. In this way, interaction with ODA in the construal develops knowledge of the construal which is linked to knowledge of ODA in the world or referent. This characteristic of EM as an active construction of a construal using ODA on the part of the model-builder is related to the characteristic of *learning involving an active construction of understanding* on the part of the learner described in §1.2.2.

2.2.3 Construals evolve by examining the familiar and realising the unfamiliar

Given an environment for constructing construals that reflect observables, dependencies and agency in an everyday sense, EM's contribution to the construal creation process is now considered. In EM, as with any other sense-making activity, we usually start with that which we already know, have an understanding of, or are familiar with. By examining the familiar, we are able to see what we do know about the subject, and what we do not know. In some cases, our subject might be very well known, but in others we might be exploring it for the very first time. An inexperienced architect may well explore his subject, the design of a library for example, in some confusion and with no previous experience of making a plan of a library to draw upon. Generally though, the architect is not completely lost as he has other experiences that might be relevant: his training as an architect, experience from designing other architectural plans, visits to his local library. A more experienced architect may well have designed many libraries,

†A model about braking distances was created as part of an undergraduate project for WEB-EM-3 [WEBEM3].

knowing exactly what is required of him and his design, but still by exercising the familiar he will come across areas of the design that he needs to make sense of, and make a decision, to continue his design. This act of sense-making, by exercising the familiar to explore the unfamiliar, is what we are 'making use of', encouraging and enhancing in EM through the use and creation of construals.

When creating a new construal, I am encouraged to start with what I already know or something I am familiar with. If I know the subject well, then it is relatively easy to highlight observables, dependency and agency, and therefore I can begin to build an EM construal by defining various observables and dependencies, and maybe later automating some aspects of agency. If for example, I started to create a construal of a bicycle, I might begin with a simple definition taking account of the wheels, their sizes, and the number of gears. Depending how well I know the subject, within a short while I would reach the edge of my understanding. This is what Vygotsky would call the Zone of Proximal Development [Vyg78] as mentioned in §1.2.3. On my bicycle, although I might be aware that the more force I exert on the pedals the faster I will go, I cannot immediately formalise this into a more accurate form such as the relationship between force and velocity. I have a vague idea of the relationship, but it remains unfamiliar territory. This is the point at which I move from creating a construal of a familiar subject using ODA, to exploring patterns of ODA in my construal that fit the unfamiliar territory (that which is not formally known to me). The unfamiliar is no longer a complete blank because I have the springboard of the familiar ODA on which to base my experiments with other patterns of ODA. I can also draw on other experiences, or other people, or other models, to compare to my own experiences with the construal. By exploring and experimenting with different patterns of ODA, and comparing it to my familiar understanding and experiences from the world, I am usually able to gain a little more familiarity of the subject. In order to become more familiar with the bicycle construal, I can experiment with different dependencies and draw on my experience of riding a bicycle to make sense of the relationship between the pedals and the speed of the wheels.

The process of acknowledging the familiar and exploring the unfamiliar is not a one-off exercise that will result in the subject being understood. (No results are guaranteed.) Rather, it is a process that continually occurs throughout the creation of a construal

1. Initially I do not know what is unfamiliar or not known to me.
2. I start by creating a construal using ODA of that which I am familiar.
3. By exercising the familiar, I find the edge of my familiar understanding and discover that there are things which are unfamiliar or not known to me.
4. I explore and experiment with different patterns or compositions of ODA.
5. By relating experiences of the construal with experiences in the world, I can become more familiar and make-sense of the subject.
6. I repeat the process, exercising the familiar including any new aspects of the construal.

Figure 2.6: A generalisation of experimental EM activity.

and throughout the sense-making activity. Neither the familiar nor the unfamiliar can be separated, they both depend on one another for the sense-making activity to proceed as shown in Figure 2.6. The activity in Figure 2.6 is a continuous cycle in which the familiar informs the unfamiliar and the unfamiliar forms the familiar.

Schrage, who discusses models in his work on innovation [Sch99], writes in *Serious Play* that:

> "...the real value of a model or simulation may stem less from its ability to test a hypothesis than from its power to generate useful surprise. Louis Pasteur once remarked that 'chance favors the prepared mind.' It holds equally true that chance favors the prepared prototype: models and simulations can and should be media to create and capture surprise and serendipity [...] That's why Alexander Fleming recognized the importance of a mould on an agar plate and discovered penicillin." [Sch99:p117,119,125]

Figure 2.6 taken with Schrage's sentiments shows a clear connection with learning in relation to the 'realising the residue' characteristic of learning in §1.2.3. The first three of the characteristics of EM emphasise the *experimental* nature of EM activity that plays an important role in supporting learning as characterised in Chapter 1.

2.2.4 Construal interaction need not follow a preconceived path

In this section it will be shown that construals for sense-making involving the exploration of the unfamiliar (as described in the previous section) should be approached without a preconceived plan. Computer artefacts that are thought-out or preconceived in advance do not encourage the sense-making activity during the creation of the artefact. The common practice of computer scientists dictates that computer artefacts should first be specified, then designed, and finally implemented. This implies that the sense-making or understanding is done at the beginning, and the final stage is a 'simple' translation of a design on paper to a program in the computer. It disregards the essential need for sense-making in the implementation stage. There are some good reasons for disregarding sense-making when the final product is of primary importance, but when the focus is on using the computer for understanding a subject, following a preconceived path can subvert the sense-making activity. Therefore, in EM it is important that construal interaction does not necessarily follow a preconceived path.

Neither is there a particular way in which EM construals have to be created (no given recipe) nor does the model-builder have to preconceive of a way to create her construal (no need to design a recipe). There is no need for a recipe for creating a construal, the recipe arises from the cooking. The cooking being the activity of creating a construal using ODA by paying attention to the familiar and the unfamiliar.

Traditional programming is like making a victoria sponge; follow the recipe and, depending on the ingredients and your skills, you will end up with a cake. EM does not offer any particular advantages for following a recipe because the path is preconceived and well-understood. However, EM is appropriate when you want to find out what makes a good victoria sponge, when you want to understand how to make a good victoria sponge, and when you want to make sense of the relationship between the ingredients and a good victoria sponge. This activity does not follow a preconceived path, it requires the model-builder to practice what is already familiar about making a victoria sponge, experiment with different patterns of ingredients (ODA), and relate new 'victoria sponge' experiences to previous baking and tasting experiences.

As EM activity is concerned with interaction that is often *prior to ritualisation*, as depicted in Figure 2.1, it is well-matched to the characteristic, described in §1.2.4, that *learning need not follow a preconceived path.*

2.2.5 Construal interaction can occur without a prescribed outcome

A construal is never considered finished because it has no prescribed outcome, in the same way that learning is never finished (as described in §1.2.4)—there is always more to 'make sense of'.

Following on from the characteristic of EM that construal interaction need not follow a preconceived path, a further step can be taken to describe that it is not even necessary for a construal to ever be considered finished. This theme is partly developed in Beynon's paper on *Liberating the computer arts* [Bey01]. The reason for this is self-evident from the description of the sense-making activity. Making sense of a subject is a never-ending process of continually expanding understanding into ever more unfamiliar territory. A construal being a computer artefact for making sense of a subject, it follows that the activity is open-ended and the construal need not be constrained to achieve any particular result. Contrast this with traditional programming where the expected output is a program that can be used by others, and there is clearly a need for the program to, at some point, be considered finished, useable or sellable. In order to produce a finished product, it makes sense that the product be specified or thought-out prior to the programming activity in such a way that the programming activity has a clear prescribed outcome. The specification then becomes the guide by which the programmer knows what a finished product should look like, and hence they will also know when it is finished. As described in the previous section, it is likely the programmer will employ some standard methods to assist in achieving the finished product (e.g. object-oriented programming) following some particular development methodology as discussed later in §3.1.

Although a construal may never be considered finished, it does not mean that it is not useable. It can be useable throughout the interaction, simply because it is the experience through interaction that enables the model-builder to develop understanding. It is not the finished product itself which is most important, it is the meaning that the construal can give, or has given, to the model-builder. Therefore the construal does not require any specification, nor does the construal need to be thought out in advance. The 'thinking' of how to create the construal is part of the sense-making activity. Any pre-thought-out specification for the construal is liable to restrain and subvert the creation of the construal. This approach of not asking for a prescribed outcome is evident in particular scientific discoveries. For example, when Faraday discovered electricity, he had not set out to find electricity or produce it, he was looking at other phenomena—trying to make sense of them—when he became aware of what later became known as electricity [Goo90].

The above discussion indicates a similarity to the characteristic, described in §1.2.5, that *learning occurs without a prescribed outcome.* Taken together, the two characteristics of EM described as '' and '' promote a particularly flexible attitude to model-building in which the process or outcome can be completely open-ended. These flexible characteristics of EM play an important role in supporting everyday learning which is explored further in Chapter 4.

2.2.6 Construal interaction is motivated by personal interest

In the previous five sections it has been shown that EM is an experimental approach to model-building (§2.2.1,§2.2.2,§2.2.3), and that it is an approach that demands a high degree of flexibility (§2.2.4,§2.2.5). Next, the reasons for creating construals in this way are examined. One of the reasons is that construals are intimately connected to the person interacting with them. They are tools for understanding and making-sense things in the world (as well as in the computer). The fact that they are very personal to the model-builder is why they are useful for sense-making. The meaning of the construal is constructed by the person creating the construal—it is subjective—and the sense-making occurs in the interplay between construal and referent as perceived by the model-builder.

Given the personal nature of construals, it follows that construals may be linked to personal experiences and personal interests. Empirical Modelling is successful when it is motivated by personal experiences and personal interest (as discovered in the 'Introduction to EM' module discussed in §6.1). Creating a construal of something you are not interested in—not meaningful—is unlikely to inspire great exploration or connect well with previous experiences. But with a subject in which the model-builder is personally interested—is meaningful—there is more motivation for exercising the familiar and better potential for exploring unfamiliar territory on the edge of the model-builder's understanding.

As shown in Figure 2.1, EM activity is concerned with the sense-making involved in progressing from interaction prior to ritualisation to ritualised interaction. Such progress involves exploring particular situations in a construal—exploration that is motivated by personal interest in the particular situation. The importance of personal interest in EM activity is relevant to learning, and demonstrates an association with the characteristic, discussed in §1.2.6, that *learning is motivated by personal interest.*

2.2.7 Construal interaction is a situated experience

In the previous section it has been acknowledged that construals are personal tools for understanding and sense-making, and if we stop there then creating construals might be seen as a private solitary activity. Although this act of making sense of a subject is very personal, it need not be totally confined to one model-builder per construal. In this section I shall explain that construals can be shared amongst model-builders, and that construals can be moved in and out of different contexts.

It is helpful to note the similarities between a construal and a story. A construal is an EM artefact that enables the model-builder to convey meaning. A story is similar in that it is used to communicate the meaning in a situation or an event. Neither the construal nor the story are necessarily precise or formal in any way. Both are open to interpretation by either the model-builder in the case of the construal, or the story-teller and listener in the case of the story. Each time a story is recounted, it is a unique explanation of the situation, but sufficiently similar to be recognised when heard again. So it is with construals. A model-builder's construal is unlikely to be the same for every interaction, and in many cases it is desirable for the construal to change, elaborating the details of the construal as the model-builder becomes more familiar with his subject. A construal can be looked at from different contexts, in order to explore other meanings for the construal. A simple example of this is in the jugs model to be examined in §3.2, where liquid in two jugs can have a different meaning in completely unrelated contexts such as the displaying the chords on a violin. A construal, although a personal artefact, can take on new meanings in a different context or situation from that which the model-builder created it, just as a story can invoke new meanings in a different context.

Stories can be taken up by other people, and construals can too. Construals taken up in this way by new model-builders are not necessarily used in exactly the same way as before or in the same context. The evidence that I have of this is that EM construals are often revisited by different model-builders. New model-builders nearly always incorporate the old construal into their own construal, or they use the old construal as a basis for further creation. I relate this to the way that people tend to remember stories that are relevant or interesting to them, that is stories that can be put in a personal context. When a model-builder take on a construal, it needs to be taken into context for it to have some meaning. Sometimes this new meaning is not all that relevant to the previous model-builder's meaning. So a construal, although a personal artefact, can take on new meanings with a different model-builder.

When a story is written down, it is similar to creating automation. In conventional software development this forces the user to 'use' the program exactly as specified by the programmer. Automation allows users to follow a specific line of interaction, but EM model-builders, unlike typical users of programs, do not necessarily have to follow the specific path laid out for them by the automation (they are free to intervene at any point)—just as someone reading a story is free to skip over sections of the story, read it backwards or in any order, as well as to try to interpret the story in whatever way they wish. The distinction between EM and software development is taken up further in Chapter 3.

The emphasis of this characteristic is on the nature of EM activity to treat construals as embedded in a particular situation or context, and that a construal used in a particular context is unique. This is closely connected with the characteristic of learning, as described in §1.2.7, that *learning is a situated experience.*

2.2.8 Construal interaction is connected to the continuity of experience

A construal's association with sense-making requires that experience plays a fundamental role in the creation and use of such artefacts. Of course, experience plays an important role in any interaction between user and computer, but as Beynon, Russ & McCarty show, traditional methods for constructing computer-based artefacts (i.e. programming) attempt to separate the experience from the construction process in a way that EM subverts by the fluid nature of its activity [BRM06]. As illustrated in Figure 2.1, the sense-making that is involved in moving from interaction prior to ritualisation to ritualised interaction is a continuous and unconstrained activity. In this way, EM offers a different perspective on the role experience plays in interactions between model-builder and computer, one that views experience as central to such interactions.

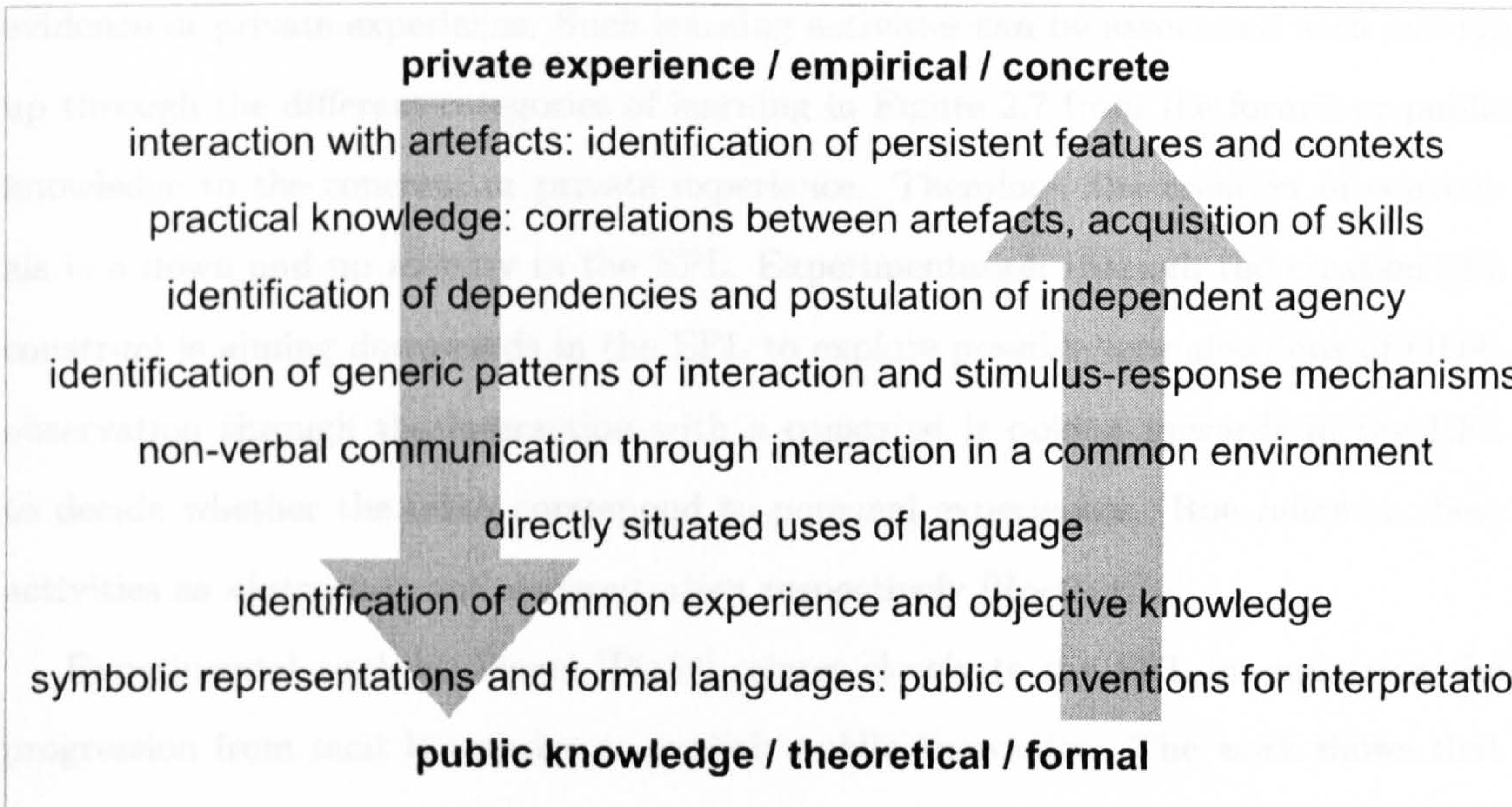


Figure 2.7: The Experiental Framework for Learning (EFL).

As described by Beynon [Bey05a], the EM approach relates to the philosophy of James' in his work on Radical Empiricism [Jam12]. Beynon argues that a construal should support the superabundant and dynamic nature of personal knowledge: "What there is to be known of Coventry is more than I can ever experience, and my personal knowledge is established, maintained and revised dynamically through my ongoing interactions with it" [Bey05a].

The Experiental Framework for Learning (EFL) introduced by Beynon [Bey97] and subsequently elaborated on by Roe [Roe03] describes different categories of learning as shown in Figure 2.7. These categories range from activities concerned with concrete situations and private experience to activities relying on formal languages and public knowledge. Roe states that learning begins from private experience: "Preliminary interactions are informed by our previous experience" *[Roe03:p73]. We start to attribute meaning to elements of a construal* and plant the roots of understanding. After a while we begin to notice patterns of observables, *dependencies and agency that are common* between experiences. As we make more sense of the subject, as we become more familiar with the subject, we are able to explain more clearly the relationships between observables. This corresponds to the downward arrow in Figure 2.7, in that concrete situations and private experience leads towards the formal use of language and public knowledge. At the same time, sense-making necessarily involves checking that a construal corresponds to the referent by comparing a chosen set of definitions to empirical evidence or private experience. Such learning activities can be associated with moving up through the different categories of learning in Figure 2.7 from the formal or public knowledge to the concrete or private experience. Therefore, the creation of construals is a down and up activity in the EFL. Experimentation through the creation of a construal is aiming downwards in the EFL to explore possible formalisations of ODA, observation through the interaction with a construal is poking upwards in the EFL to decide whether the ODA correspond to personal experiences. Roe refers to these activities as *abstraction* and *concretisation* respectively [Roe03:p76].

Experimental work by Piaget [Pia74] relates closely to the EFL in explaining the progression from tacit knowledge to explicit public knowledge. The work shows that there is a development gap between succeeding in performing an action and being capable of explaining the action. Experiments on young children discovered that almost none could describe verbally the movement of their hands and feet when walking on all fours, even after performing an example walk themselves [Pia74:p3]. Older children were able to correctly describe their behaviour. Another experiment [Pia74:p15] with children swinging and launching a ball on a piece of string shows that although competent at the skill of hitting a target, the children did not realise that the ball's trajectory was determined by both the release point and the direction of rotation. These examples (and others) show that children of different ages are not able to describe in language the skills that they have learnt. They have developed a practical level of knowledge (in the upper realm of Figure 2.7) but as yet have not progressed to levels further down the EFL. This is further evidence, in children at least, that learning starts from the realm of private experience in the practical/concrete sense, before moving towards public knowledge in an abstract sense.

Other work in the Geneva school by Karmiloff-Smith and Inhelder [KI75] points to the importance of "constructing and extending theories of action" for discovery in early childhood. This theory construction activity is evident in the downward movement of the EFL where similar or repeated private experiences are charactised in a way that can be more formally explained, such as developing an understanding from repeated experiences of rainbows occurring when bright sunshine follows heavy rain. Karmiloff-Smith and Inhelder also point out that scientists as well as children have a similar tendency to explain phenomena by constructing a unified theory and therefore it may be a deep-rooted function for learning and discovery [KI75].

EM can support the categories of learning in the EFL on all levels as demonstrated by Roe [Roe03]. Furthermore, the fluid nature of EM activity (as depicted in Figure 2.1 means that the learner can move between the levels (moving up and down the EFL) in the stream of model-building. The continuous and unconstrained interaction with a construal offers a model-builder the continuity of experience necessary for learning. In this way, EM is aligned to the characteristic of learning, introduced in §1.2.8, that *learning is a continuous experience.*

2.3 Connections between EM and learning

2.3.1 The common theme of sense-making

In the discussion of the *eight characteristics of EM*, the word 'sense-making' has been introduced. Sense-making is a theme which runs throughout this thesis and is central to the EM principles and tools. It is taken literally to mean the activity of making sense of a situation or phenomena. As Beynon & Russ point out, building models is intimately connected to sense-making [BRM06], as will be demonstrated further in Chapter 3. Furthermore, sense-making is an important aspect of learning [Jon06:p3]. Jonassen relates sense-making to 'conceptual change', introduced in §1.1.4 as the mechanism underlying meaningful learning, and model-building using technology is one way for conceptual change or sense-making to arise in learners [Jon06:p3].

This thesis illustrates how EM can be used to help learners construct models of what they are studying or other phenomena of interest. Building models of phenomena and situations using EM tools facilitates the process of sense-making in learners.

2.3.2 Corresponding characteristics

There is a close correspondence between the eight characteristics of EM described above and the eight significant characteristics of learning as can be observed from Figure 2.8. This correspondence forms the basis for the argument that *EM is a suitable approach for supporting the experimental, flexible and meaningful characteristics of everyday learning.*

The discussion of EM in this chapter illustrates the point that EM is a learning tool

Characteristic of learning described in Chapter 1	*Related EM principle*
learning occurs when constructing in the world (§1.2.1) learning involves the active construction of understanding (§1.2.2) learning results from realising the unknown (§1.2.3)	construals are constructions in the computer (§2.2.1) construals are active constructions of meaning through ODA (§2.2.2) construals evolve by examining the unknown (§2.2.3)
learning need not follow a preconceived path (§1.2.4) learning occurs without prescribing an outcome (§1.2.5)	construal creation need not follow a preconceived path (§2.2.4) a construal is never finished (§2.2.5)
learning is motivated by personal interest (§1.2.6) learning is a situated experience (§1.2.7) learning is a continuous experience (§1.2.8)	construals are personal to the model-builder (§2.2.6) construals are personal to the situation (§2.2.7) construals are closely connected to continuing experience (§2.2.8)

Figure 2.8: The correspondence between EM and the characterisation of learning in Chapter 1.

that people learn *with*, not *from*. Therefore it is particularly well-suited to *supporting* ET with more of the characteristics of learning in the everyday sense—EM is less like Figure 1.2 and more like Figure 1.1. The integrated EM approach outlined in this thesis has advantages for supporting some forms of learning that assume the characteristics laid out in Chapter 1, but not necessarily for other forms of learning. While the following chapters demonstrate how EM helps achieve the eight characteristics set out in Chapter 1, it should also be admitted that EM might not be a suitable approach for all forms of education. It should be acknowledged that the goal is not necessarily to fit in with all forms of existing education, but to alleviate the tensions between technology enhanced learning and everyday learning. If EM can support more of the everyday aspects of learning then there is potential for the tensions to be reduced—in a way that might lead to a new paradigm for learning and education that is expected in the vision for ET discussed in Chapter 1. The next chapter is devoted to showing *why* EM is different to traditional ET, and the following chapters then demonstrate *how* EM is better suited to supporting the eight significant characteristics that are a feature of everyday learning.

Chapter 3

Distinguishing Empirical Modelling from programming

The introduction of EM as an approach to constructing computer-based construals as given in Chapter 2 invites a comparison with programming. A naive view would be that EM is 'just another programming technique', but critical tensions between EM and programming demonstrate that this view is incorrect. This chapter differentiates EM from programming on a fundamental level in order to show that EM offers a completely new approach to constructing computer-based artefacts as set out in Chapter 2. Furthermore, the differences stand out when considering the design and use of educational technology. From a conventional programming perspective, there are typical roles for *student*, *teacher* and *developer* for using, specifying and implementing respectively. In EM, these roles are blended because all interaction is of the same essence. EM activity is more like the use of spreadsheets as discussed in the second half of the chapter.

3.1 Five points of contrast with programming

EM has been developed over a number of years mainly by computer scientists. The tools have similarities with programming tools, and some computer scientists have used the EM tools for programming-like activities. However, the experimental, flexible and meaningful characteristics of EM suggest that it is different from programming. This section highlights five points of contrast, summarised as:

- Programs are more constrained than EM construals;
- Programming entails many discrete phases;

- Programming is concerned with developing an end product;
- Programming is concerned with the correctness of a program;
- Programming makes a distinction between development and use.

To understand the fundamental difference between EM and programming, it is helpful to compare it to the difference between radical design and routine design in engineering [Vin93]. Michael Jackson (of Jackson Software Development methods fame) argues that programming is often treated as routine or normal design [Jac06] in which "the engineer knows at the outset how the device in question works, what are its customary features, and that, if properly designed along such lines, it has a good likelihood of accomplishing the desired task" [Vin93]. In radical design, by contrast, "how the device should be arranged or even how it works is largely unknown. The designer has never seen such a device before and has no presumption of success. The problem is to design something that will function well enough to warrant further development" [Vin93]. Jackson views conventional software development methods as reflecting routine design because they treat the specification as a solid interpretation of the world, and therefore the design process is concerned with a reduced problem of how to turn the specification into a program [Jac06]. The right-hand side of Figure 3.1 illustrates routine design with respect to programming. Jackson believes that radical design cannot be fully be addressed through the concepts of routine design [Jac06]. EM can be considered more like radical design because it is concerned with interaction that negotiates meaning between a situation in the world and a construal, and with interaction that is prior to ritualisation as described in Chapter 2 and illustrated in Figure 2.1 on page 32. EM is closer to 'immature' design where the situation is not well understood, where there is no presumption of success and where the emphasis is on creating something that might stimulate further exploration. The left-hand side of Figure 3.1 illustrates radical design from an EM perspective. Figure 3.1 depicts the fluid interpretation between the world and the construal that is characteristic of early interactions prior to ritualisation. Figure 3.1 also reflects the possibility of many different outcomes from creating a construal. For example, there have been many resulting models from the 3D room model described in Chapter 4: an instrument for a teacher to give presentations relating to 3D graphics; a walkthrough of interactions on a 3D room for a student to

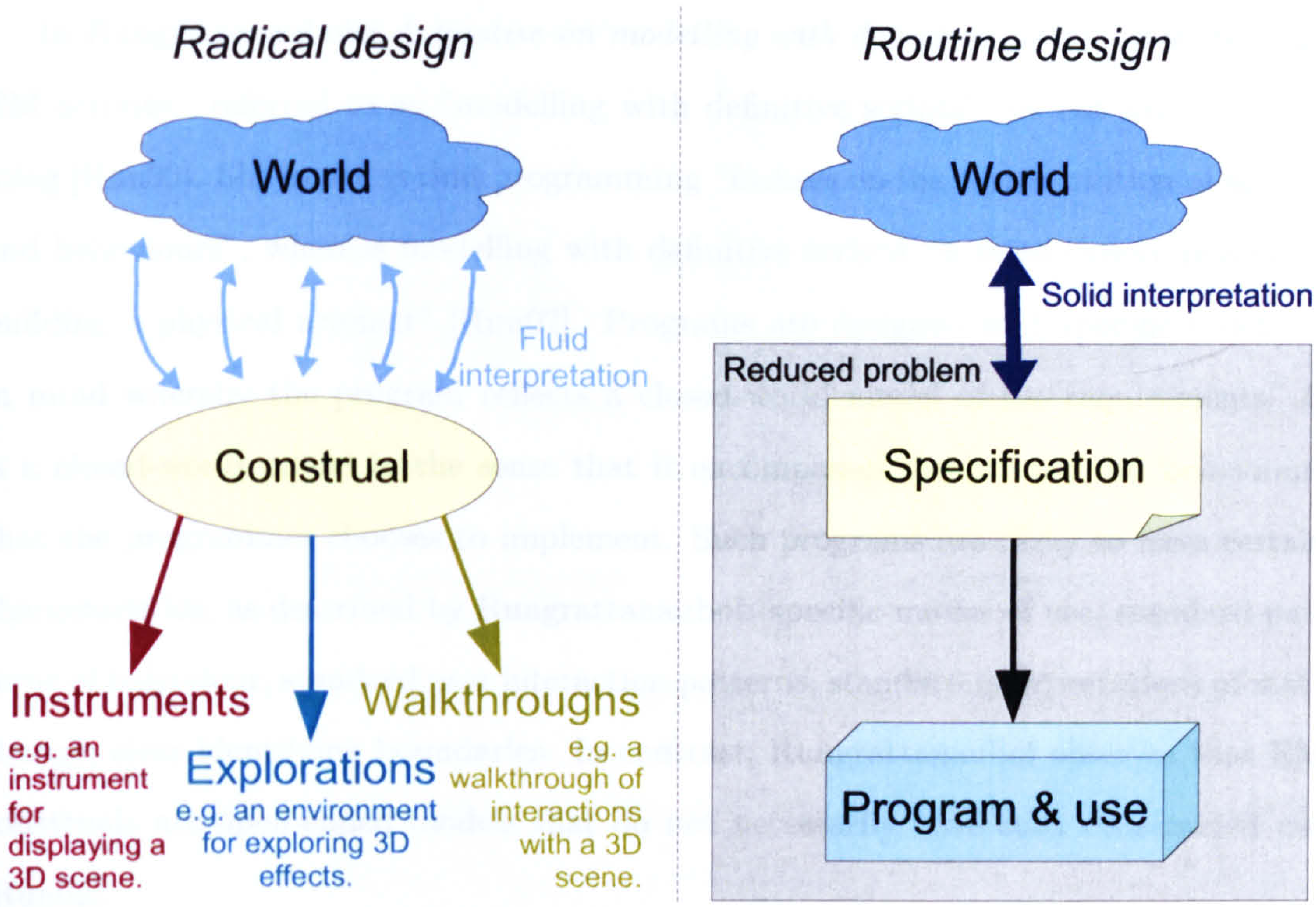


Figure 3.1: Contrasting EM and programming from a radical and routine design perspective.

follow; and, an exploration environment for students learn about 3D issues in computer graphics.

The contrast between radical and routine design is relevant to the following five sections which elaborate on the five distinctions outlined above between EM and programming.

3.1.1 Programs are more constrained than EM construals

The discussion in Chapter 2 associates EM activity with the construction of construals that reflect everyday concurrency in a situation or referent. The goal of such activity is to 'make sense of', realise, and understand the nature of some aspect of the referent or the referent itself. The activity is concerned with experience that comes before a formalised understanding or that is *prior to ritualisation*—see Figure 2.1 on page 32. A model-builder interacting with an EM construal is exposed to experiences that can be unpredictable in a fresh or raw sense, and a model-builder typically does not know if such interactions can be repeated or ritualised. Programming is more concerned with creating artefacts with predictable behaviours: a program is a ritualisation of a specific function that the user can repeat over and over in a predictable manner.

In Rungrattanaubol's *A treatise on modelling with definitive scripts*, it states that EM activity—referred to as "modelling with definitive scripts"—is not like programming [Run02]. She explains that programming "focuses on the representation of actions and behaviours", whereas modelling with definitive scripts "is more closely related to building a physical artefact" [Run02]. Programs are designed with specific functions in mind whereby the program reflects a closed-world model of the requirements. It is a closed-world model in the sense that it encompasses the actions and behaviours that the programmer chooses to implement. Such programs are likely to have certain characteristics, as described by Rungrattanaubol: specific modes of use, standard patterns of behaviour, standard user interaction patterns, standard interpretations of state change, clear identifying boundaries. In contrast, Rungrattanaubol observes that EM construals are open-ended models that do not necessarily have such constrained use [Run02].

Programs are closed-world models of specific ritualised behaviours due to the solid interpretation of the relationship between the program and the situation in the world as shown in Figure 3.1. The interpretation is solid in that the program has a single clear correspondence to the world defined by its specification. EM, on the other hand, is about developing a correspondence between the construal and the world—a correspondence which might change and grow stronger as the construal is constructed. EM construals are open-ended models that have a very fluid interpretation of the world. It is therefore important to distinguish the constrained nature of programs from the open-ended nature of construals.

3.1.2 Programming entails many discrete phases

EM is essentially an activity that involves interaction with a state to develop a correspondence between a construal and a situation in the world. The interaction through model-building involves a blended mix of creation, experimentation, observation, adaptation, exploration and manipulation—all of which are state-changes. Programming activity, in contrast, involves many discrete phases of different character as shown on the right-hand side of Figure 3.2. The traditional view of software development is captured in the systems life cycle as explained in Pressman's book on *Software Engineering* [Pre05]. This is a structured sequential process as shown on the left-hand

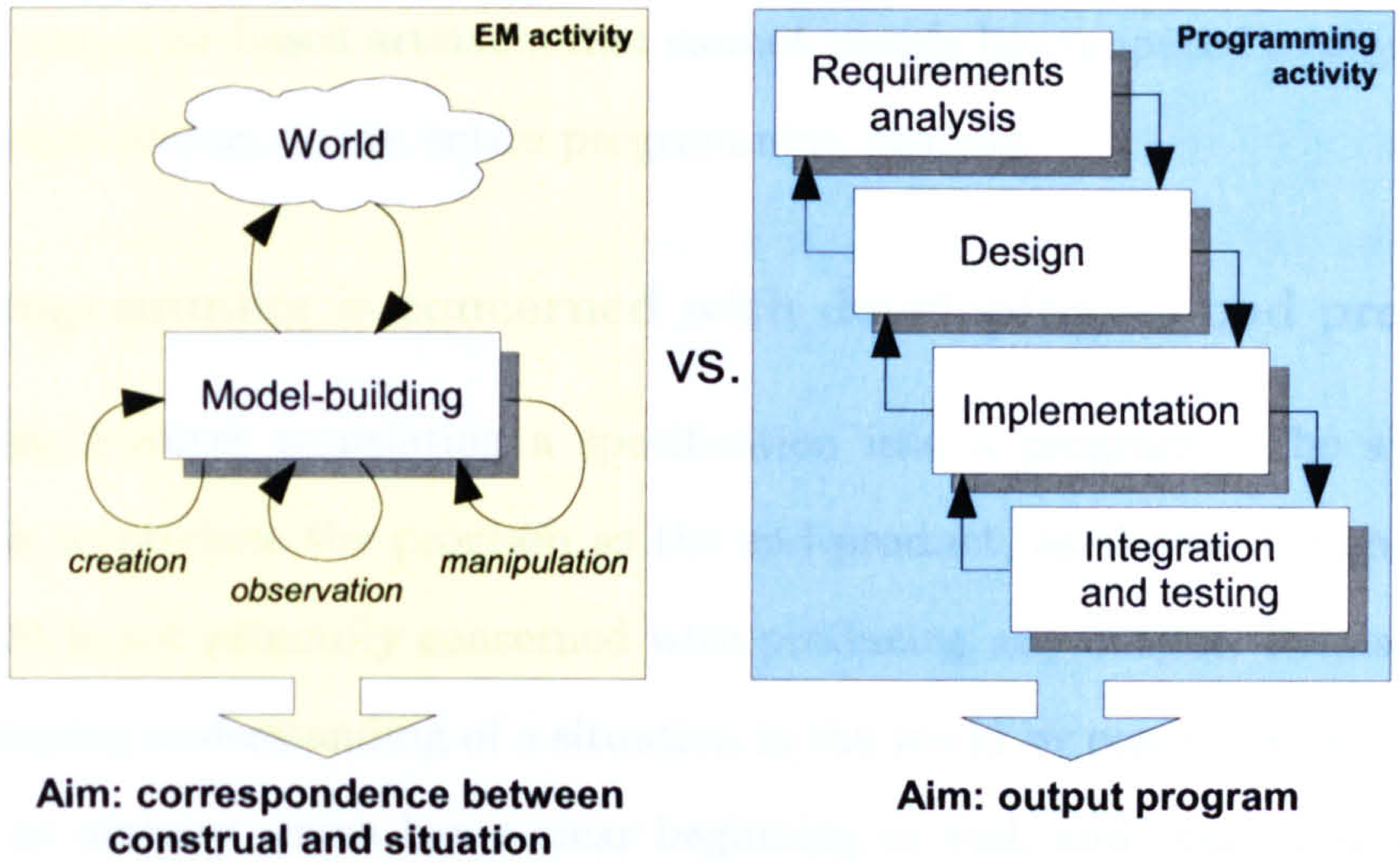


Figure 3.2: The discrete phases of programming contrasted with EM.

side of Figure 3.2, starting with requirements, moving through specification, design, implementation, and ending with testing and integration. Although the systems life cycle model has been highly criticised, its successors retain similar discrete phases for requirements, specification, design and implementation. Newer, more popular, methods generally involve the iteration or repetition of similar discrete phases (e.g. rapid prototyping involves iteratively specifying and implementing, and eXtreme Programming repeats phases such as coding, testing, listening and designing [Bec00]). As yet, there is little evidence of any fundamentally new methods that could support the fluidity of radical design.

Recent work by Beynon, Boyatt & Russ shows that EM necessitates a radical rethink of programming whereby there is no distinction between the stages of specification, design and implementation [BBR06]. Such ideas are reflected in programming practice as discussed by Weinberg, who states that "[in most cases] we do not know what we want to do until we have taken a flying leap at programming it." [Wei88:p12]. Weinberg goes further to undermine the discrete stages of software development: "Specifications evolve together with programs and programmers. Writing a program is a process of learning—both for the programmer and the person who commissions the program." [Wei88]. A full discussion of EM as an approach to software development is beyond the scope of this thesis (and has already been comprehensively covered by Rungrattanaubol [Run02]). The important point is that the continuous blended activities in EM are quite distinct from the discrete phases involved in programming. EM is a *holistic* approach to building computer-based artefacts that cannot simply be compared to any particular phase of programming, or the entire programming process.

3.1.3 Programming is concerned with developing an end product

Programming involves translating a specification into a program. The aim of programming is to produce the program as the end-product, as shown in Figure 3.2. In contrast, EM is not primarily concerned with producing any output. Primarily EM is about developing understanding of a situation in the world by constructing a construal. EM is not an activity which has a clear beginning or end, and construals are always subject to revision and extension. EM does not necessarily lead to a 'final' model as an end-product (see the discussion of EM characteristics in §2.2.4 & §2.2.5). In a similar manner to radical design, EM activity may result in many outcomes as illustrated in Figure 3.1, and the outcomes are not predictable like they should be in successful routine design.

Because the aim of programming is to produce a single end-product, programming is not well matched to the aims of everyday learning which typically might involve many unpredictable outcomes. From this perspective, EM has much more in common with learning than programming because it is an activity that is well aligned to sense-making and developing understanding, and the outcomes of EM activity are not constrained. The activity of sense-making through the construction of a construal is really concerned with learning about a situation, not producing a model.

3.1.4 Programming is concerned with the correctness of a program

Given that a programmer approaches the programming task with the specification already defined, it is important that the program produced accurately represents the specification. This leads to an area of computer science known as program validation and verification. Many computer scientists are concerned with what Jackson calls 'reduced problems' [Jac05]. A reduced problem is simply turning a 'functional specification' into a program without the concerns of the situation in the world that the program relates to, as illustrated in Figure 3.1. Dijkstra, famous for his many contributions to computer science, particularly formal verification, believed that the activity of coming to a functional specification was for others to address, and that computer scientists should solely concentrate on applying formal techniques to develop programs that 'correctly' adhere to the functional specification [Dij89]. Jackson is one of a few people (supporters of eXtreme Programming included [Bec00]) who advocate programming methods that are concerned with the relationship between the world and the program/specification.

Where there is a single unambiguous interpretation of the world in the specification, and the world is therefore placed in the background, it is possible to reason about the correctness of the program in relation to the specification. However, EM *is* fundamentally concerned with negotiating the interpretation of a situation in the world with respect to the artefact. Therefore it is much harder (if possible at all) to evaluate the 'correctness' of the artefact. The fluid interpretations that are possible with a construal are open-ended, changing, and not subject to correctness concerns. If EM is like radical design, as depicted in Figure 3.1, then the concerns are simply whether a construal might stimulate further exploration and better understanding. EM is much more concerned with the usefulness, rather than correctness, of an artefact.

3.1.5 Programming makes a distinction between development and use

At the end of a programming activity, when the program is complete, it is delivered to the user where the program submits itself wholly to software use, as depicted on the right-hand side of Figure 3.3. The activity of programming implies a distinction between development and use. The programmer develops the program, and the user uses it. The user does not develop the program, and rarely does the programmer use the program. The software engineering process often involves a third party—the client—as shown in Figure 3.3. The client is described as the person who specifies the requirements for a new piece of software [Pre05].

In EM, development and use are combined, and can be treated as one and the same activity. The distinction between construal creation and program development/use is depicted in Figure 3.3. When the aim of the construal creation activity is developing a correspondence between the construal and the situation modelled, it is clear why constructing a model and using it are inseparable. Rungrattanaubol shows that because EM "respects the continuity of the modeller's perception, design and use can be inter-

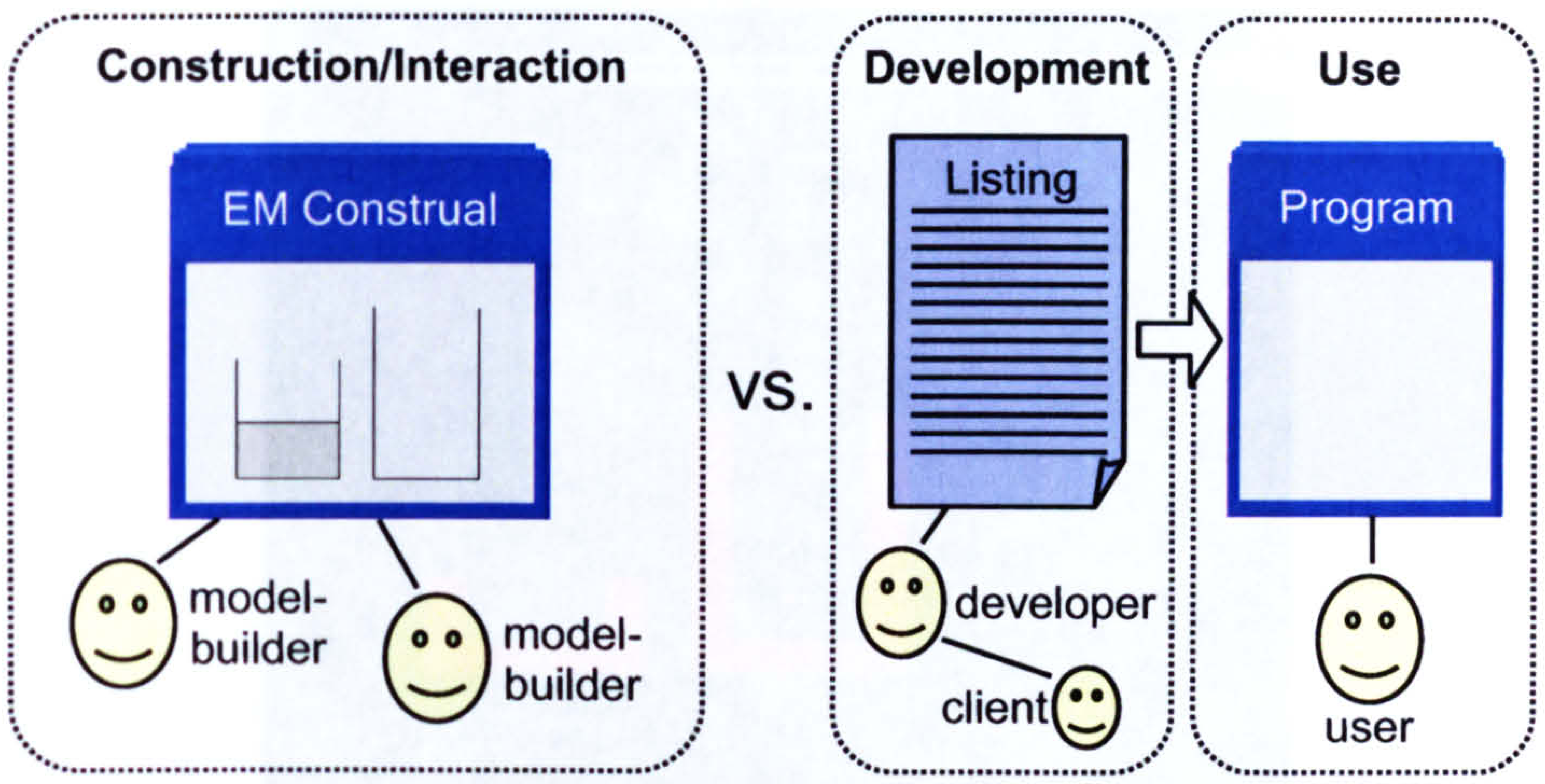


Figure 3.3: EM contrasted with conventional software development and use.

leaved without interference, and the role of the modeller is more aptly characterised as 'designer-user' rather than 'design-exclusively-or-user"' [Run02]. The consequences for drawing no distinction between development and use are highlighted later in the chapter when considering educational technology.

3.2 Illustrating the distinction

An example of three varieties of an educational artefact are used in this section to elaborate on the differences between software development and EM. JUGS is a simple educational program that was initially developed by Ruth Townsend for the BBC Microcomputer in 1982 (see Figure 3.4 for a screenshot of JUGS restored with the BeebEm BBC Emulator). The underlying educational objective of the JUGS program is to familiarise school children with elementary concepts of number theory. The idea is to create an environment for exploration in which students can come to appreciate that the highest common factor of two positive integers determines what numbers can be derived by repeatedly applying addition and subtraction operations. The program presents the student with two jugs of non-identical integer capacities and options to fill or empty each jug and to pour from one jug to the other. The student is expected to apply these operations to reach a target quantity. The mathematics embodied in the use of JUGS is, given two integers `m` and `n`, the set `AS(m,n)` of numbers that can be generated from `m` and `n` by additions and subtractions alone is the set of multiples of `hcf(m,n)`. In applying fill, empty & pour operations to two jugs that have integer capacities, it follows that every operation generates a quantity of liquid that is in `AS(m,n)`

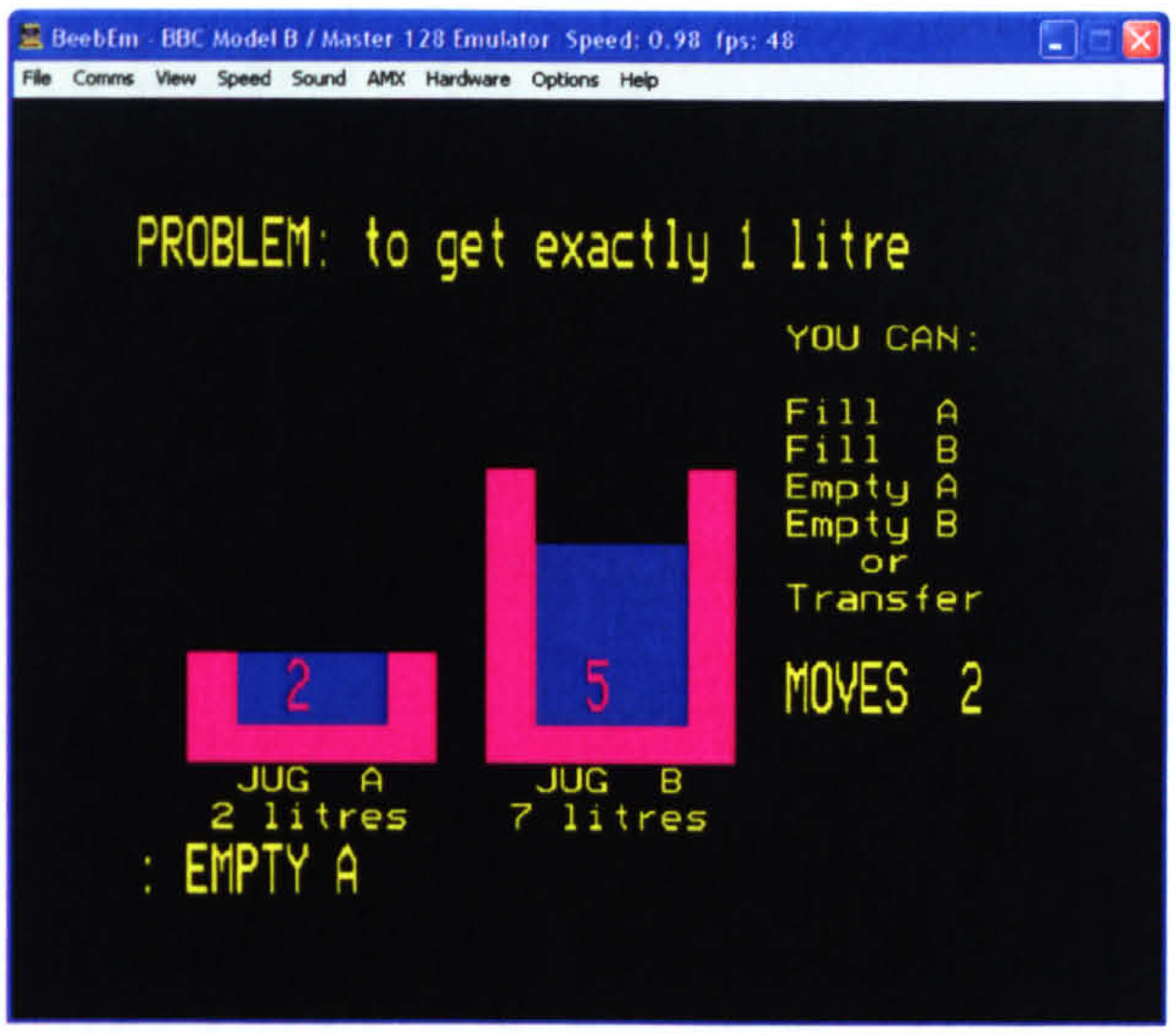


Figure 3.4: The JUGS program (running on the BeebEm emulator).

and that this quantity is restricted to be positive and cannot exceed `max(m,n)`. It is also true that all quantities satisfying these constraints can be derived in this way.

3.2.1 Developing a JUGS program

Some insight into the development of the JUGS program can be gained from analysing the source code. The program is 450 lines of BBC BASIC code containing 36 procedures. There are global variables for the capacity (`capA` and `capB`) and the level of the liquid (`levelA` and `levelB`) of each jug and for the target quantity of liquid (`target`). The code that assigns and uses these 5 variables is a tiny proportion of the entire program (only 44 lines)—less than 10%. There is a procedure for demonstrating how the fill, empty & pour operations can be used to reach a specific target which uses an algorithm to obtain the highest common factor. The code that implements the relevant mathematics for this is only 11 lines long. The majority (at least 51% or 229 lines) of the code is concerned with drawing or printing to the screen. Another proportion (19% or 86 lines) is concerned with the input and decoding of commands. Figure 3.5 shows the source code together with the control flow of the program, with the 5 variables relevant to the task in bold. Many of the connecting lines have a relationship to dependency maintenance in that the procedures are updating part of the state, particularly the screen state. For example, whenever the level of either jug changes, the procedure `PROCdisplayupdate` is called to update the jugs on the screen in order that changes to the `levelA` and `levelB` variables can be observed.

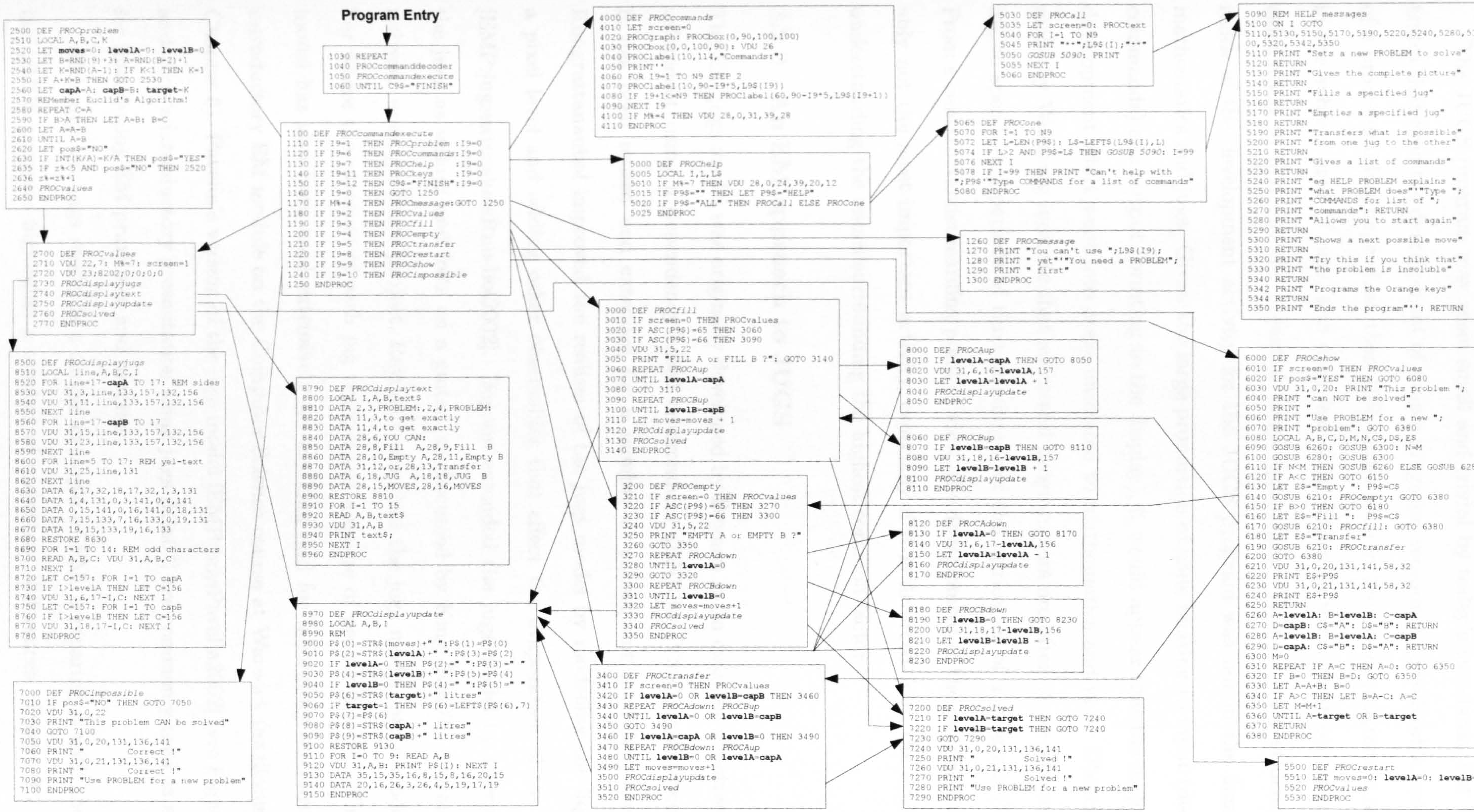


Figure 3.5: The listing and flow of the JUGS program.

The JUGS program may seem small and trivial by today's standards, but it illustrates that even the development of simple programs requires a considerable knowledge of a procedural language. Less than 10% of the code in the JUGS program is actually in the subject area of jugs, and even less of it relates to the highest common factor calculation to find out the potential target quantities. This provides evidence that much of the development activity for the JUGS program was not in the domain of mathematics and jugs. Given the large proportion of code relating to input (decoding commands) and output (printing to the display), it would appear that the majority of development effort went into areas unrelated to mathematics and jugs. Observations made by Weinberg indicate that successful programmers are often those who are willing to spend long periods of time puzzling over the intricacies of a program [Wei88]. From a constructionist learning point of view, the development of the interface is probably not the most important activity, especially if it is divorced from the activity of understanding the issues surrounding the highest common factor.

3.2.2 An EM approach to JUGS

The EM jugs model was originally developed by Beynon as an example of software development using a dependency-based environment [BNR89] [EMP:jugsBeynon1988]. It has been reused and extended for a number of subsequent modelling exercises. Rungrattanaubol improved the realism of the jugs model by modelling the liquid at a pixel level and adding other observables that effect the evaporation of the liquid [EMP:jugsextensionsRun-bol2002]. Beynon extended the jugs model by interpreting the jugs as strings and frets on a guitar, as discussed by King [Kin07]. During a final year undergraduate project, Reynolds morphed the jugs model into a model of a bar where the contents of each jug represented a queue of people [Rey04]. The jugs model has also been used extensively for teaching and for laboratory sessions in an introductory EM module on the Computer Science course at Warwick (as discussed in Chapter 6). Pavelin's version of the jugs model [EMP:jugsPavelin2002] (see Figure 3.6) and King's introductory presentation using jugs [Kin07] are commonly used as models for introducing EM principles and practice.

Although the jugs model is a simple example of EM, comparing it with the original JUGS program highlights a number of important differences. Figure 3.7 shows

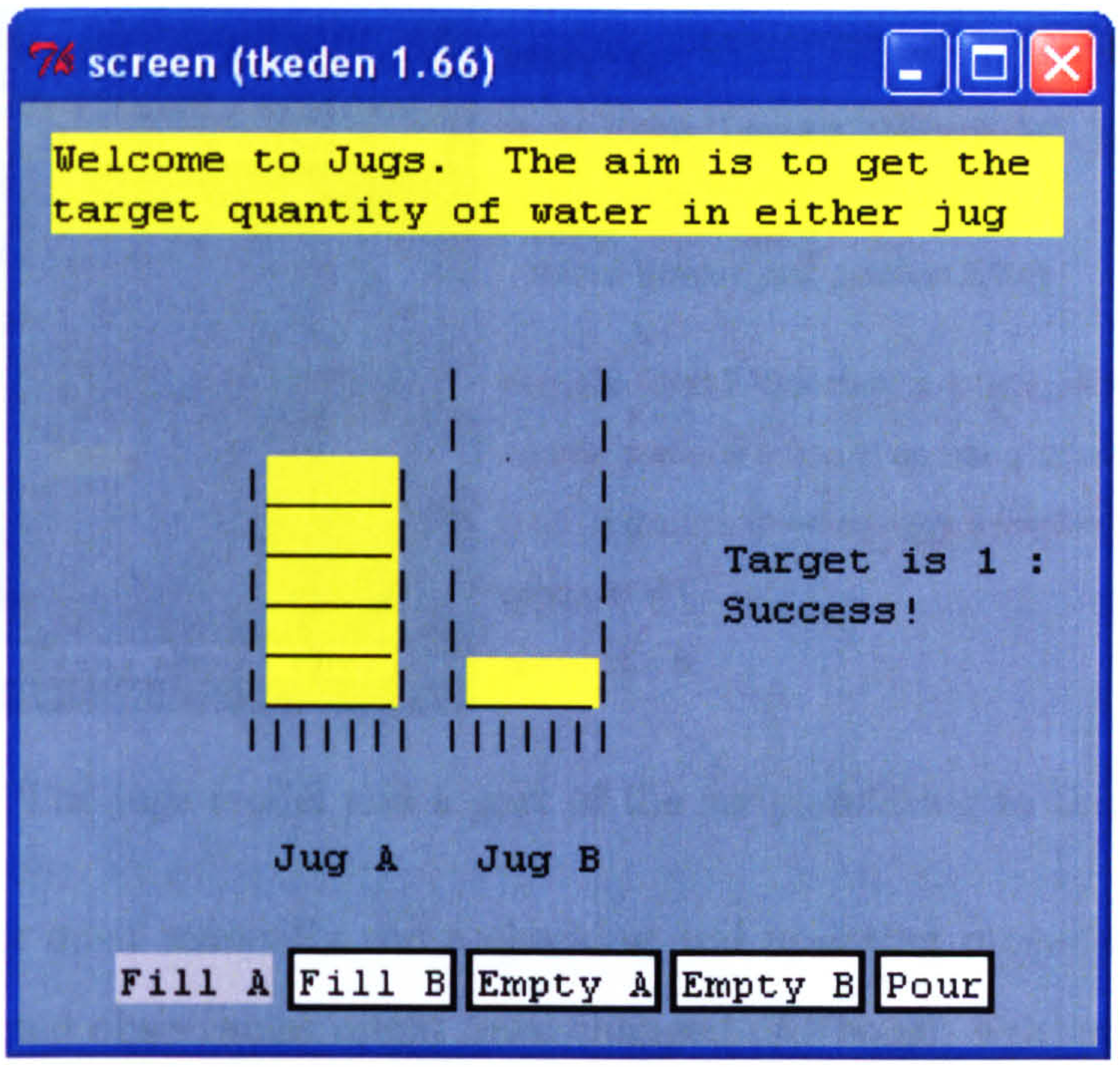


Figure 3.6: Pavelin's variant of the jugs model [EMP:jugsPavelin2002].

the jugs model and a definitive script that embodies the part of the jugs model relating to the status bar. During interaction with the jugs, the contents of the status bar corresponds to an aspect of the current state. For example, if either of the jugs is currently filling, emptying or pouring, then the status bar will show "updating". Both the screen state and the underlying state are maintained by dependency. The current state in Figure 3.7 is that the target of 1 unit of liquid has been met in Jug B. In other words, `contentA` is equal to `target`. Through dependency, the observable `finish` is true, as long as `contentB==target` remains true and assuming `active` is false. Once again through dependency the observables `update_status`, `status` and finally `status_box` are maintained in response to changes to `finish` and other observables. The `status_box` definition is defined in the SCOUT notation for screen layout, and embodies the status bar artefact displayed on the jugs screen as highlighted by the red outline. The two red shaded areas represent two parts of the `status_box` string, the first dependent on the `target_text` and the second dependent on the `status`. The benefit of using dependency is evident when comparing the script in Figure 3.7 with the part of the program listings in Figure 3.5 involving `PROCsolved` (second from the right at the bottom). The EM jugs model shows "Success!" in the status bar by dependency whenever the selected observables are in the described state. With the JUGS program,

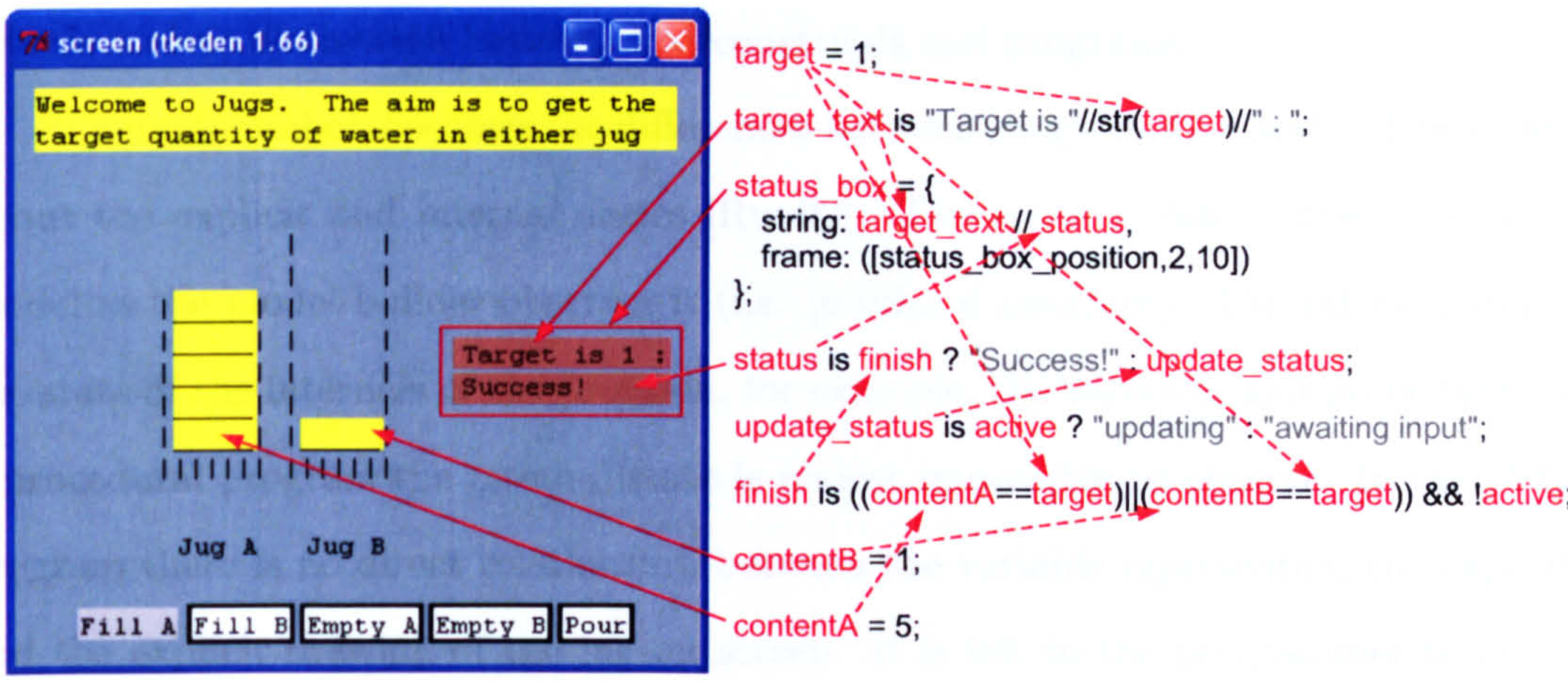


Figure 3.7: The jugs model and a part of the script relating to the status bar.

the programmer must manually run a checking and updating procedure whenever he deems the selected observables might have changed. Although writing a procedure to update the status bar may be fairly trivial (such as in `PROCsolved`), reasoning about all the places where the procedure should be called from is more difficult. Furthermore, if a program changes over time, by adding another jug for instance, it becomes even harder to ascertain whether the "Success!" in the status bar will always correspond to realising the target. An issue was discovered in the original JUGS program relating to this update problem: when the target is reached "Success!" is displayed as expected, but if you then make an error, or request help, then "Success!" disappears. Such problems are rarely a concern for a model-builder using dependency.

If the model is interacted with in a restricted way (i.e. using the interface buttons alone) then, as Rungrattanaubol points out, it serves the same function as the JUGS program [Run02]. However, the jugs model can be interacted with in an open-ended manner that has no counterpart in the JUGS program. When the JUGS program is executed, it is a closed-system and there is no potential for observing the internal state of the program or alter the state at any time during the execution (except in ways allowed by the specification). In the EM model of jugs, the underlying state is exposed and open to change at any time during construction and use (between which no distinction is made as described in §3.1.5). The number of models that have built upon jugs demonstrates that models can be extended rapidly for a variety of purposes. Furthermore, extensions can be performed in the stream of modelling because EM construals are visible and alive, constantly running even during modification. This reveals a clear distinction between EM construals and programs.

Rungrattanaubol discusses the differences between programming and EM by talking about the explicit and internal states [Run02]. The *explicit state* is the state of the model as the model-builder observes it (i.e. graphical interface). The *internal state* is the state of the internals of the program, for example, the variables and procedures. In a procedural program the internal state is almost impossible to observe. In the JUGS program there is no direct relationship between the variable representing the capacity and the explicit drawing of the jug on screen. It is left to the programmer to decide under what circumstances the screen should be redrawn, given that the screen need not be updated in every situation. In EM, with the use of dependency and notations such as DoNaLD and SCOUT [War04], the model-builder can construct an explicit state that is linked to the internal state, and vice versa. The internal and explicit states are kept consistent at all times; a change in the internal state will be immediately reflected in the explicit state as viewed by the model-builder. In the jugs model, for example, it becomes difficult to differentiate between internal state and explicit state: is `status` explicit or internal? Internally it is a string, explicitly it is a part of the status bar that can be observed. In EM, all the internal states have the potential to be explicitly observed, hence it is not necessary to differentiate as is common in programming.

3.2.3 JUGS by object-orientation

It could be argued that our comparison with a program written in BBC Basic is unfair, because nowadays object-oriented (OO) languages are more commonplace and this makes programming easier or removes some of the problems of procedural programming. From a novice perspective there are difficulties associated with learning OO languages when compared to high-level procedural languages such as BBC Basic. During the time of the BBC Micro, teachers wrote their own small, highly specialised programs for their students. How often does a teacher write a Java program for their students? Educational technology is usually built by professional programmers and it is often general purpose software designed for a wide range of teaching tasks (e.g. Toontalk [Kah96], Imagine Logo [KB00]). Programming in Java in many ways is more difficult than the days of the BBC Micro. One only has to compare the number of lines required to write a "hello world" program in Java to a similar program in Basic to see that OO

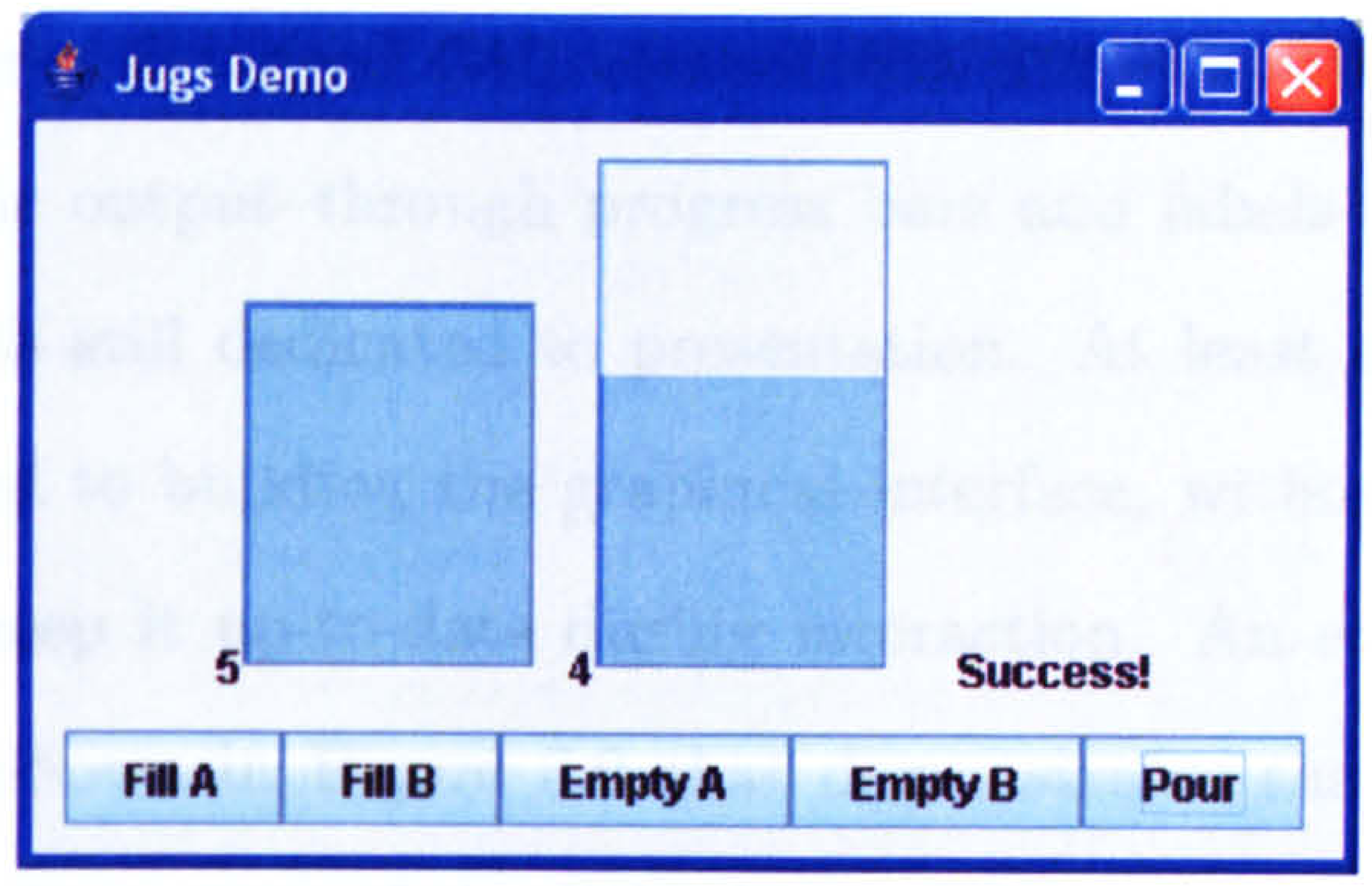


Figure 3.8: The Java JUGS program.

programming does not always make programming easier for novices. Sheetz et al [She97] explore in detail the difficulties of OO design and programming. They found that learning the basic concept of objects was generally difficult, even before moving on to more complicated issues such as class reuse. Wiedenbeck et al [WRSC99] discovered in experiments that procedural programming students' were superior to OO programming students' in the comprehension of large programs in every category analysed. The findings suggest that although OO programming is a successful paradigm for software development, it is not necessarily an easy approach to programming for novices.

A Java version of the JUGS program was implemented by the author to demonstrate the differences when compared with the Basic JUGS program and the jugs model. The development of Java JUGS was undertaken in one morning, and it contains only very basic functionality compared to the Basic JUGS program. As shown in Figure 3.8, there are two jugs displayed and 5 buttons for filling, emptying and pouring.

Java JUGS is only 222 lines long split across two classes. The use of classes can be seen as an advantage over the original JUGS program because Jug is a class and so therefore creating two or more jugs is a simple matter of declaring and instantiating each Jug object. I was also able to make use of the standard GUI (Swing) components and layouts to arrange the buttons and jugs (as opposed to the primitive lines and text in BBC Basic). The jugs shown in Figure 3.8 were in fact standard JProgressBar objects (oriented vertically) that had a specific size (the height of the jug) and their progress value represented the level of the liquid in the jug. The use of pre-built components can speed up development time and reduce the amount of code required to perform input and output. It also ensures that Java JUGS fits in with the operating system (in this case Windows) and looks and feels like a standard application.

Although the use of standard GUI components simplifies the input–through pre-built buttons–and the output–through progress bars and labels–a large proportion of code in Java JUGS is still dedicated to presentation. At least one third of the code (80 lines) is dedicated to building the graphical interface, without including the code that is required to keep it up-to-date during interaction. An extract from the main Java JUGS class is shown in Figure 3.9 that demonstrates the nature of this code. In comparing this code to that of the BBC Basic JUGS, the latter is concerned with constructing the interface by lines and strings, whereas the former is constructing the interface by buttons, labels and progress bars. The objects may be different, but the task is the same, and the difficulty of updating the interface and maintaining consistency in Basic JUGS is inherited in Java JUGS. From an EM standpoint, these difficulties come from the apparent division of the interface from the underlying core of the program. In Java JUGS, the interface is quite independent of the Jug class, encouraging the development of an underlying abstract model that is separate from the interface through which interactions with the jug are performed. In the EM model, dependency ensures that the interface is the model. The separation between interface (the explicit state) and underlying operation (the internal state) is blurred, usually to the extent that model-builders do not talk about 'the interface'. Whereas a program can consist of an interface and an underlying abstract or mathematical model, an EM artefact creates no separation and therefore in referring to a model it implies the whole artefact (not some abstract/mathematical model underneath the program).

Even if we agree with Meyer's contentious claim that "object-oriented designers usually do not have to spend their time in academic discussions of methods to find the objects: in the physical or abstract reality being modelled, the objects are just there for the picking!" [Jac05:p14], object-orientation can lead to other problems. Information hiding as a feature of object-orientation has its advantages, but deciding *what to hide* in classes can be troublesome. There are some objects or methods that might clearly belong to the Jug class, but there are others where classification is not so trivial. In experiments with microworlds, Goldstein et al. [GKN01] observed a number children grappling with similar concerns as to whether to put the X in the Y or the Z. As a personal example, during development I was unsure whether to put the 'fill' and 'empty' methods inside the Jug class. The reason I did not was because these two operations

```java
private void buildGUI() {
    JFrame frame = new Jframe("Java JUGS Demo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JPanel panel = new JPanel(new BorderLayout());
    frame.getContentPane().add(panel);

    panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

    //Add labels for jug content/level.
    levelAlbl = new JLabel("0");
    levelBlbl = new JLabel("0");

    //Add progress bars to represent jugs
    levelApb = new JProgressBar(SwingConstants.VERTICAL,0,jugA.getCap());
    levelBpb = new JProgressBar(SwingConstants.VERTICAL,0,jugB.getCap());
    levelApb.setMaximumSize(new Dimension(80,20*jugA.getCap()));
    levelApb.setMinimumSize(new Dimension(80,20*jugA.getCap()));
    levelBpb.setMaximumSize(new Dimension(80,20*jugB.getCap()));
    levelBpb.setMinimumSize(new Dimension(80,20*jugB.getCap()));
    levelApb.setAlignmentY(Component.BOTTOM_ALIGNMENT);
    levelBpb.setAlignmentY(Component.BOTTOM_ALIGNMENT);

    //Add panel for jugs
    JPanel jugpanel = new JPanel();
    jugpanel.setLayout(new BoxLayout(jugpanel, BoxLayout.X_AXIS));
    jugpanel.add(levelAlbl);
    jugpanel.add(levelApb);
    jugpanel.add(Box.createRigidArea(new Dimension(10,0)));
    jugpanel.add(levelBlbl);
    jugpanel.add(levelBpb);
    jugpanel.add(Box.createRigidArea(new Dimension(10,0)));
    panel.add(Box.createRigidArea(new Dimension(0,10)));
    panel.add(jugpanel);
```

Figure 3.9: An extract from Demo.java that constructs the GUI for the Java JUGS program.

are similar to the 'pour' operation which has to work from one jug to another, and so I placed them all together with the button functionality. The emphasis on objects in Java JUGS leaves question marks around where to put the agency: is it in the object's methods, or is it in the button's action? Another difficulty I came across when deciding what to put in the class was with the interface components. My Jug class described the abstract properties of the jug, i.e. its capacity and its current level of liquid. When I constructed the user interface I created two progress bar objects for the graphical display of their capacity and content. After this I struggled with my reasoning as to the 'correct' place for these components. Should the progress bar be placed inside the Jug class or should it be placed with the other GUI components? Initially I decided the jug interface components should be put inside the jug class, but then after further deliberation I realised that this might severely limit further use of the jugs if it were to be morphed into another application such as King's guitar strings [Kin07]. A more experienced object-oriented programmer would no doubt spend less time picking the relevant bits for a class, but still there is the issue, as there is with traditional programming, that making decisions about the structure of a program may restrict future developments—which is not desirable for learning in a constructionist sense.

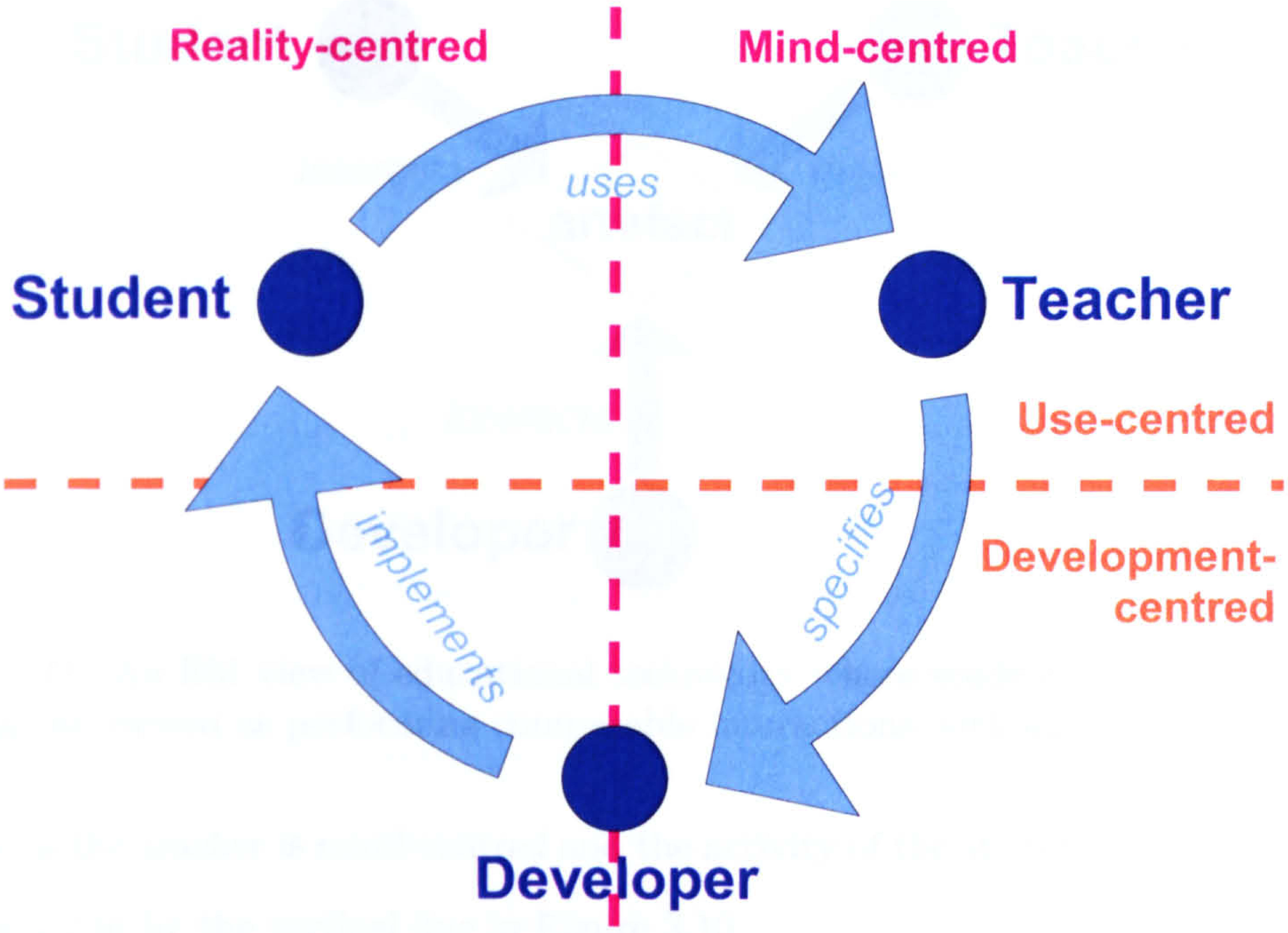


Figure 3.10: A typical software development view of educational technology displaying the perspectives of student, teacher and developer.

3.3 Implications for educational technology

Educational technology generally follows a pattern of development and use that mimics traditional software where the *student* can be thought of as the user, the *teacher* as the client, and the *developer* remains the same. The activity depicted in Figure 3.10 is a typical software development view of educational technology. The cycle can be seen as starting with the teacher who *specifies* requirements for a computer-based tool for supporting student learning. The developer takes the specification and *implements* a solution (e.g. a program or group of programs). A number of students can then *use* the tool to support or supplement their learning activity.

Each of the three roles are associated with distinct activities, typically: specification, implementation and use. Furthermore, the concerns in these activities create two important tensions. The first tension is that the developer is focussed on development, whereas the student and teacher are focussed on use. The horizontal line in Figure 3.10 signifies the division of software development versus software use. The second tension is that the teacher is concerned with teaching particular concepts (public knowledge in the lower realm of the EFL), but the student is relating to the technology and their experience of it (practical knowledge in the upper realm of the EFL). Therefore the

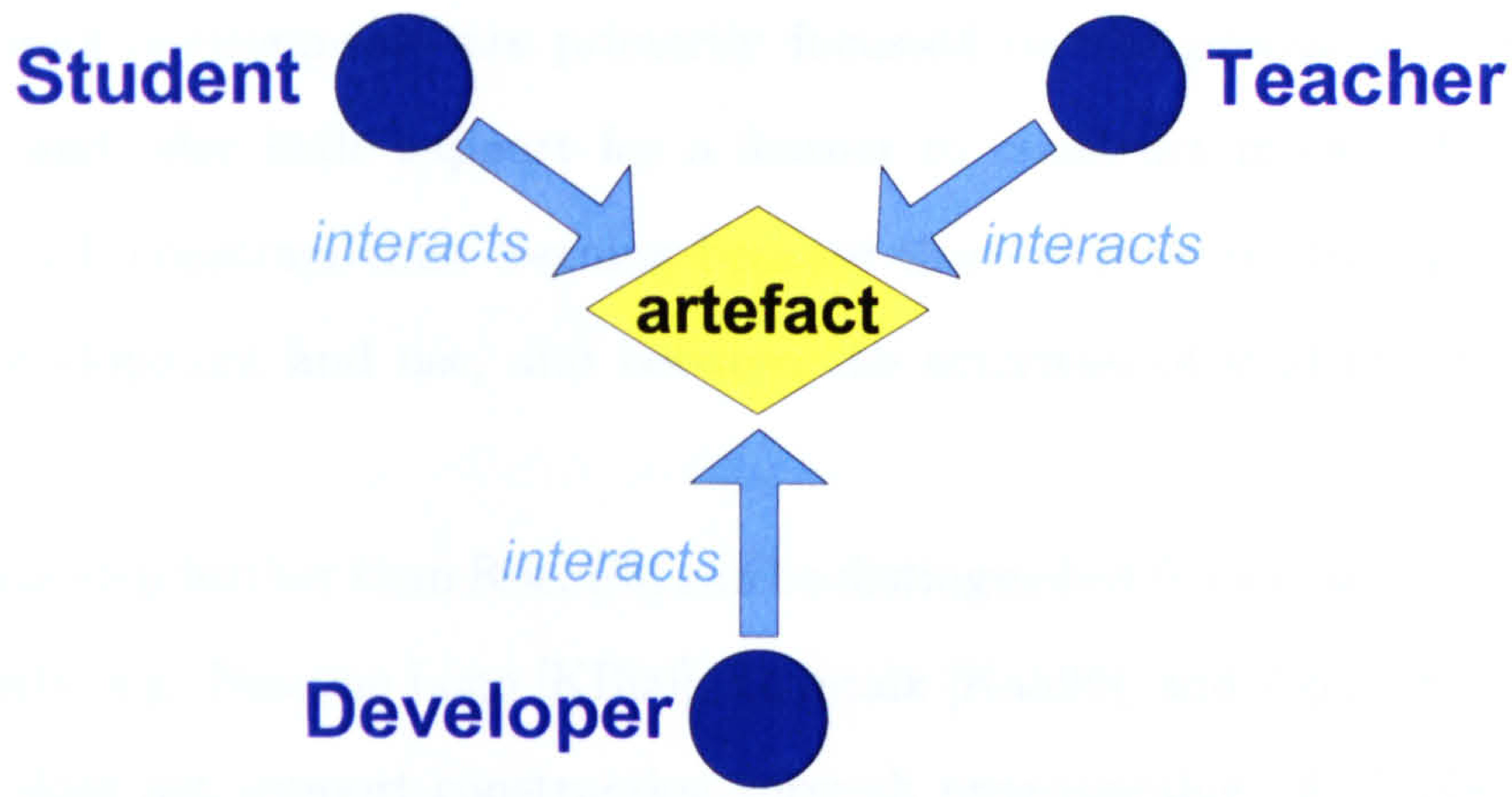


Figure 3.11: An EM view of educational technology where student, teacher and developer can be viewed as performing comparable interactions with an artefact.

activity of the teacher is mind-centred and the activity of the student is reality-centred, as depicted in by the vertical line in Figure 3.10.

EM is primitively concerned with modelling state-as-experienced, and all interactions with an EM artefact are state-changes. The activities of the student, teacher and developer are comparable because state-change occurs only through re-definitions. A student exercising the artefact is changing the state, a teacher preparing the artefact for a particular exercise is changing the state, and a developer constructing the artefact is changing the state. In other words, at a primitive level all possible interactions with the artefact are similar activities as in Figure 3.11. This is not to say that all interactions have to be at the level of making a single redefinition to an observable through the input window, an interaction such as a button press can be linked—through dependency or agent actions—to a series of redefinitions or state-changes in the artefact.

Following the EM approach, the implications for learning are that there is more potential for learners to interact with an artefact in a wide range of activities that would normally only be associated with either students, teachers or developers. A student playing with the jugs model can be a teacher by changing the initial state to make the target quantity impossible to reach, or a developer by adding a third jug for example. Furthermore, the blurring of the roles of student, teacher and developer removes the tensions between development and use, and between reality-centred activity and mind-centred activity, as found in Figure 3.10. Roe, Pratt and Jones [RPJ05], in reflecting on their experiences of building microworlds for mathematics education, highlight the need to give learners control to express their own ideas in their models. They add that web-based learning environments are primarily focussed on the production and delivery of content and offer little support for a learner to construct models [RPJ05]. EM facilitates such constructionist learning because there is little to distinguish between software development and use, and between the activities of students, teachers and developers.

To go one step further than Roe, EM can be distinguished from many constructionist environments (e.g. Imagine Logo [KB00], Toontalk [Kah96], and Agentsheets[Rep93])† because it does not support construction through programming—EM offers an alternative paradigm for construction based on observables, dependencies and agency (as described in §2.2.2). Programming as a method for learning by constructing artefacts has had some success. For example, ToonTalk is a visual environment for doing computations where each operation (such as assignment, branching, and looping) is given a visual metaphor (i.e. a loop is represented by a robot performing an operation many times) [Kah96]. Although ToonTalk wraps up the programming constructs in visually appealing ways, the conceptual difficulties of programming and object-orientation (as highlighted in §3.1) are hidden rather than resolved. Metaphors may help children to grasp the nature of the procedural programming, but they do not take away the difficulties of programming. EM is a possible solution for what is described as the need to put 'the learning back into e-learning' [RPJ05] by taking away the constraints of programming.

3.4 Spreadsheets as an illustration of modelling with dependency

Spreadsheets offer an environment quite different from programming—most notably in the way that development and use are integrated. In some respects, EM's distinction from conventional software development can be appreciated by demonstrating the similarities between EM and spreadsheets. The aim of this section is to trace the history of an application which may not have initially been conceived as a modelling environment but which is now used for a wide variety of modelling activities. Spreadsheets, as well as being related to EM, are also important applications in education. Therefore, a

†Roe discusses Imagine Logo [Roe03:p3] [RPJ05], Toontalk [Roe03:p4], and Agentsheets [Roe03:p36] in relation to EM.

treatment of the topic of spreadsheets is highly relevant to strengthen the connections between EM and learning developed in this thesis.

3.4.1 Tracing the rise of the spreadsheet

Before the arrival of computers, a 'spread-sheet' was the name given to a large sheet of paper containing rows and columns that could be used for financial activities such as book-keeping. When computers became commercially available, some of the first batch programs were designed to deal with payrolls and other finance-related activities. Many of these programs produced printed output in the form of large lists or tables similar to the traditional spread-sheets. It was not until Dan Bricklin came up with the idea of an interactive visible calculator, and developed the VisiCalc application in 1979 [Pow04], that computer-based interactive spreadsheets started being used †. In 1983, two years after the IBM PC was launched, Lotus Development Corporation released the Lotus 1-2-3 spreadsheet application for the IBM PC, which offered many benefits over VisiCalc [Pow04]. This application turned out to be the "killer-app" for the IBM PC, and played a significant role in the huge success of the IBM PC [Pow04]. In the late 1980s as Microsoft Windows took over from MS-DOS, Microsoft was quicker to develop a Windows spreadsheet application, and hence Microsoft Excel took over from Lotus 1-2-3 as the leading spreadsheet application [Pow04].

Spreadsheets have been widely used for financial and accounting purposes, but there are many other applications where they have made a significant impression (e.g. mathematics modelling, cellular automata and neural networks). Spreadsheets are used by software developers for building applications. Since Visual Basic for Applications was introduced into Microsoft Excel in 1993, developers have been using the Excel platform for small, often client-specific applications. For example, the Warwick Spreadsheet System (WSS) developed by Beare makes use of Excel and provides an environment teaching mathematical modelling that can be used in science and mathematics education [Bea96]. The merits of applying spreadsheets principles to software development have been noted by Nardi [Nar93]. Spreadsheets have also played a roll as a general paradigm for computing practice. The first integrated office application that Lotus Development Corporation released, called Symphony, was based wholly on the spreadsheet

†The first known implementation of an interactive spreadsheet program was developed on an IBM mainframe at Imperial Chemical Industries in the UK and was used as early as 1974 [Pow04].

principle: "While different environments display information in slightly different ways, Symphony is always basically a spreadsheet. The Row/Column structure is there, to one degree or another, regardless of which environment you are using" [Bad85]. There is plenty of evidence to suggest that spreadsheets have much to offer for constructing and using computer-based artefacts.

The potential of spreadsheets is perhaps best summarised with a quote by Alan Kay: "A spreadsheet is a simulated pocket universe that continuously maintains its fabric; it is a kit for a surprising range of applications." [Kay84]

3.4.2 Spreadsheets for learning

The JUGS program described in §3.2 is typical of ET around the time when computers were first introduced into schools. Such simple programs were often developed by teachers themselves. As software became able to perform ever more complex tasks, programming became a more specialist task. Simple programs by teachers were replaced by fully-featured educational software by teams of programmers. It was not feasible for all teachers to learn to program and to write their own programs. There was still a need though for teachers to build their own computer-based artefacts for specific topics.

Baker & Sugden, in the first article of their electronic journal *Spreadsheets in Education*, discuss how spreadsheets have been used in education since Lotus 1-2-3 was first introduced [BS03]. At this time computers were seen as having potential in an educational setting, but in most cases teachers or students had to learn a programming language. The other option was to purchase a specific piece of software, but each subject required at least one piece of software, and every piece of software had its own interface and style of use. However, this changed when spreadsheets became available. They offered an environment where students did not have to learn complicated languages or interfaces, and were not confined to one particular subject or use [BS03].

The first applications of spreadsheets in education were for teaching mathematics, but now spreadsheets are used in all areas of education [BS03]. For example, the Warwick Spreadsheet System covers a wide range of topics in the sciences [Bea92]. Shinners-Kennedy [Shi86] discusses the use of spreadsheets in computer science education and promotes "using the spreadsheet as an operating system" with an aim "to 'build' robust, interactive machines". Assembly programming has been taught using spreadsheets with several advantages over conventional assembly tools because using a spreadsheet gives the potential to observe the internal state and intervene at any instant.

The wide usage of spreadsheets in education indicates that the characteristics of spreadsheets are well-suited to learning. Baker & Sugden [BS03] suggest that it is the middle way between programming (software development) and buying off-the-shelf products (software use)—spreadsheets offer a combined interface for constructing and using an artefact or a model. Steward, in a paper on spreadsheets in mathematics education [Ste94], adds that a spreadsheet exposes the underlying relationships of a model, whereas a program hides the procedures from the user. Beare states that "spreadsheets promote open-ended investigations, problem-oriented activities, and active learning by students" [Bea92]. It is also true that spreadsheets are popular in education because they are readily available, familiar to existing users, and easy-to-learn for new users [BS03]. Spreadsheets enable us to build on what we already know–exploit our knowledge–in a familiar environment.

This resonates with Rungrattanaubol [Run02] characterising the spreadsheet as an 'open-modelling' environment, as opposed to a closed-world application such as programming. The distinguishing characteristics of 'open modelling' are that it "represents situations, allows meanings to evolve, and offers 'what if' experiment". Simulations and microworlds, although they have been shown to have potential in education, often share the same problems as closed-world applications. As described by Jonassen, "In microworlds, the model is not explicitly demonstrated. Learners have no access to the model and they cannot change it, except to manipulate a set of preselected variables within the model." [Jon06:p14]. Furthermore, research has shown that interacting with microworlds and similar simulation environments does not lead to development and change of mental models [Jon06:p14]. Model construction and manipulation—like in a spreadsheet—is necessary for learners because they learn more than they do from "trying to induce the underlying model in a black-box simulation" [Jon06:p14].

3.4.3 Comparing EM and spreadsheets

The connection between spreadsheets and EM arises in only some aspects of spreadsheet use. Spreadsheets used for business purposes often involve data analysis and visualisation including charts and graphs. In this case the spreadsheet is a tool with a specific goal in mind (i.e. calculating incomings and outgoings for book-keeping). The other use for spreadsheets is more common in education where the spreadsheet can be used for modelling, as described in the previous section. This use of spreadsheets has some of the characteristics of the model-building activity in EM described in Chapter 2. The following example of building a model of the game Sudoku highlights some of the characteristics of model building with spreadsheets that are relevant to EM as an approach to learning.

3.4.4 Illustrating spreadsheets for learning with Sudoku

The game Sudoku originates from Japan, where Sudoku means "one number", referring to the fact that the puzzle grid must contain only one number (1-9) in every column, row and region. Although Sudoku is generally played with the numbers one to nine, the numerals are only symbols and could be replaced by letters, pictures, or any other symbols. Numerous variations of Sudoku exist, but the original game is based on a square grid with 9 columns, 9 rows, and 9 square sub-regions (of 3x3 cells). Sudoku is similar to Latin Squares with an added constraint that each region can only contain one of each number (or symbol). A puzzle starts with some cells supplied as clues from which logical inferences *[obvious or immediate observations]* can be made about the contents of other cells (see Figure 3.12 for an example of a half-completed Sudoku puzzle). The majority of puzzles can be solved by making logical inferences at every step until every cell is filled. The most difficult puzzles contain steps that cannot be logically infered from the current state, and require trial and error to arrive at a solution—these are usually called "impossible" puzzles [Wik07b].

There are a number of Sudoku programs available, offering large volumes of puzzles and help with solving each puzzle. This "help" usually involves offering a "suggested next move" when the player needs it. This then leads to automated solving of the puzzle. For example, Redleg Sudoku (available from `http://www.redleg.biz`) shown in Figure 3.13 offers two supports for solving Sudoku: click the hint button and the program places a number in a square for you; click the solve button and the program instantly fills in the complete solution. This approach might be suitable for fast solving, but it does not give any indication to the human solver how the

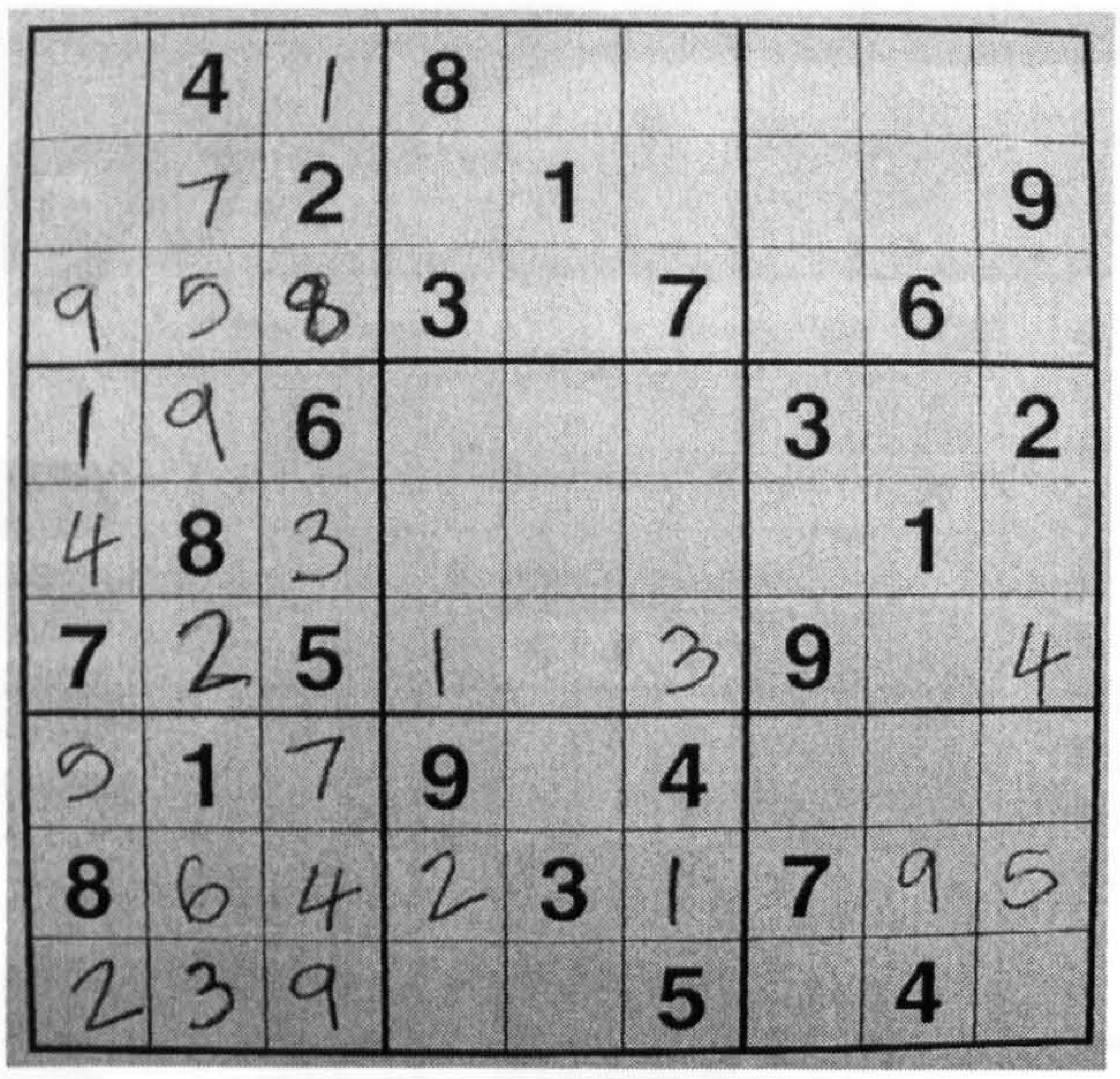


Figure 3.12: A half-completed Sudoku puzzle (from The Guardian [Gua07]).

solution was found. There are several programs, such as Sudoku Puzzle Game and Solver (`http://www.muddyfunksters.com/sudoku/sudoku.htm`), MPS-Sudoku 2007 (`http://www.ltucshop.com`), and DKM Sudoku online (`http://www.dkmsoftware.com/sudoku/`), that offer similar functionality. Although this type of assistance might be useful in some cases, it rarely leads to significant learning about the nature of solving Sudoku puzzles. It is equivalent to a child who asks how to do something, and a teacher who simply does the thing for the child. (The most likely learning outcome is the knowledge that clicking the "help" button will solve the puzzle!) A better example of a program that supports learning is Sudoku Solver by Logic (available from `http://www.sudokusolver.co.uk/step.html`) which offers a list of logic rules which can be applied in turn to solve the puzzle. This is an improvement because at least at each step the human solver can observe which logic rule was used, and information on a separate page gives an explanation of the more complex rules. However, this program is still automating the solving process such that the human solver plays only an observation role in the process. In terms of offering the human solver support for solving, the simplest program with the least automation is perhaps the most effective. In MPS-Sudoku 2007 each cell offers support for recording the potential numbers for that cell. Clicking on one of the numbers crosses it out, thereby eliminating the possibility that the number can be placed in the cell. It is not automated, the human solver must work out which numbers can be eliminated. It is useful though as it offers some

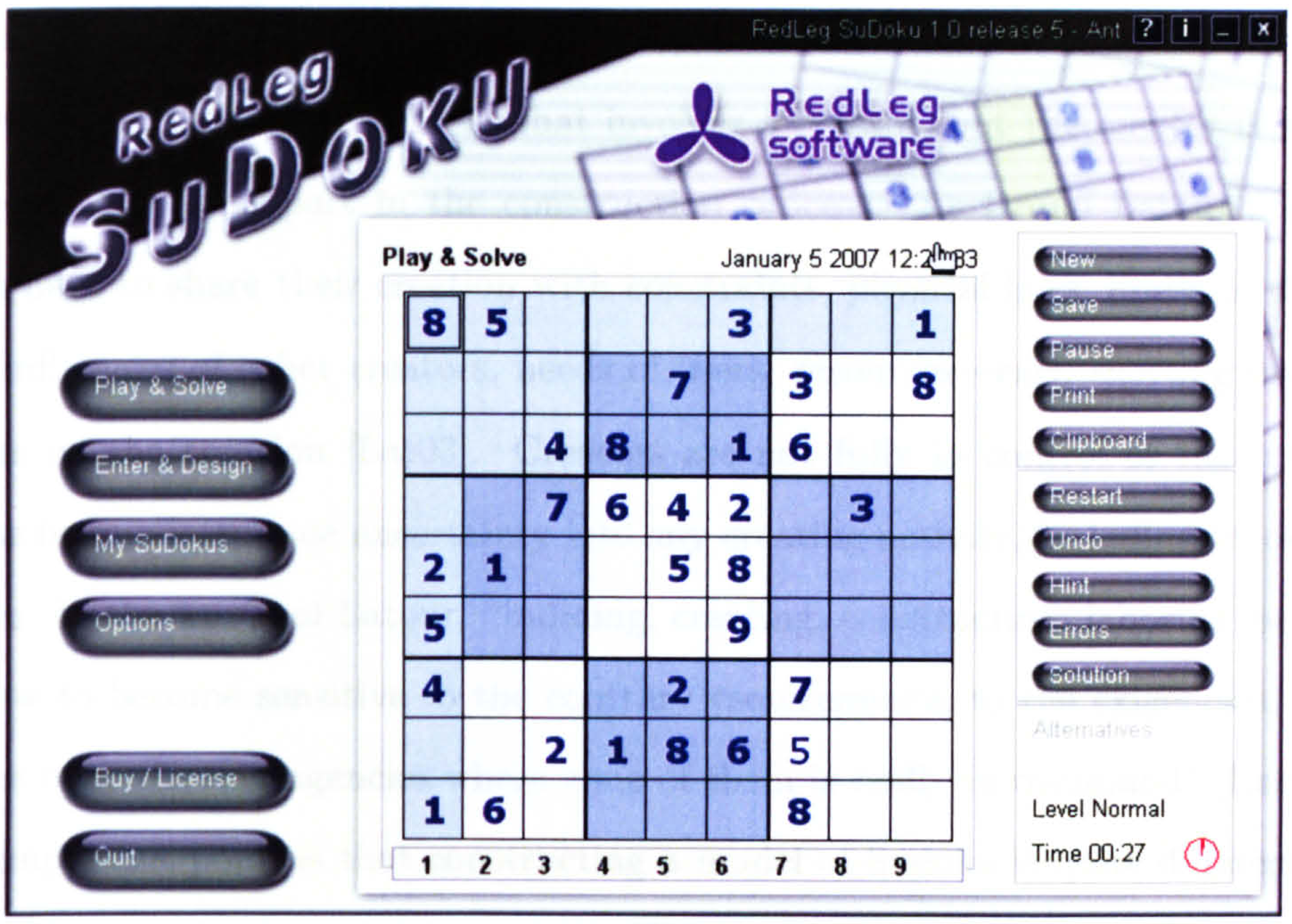


Figure 3.13: A Sudoku program produced by RedLeg Software.

support for solving which is difficult to maintain when solving Sudoku on paper.

3.4.5 Constructing a Sudoku model experimentally

In contrast to the use of a Sudoku computer program, the construction of a Sudoku model adds new dimensions to the learning. In constructing a model there are two aspects to the sense-making outlined in §2.2.3: reinforcing what is already known, and becoming aware of that which was previously unknown. These two aspects are interdependent, and are always found together—when reinforcing what is known, there will be unknowns arising (as discussed in §1.2.3). This should be self-evident when looking at our own learning. Although you would think that we do not always need to reinforce what we already consider to be knowledge, but really, any new learnings must involve that knowledge. Try to use a blank sheet of paper (and no previous knowledge) to learn something new and it will be a struggle. But take a sheet of paper with a Sudoku puzzle and using some knowledge of logical puzzles (even something simple like noughts and crosses) it may be possible to learn something about how to solve Sudoku. The same applies to any type of learning activity, it is usually grounded in something already known—when learning to drive on the road, the learner is generally already aware that cars drive on the left (at least in the UK), that you stop at red lights, that applying the brake helps you slow down or stop, and the learning involves taking this understanding a step further to be able to drive.

Model construction is a process that involves a creator, but the creator is not the only agents to play a part in the construction process. As Bruno Latour suggests, creators have to share their creation with constraints, physical laws, limits of the materials, influences of other creators, needs of users, resources—each exerting opposing pressures on the creation [Lat03]. Creators are not fully in control of the creation. All these factors introduce uncertainty into any creating activity, including model construction. In the words of Latour, "building, creating, constructing, laboring means to learn how to become sensitive to the contrary requirements, to the exigencies, to the pressures of conflicting agencies where none of them is really in command" [Lat03].

An important point is that constructing a model of Sudoku is quite different from the task of writing a computer program to solve (or "help with") Sudoku puzzles. The writing of a program would usually involve a design which may have to be preconceived and thought-out in advance, before implementation is able to commence (as discussed above). The implementation stage would be an exercise in programming, not necessarily in understanding Sudoku. However, constructing a model is about making sense of Sudoku, leaving the potential for exploration of that which is both known and unknown.

3.4.6 Dependency in spreadsheets for meaningful learning

To construct a Sudoku artefact, a spreadsheet can be created using an environment like Microsoft Excel. There are some obvious benefits to using a spreadsheet as the environment is designed for two dimensional grids of cells. First steps in building a model might involve creating a sheet in which numbers can be placed in a 9x9 grid, and then inserting an initial Sudoku puzzle as shown in Figure 3.14—just as you might do in preparing a sheet of paper for playing Sudoku. In this example, the next step that was taken was to add a cell for each column, row, and region to count the number 1s, 2s, 3s, and so on. This adds some guidance to the solver to determine if a number has already been placed or if a number has been placed more than once (i.e. a mistake has been made). Assuming the puzzle is in the range A1:I9, a dependency for counting the number of 1s in the first column is placed in the cell A11 with the definition `=COUNTIF(A1:A9,=1)`. Any changes to the puzzle grid will also be reflected in any cells dependent on the puzzle grid, so if a mistake is made by inserting too many 1s, it will

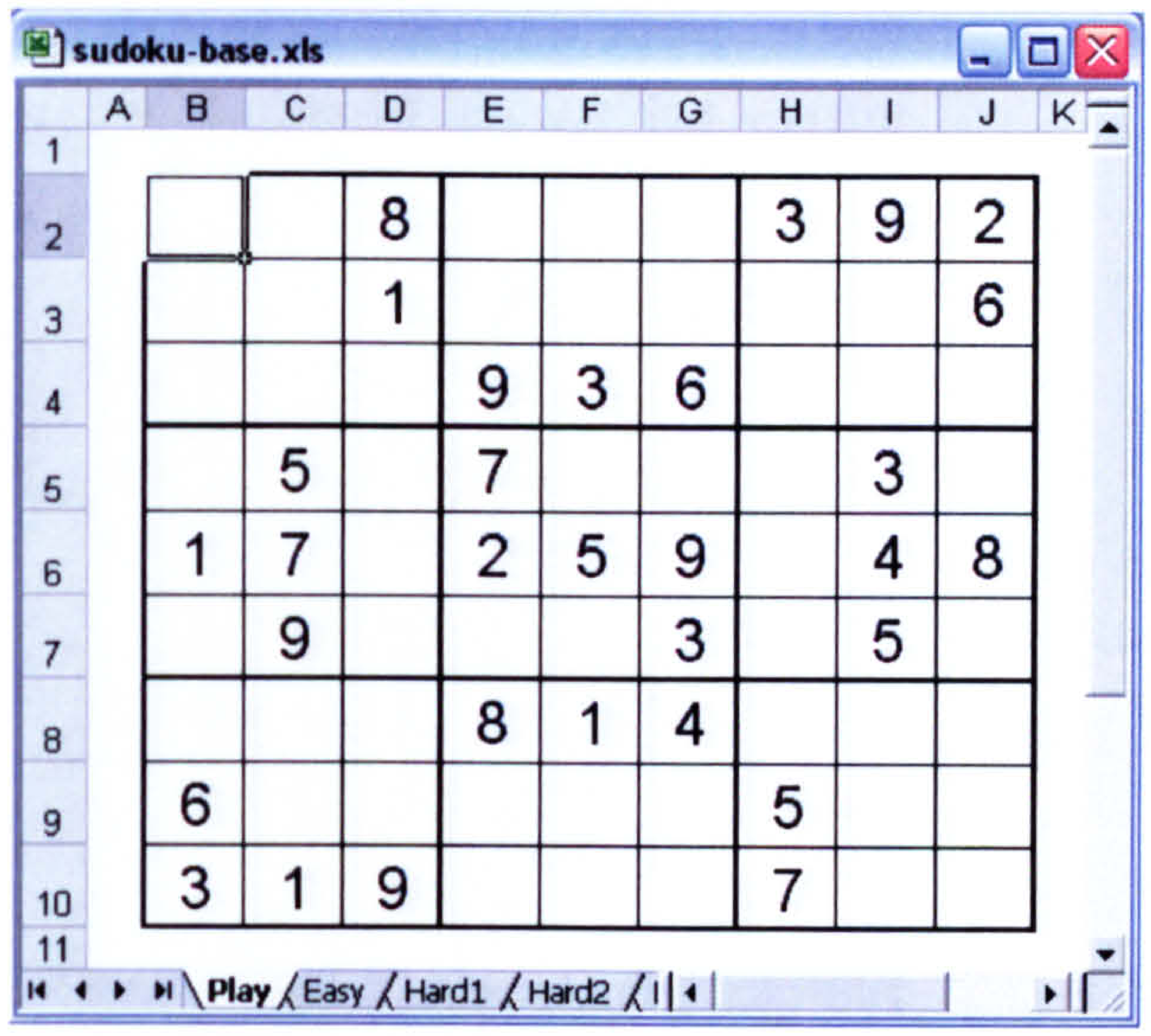


Figure 3.14: An early spreadsheet Sudoku model.

be immediately visible to the solver.

As discussed in §2.2.2, dependency is an important principle in EM. Firstly, it is important because changes that are made to a model are instantly observable. It allows models to be built without a burden on the modeller to maintain the relationship between elements of the model (the observables). Secondly, dependency definitions tell the model-builder about relationships between the observables. Basically, observables refer to things in the world, and dependencies refer to the relations between the observables. An environment like a spreadsheet (or other EM tools) offers the potential for both observables (i.e. values in cells) and dependencies (i.e. formulae in cells) to be changed, and when changes occur (through acts of agency) dependency makes the effect apparent to the modeller consistently throughout the model. [Geh96]

Returning to the Sudoku model, further dependencies were added to describe other features of the current state of the puzzle. Many of the dependencies are built on top of existing dependencies. For example, for each cell we can determine which numbers can potentially be placed using the column, row and region counts defined above. If, when examining an empty cell, the count of 1s is zero for the column, row and region then the cell may potentially be a 1. If we examine the entire puzzle (of empty cells) for every number (1 to 9), then we have a complete map using dependency of which numbers can be placed in which squares. In Figure 3.15, I have made the background colour of each square dependent on whether a '4' can potentially be placed in the square. The model now becomes more like an artefact to support the solving of Sudoku. Whilst not

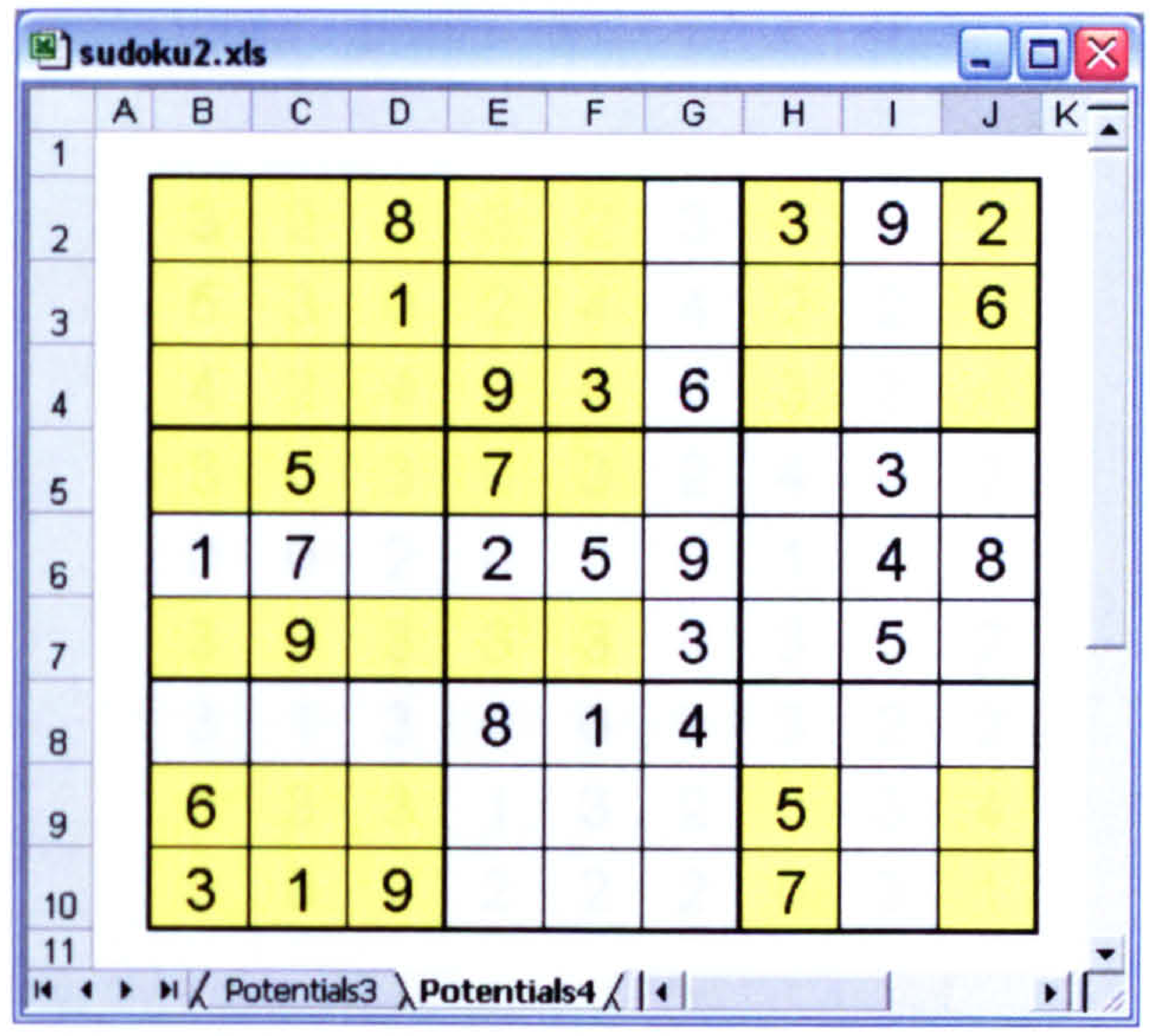
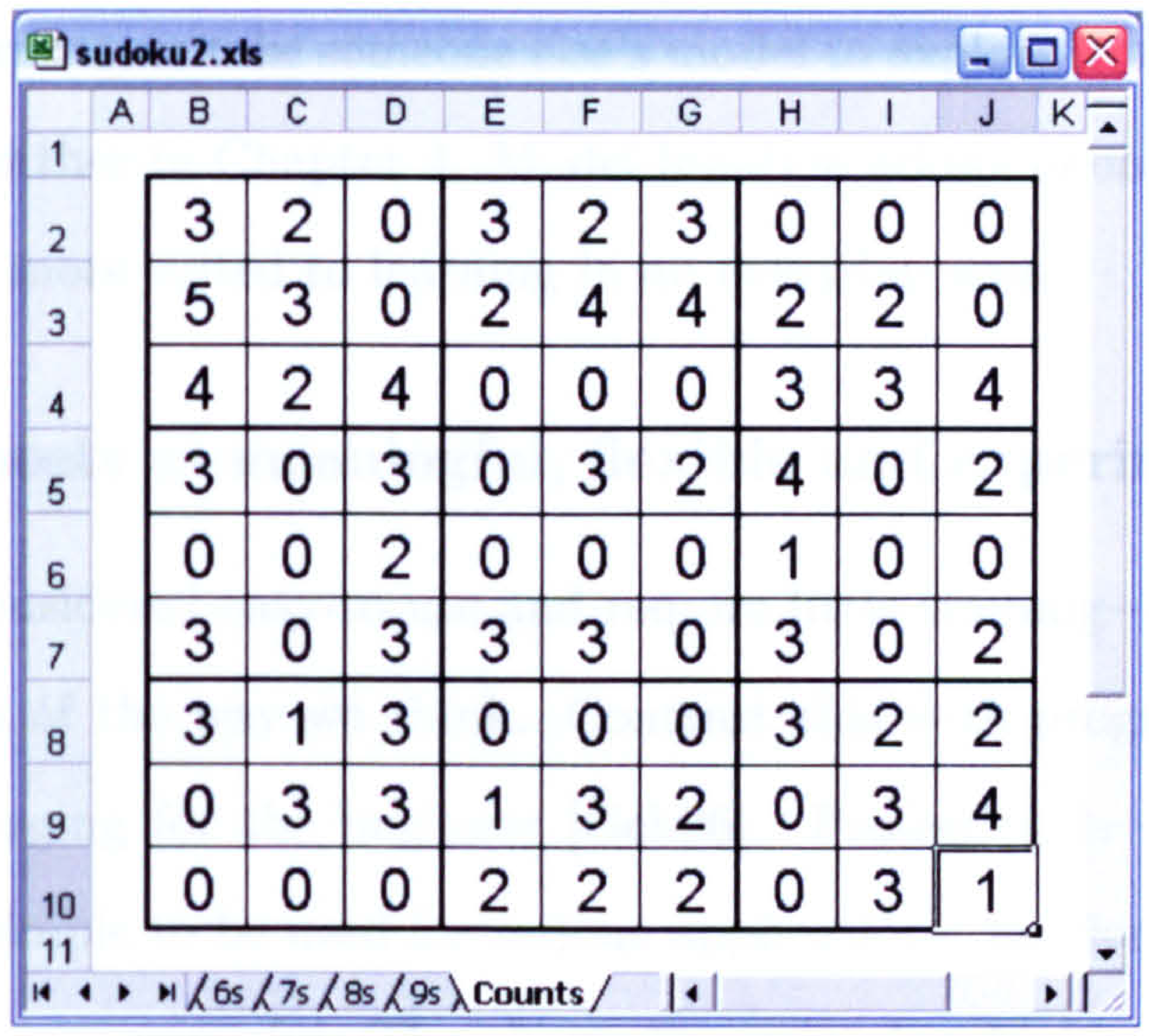


Figure 3.15: Using dependency to highlight the areas where there is potential to insert a '4' in the Sudoku grid.

solving the puzzle, it may give some clues as to where to look to make the next inference in the puzzle. By using dependency we can also deduce why we can or cannot make a particular inference. There is meaning behind the clues, the reason for an inference can be traced from the observables and dependencies relating to the cell—e.g. the definition of the background colour of the cells in Figure 3.15 is dependent on another group of cells determining if there is a '4' in the row, column and area. Compared to the program solvers discussed above which offer the clue without the reasoning, clues in the spreadsheet Sudoku can be investigated, dismantled and manipulated.

By adding more dependencies we might be able to learn more about the nature of solving Sudoku. Using the observables defined above that explain for each cell what potential numbers can be placed, we can create another dependency which counts the number of potential numbers that can be placed. Figure 3.16 shows another layer to the spreadsheet, which counts the potentials from nine sheets like Figure 3.15. With this information, if we find a cell with only one potential number (like the bottom-right square in Figure 3.16), then that cell can only contain one thing and we can make a step by filling this cell with the only possible number (in this case it must be a '4' because Figure 3.16 shows it as a potential, and it must be the only potential because of the bottom-right square in Figure 3.16). Not only is it clear from the dependencies why we can make this step, but it also relates to the way a human may solve the puzzle—i.e. by looking for squares that there is only one possible value to be placed. This is an



3	2	0	3	2	3	0	0	0
5	3	0	2	4	4	2	2	0
4	2	4	0	0	0	3	3	4
3	0	3	0	3	2	4	0	2
0	0	2	0	0	0	1	0	0
3	0	3	3	3	0	3	0	2
3	1	3	0	0	0	3	2	2
0	3	3	1	3	2	0	3	4
0	0	0	2	2	2	0	3	1

Figure 3.16: Using dependency to make a step in solving the Sudoku puzzle.

important feature of EM models, that there is a correlation between the model and the world (or the model-builder's view of the world). The reason this correlation exists is because the model is built on the principles of observables and dependencies that are meaningful connections between the construal and the referent as described in §2.2.2.

3.4.7 The flexible nature of learning using spreadsheets

Another characteristic of the model building in a spreadsheet is the incremental development. As mentioned above, a traditional program is usually prescribed first and implemented second. In model building, the model evolves incrementally as new observables and dependencies are added, or existing ones refined. All this is occurring on-the-fly, in response to changes in our understanding of the domain. We are concerned with the constructing activity only in as much as it enables us to make sense of that which is being modelled. The model need never be considered 'complete' or 'finished'. It is an artefact for making sense, for understanding, and for learning.

The incremental nature of modelling leads to a more flexible approach to learning that is characteristic of lifelong learning where it might be appropriate for the learning to occur over a long period of time, revisiting previous ideas and adding to an existing knowledge of some domain. The incompleteness of models and the potential for continuous refinement of models encourages modellers to revisit models, either to explore the domain in more detail (possibly in light of some new experiences) or make use of the model in a new situation. It is possible to combine existing models, reuse an existing model for a new purpose, or take someone else's model to explore from another perspective, as discussed further in Chapter 4. Model-building allows open-ended exploration of a subject that is more suited to learning in an everyday sense.

3.4.8 Spreadsheets as meaningful, flexible and experimental

Spreadsheets are considered easy-to-use and require little training—they appear to be a natural extension of the way we think. Contrast this with programming, which is conceptually challenging for the beginner [Geh96]. Professionals often say that the spreadsheet is too simple to be used for serious applications, but for the application of understanding it is very powerful. The above discussion has highlighted three reasons why this might be so. Firstly, there is the direct interaction and instant feedback that the user experiences using a spreadsheet, which encourages *experimentation.* The second reason is that the spreadsheet can *flexibly* moulded during an open exploration that is likely to involve mistakes and changes of direction. The third reason is that the values and definitions in the spreadsheet correspond to a situation in the world in the eye's of the model-builder—spreadsheets act as *meaningful* mediators. In EM these are termed 'observables' and 'dependencies', and are what we use to construct models—in the same way as a spreadsheet uses values and definitions to construct models such as the Sudoku example discussed above.

Chapter 4

Learning in Computer Science

This chapter explores the potential for EM to be applied to learning specific topics in *databases*, *computer graphics*, and *artificial intelligence*. EM has grown up in a computer science environment. When the idea first arose in the 1980s, even before the term 'Empirical Modelling' was conceived, it was developed by computer scientists and early applications of EM were in software engineering and concurrent systems [Bey07a]. Later on, the first applications of EM to education were for computer science education. Examples include the EDDI database environment that was used for a number of years to teach relational databases at the University of Warwick [BBRW03] [BCY95]. It is natural that the most extensive account of EM as ET in this thesis comes from computer science. The account in the current chapter demonstrates the *experimental*, *flexible* and *meaningful* strands of the *eight significant characteristics of learning* described in Chapter 1. In each of the following three sections a particular strand is emphasised, although there is some cross-over as the eight characteristics are evident in each of these three learning situations. In particular, the section relating to databases explores EM's support for the *meaningful* characteristics of learning, the section connected to graphics examines the *flexible* characteristics, and the section referring to artificial intelligence is concerned with the *experimental* characteristics.

4.1 Meaningful learning in databases

In this section I will show how EM can offer support for teaching a specific topic in relational databases, namely an algorithm for testing lossless join†. In particular, I

†Previously discussed in a joint paper presented at the International Conference on Advanced Learning Technologies 2005 in Kaohsiung, Taiwan [BH05b] in which my contribution is the construction of

Input: A decomposition of a relation R of attributes $A_1, A_2, ..., A_k$ into sub-relations $R_1, R_2, ..., R_n$ and a set of functional dependencies (FD) for R.
Output: True when $R_1, R_2, ..., R_n$ is a lossless decomposition of R.
Algorithm:

1. Create a table with n rows (corresponding to the sub-relations $R_1, R_2, ..., R_n$) and k columns (corresponding to the attributes $A_1, A_2, ..., A_k$).
2. Initialise the table by filling the cell at row i column j with α_j if the attribute A_j is in the relation R_i else β_{ij}.
3. Repeatedly transform the table until no further changes can be made:
   (a) Consider a functional dependency $X \rightarrow Y$
   (b) If two or more rows of the table have matching attributes of X then transform their Y attributes to match:
      i. If α_i is on one row, put α_i in the other rows.
      ii. If β_{ij} is on one row, put β_{ij} in the other rows.
4. If there is a row containing all α's $(\alpha_1, \alpha_2, ..., \alpha_k)$ then the decomposition of R into $R_1, R_2, ..., R_n$ sub-relations has a lossless join.

Figure 4.1: The Testing Lossless Join (TLJ) algorithm.

highlight EM's support for the characteristics of the meaningful strand (§1.2.6–§1.2.8) of the eight significant characteristics of learning outlined in Chapter 1.

4.1.1 An algorithm for testing lossless join

In relational database design, as described by Ullman in his book *Principles of Database Systems* [Ull82], it is important to be able to determine whether a decomposition of a relation R into two or more smaller relations $(R1, R2, ...Rk)$ results in a loss of information (which would usually be undesirable). A particular decomposition has a lossless join if, for any given extension r of R satisfying all the functional dependencies (FDs) that hold in R, the natural join of the projections of r onto each of the k subschemes is r itself [Ull82:p227]. The Testing Lossless Join (TLJ) algorithm, as specified by Ullman [Ull82:p227], is a standard component of the relational database theory that can be used to find out if a particular decomposition is lossless. The algorithm is described using pseudo-code in Figure 4.1.

For the purposes of this exposition, without loss of generality, all FDs will be as-

the TLJ model.

sumed to be of the form $X \rightarrow S$, where S is a single attribute. The representation of the entries in the table can also be simplified so that they have numeric values. Specifically, *a*s and *b*s can be represented by integers: each a by 1 and the initial b entries by integers greater than 1. When several values are to be equated, it is then appropriate to equate all values to the least. This representation, which is suited to computer implementation, is valid since the indices of *a*s and *b*s are redundant, and all comparisons are made between elements in the same column.

4.1.2 Educational technology for the TLJ algorithm

The TLJ algorithm is in some respects a natural target for computer support. For instance, a student who is exercising the algorithm (or a teacher who is demonstrating the algorithm) typically indicates the successive modifications that are made to the array by crossing out entries and inserting their new values until such time as the array entries become difficult to read, then making a new copy of the array and repeating the annotation process. This is an error-prone process that does not always give a clear indication of the precise steps carried out or the reasoning behind each step. It is easy to see that, when we consider the possible motivations and issues that arise in learning, presenting or assessing the TLJ algorithm, the list of requirements becomes very large. The teacher alone will typically want: a dynamic way of presenting the algorithm that draws attention to the specific observations and actions that are being carried out at each step; to be able to simulate the operation of the algorithm in full; to be able to experiment with different sets of FDs, perhaps even whilst the algorithm is being executed; to emulate errors that a student might make in exercising the algorithm; to use the model as the basis for exercises that test a student's understanding as comprehensively as possible. In devising exercises or an examination question, the teacher will not wish to restart the algorithm from scratch at each new iteration required in the design. The range of situations in which the teacher could exercise the algorithm is vast, and a complete description of all the situations is impossible to specify prior to interaction. Unless the teacher enjoys the dedicated support of a developer, they will ideally want to be able to adapt the model relatively painlessly themselves to take account of different perceptions of what the student requires, and of any special, possibly idiosyncratic, misunderstandings they have.

As a requirement for a conventional program, this presents a formidable challenge. What is more, it is quite apparent that the requirement is not in any sense complete. In constructing a conventional program to meet these needs, there will invariably be optimisation for specific purposes that will prove obstructive to future extensions. The key to addressing this problem is to recognise that what is required of a model to support the learning of the TLJ algorithm is a form of automation that can integrate fully with the activities that the teacher can perform manually as the need arises. The teacher (as a learner) should be able to attend to personal interests or concerns (§1.2.6), new situations or contexts as they arise (§1.2.7), and previous experience as it continually evolves (§1.2.8). In effect, this allows human discretion and intelligence to be exercised in situations where there is no satisfactory preconceived fully automated solution that can be applied. This is the function of the EM construal for the TLJ algorithm to be described below.

4.1.3 EM support for the TLJ algorithm

The primary observables in the TLJ algorithm are the contents and attributes of the table that is generated in executing the algorithm and the FDs that are associated with these attributes. Both teacher and student come to understand the algorithm in terms of just these observables; building a construal to embody these observables, and the patterns of dependency and agency to which they are subject, is also a most appropriate way for the developer to provide support for the manual, semi-automated or fully automated interaction that must accompany the learning of the algorithm.

Learning the TLJ algorithm is linked to a pattern of observation that applies at each iteration. The learner consults the current state of the table with a specific FD $X \rightarrow S$ in mind, observes the pattern of tuples that arises in the columns associated with the left-hand side X of the FD to detect where there are duplicates, then observes how this pattern applies to the column associated with the right-hand side S of the FD. The core step of the algorithm is the substitution of the resulting transformation of the column associated with S for the original column.

For a particular table and FD, the above ingredients of the core pattern of observation can be displayed pictorially as in Figure 4.2. The arrows in this figure represent dependencies between observables, expressing the way that a given state of the TLJ

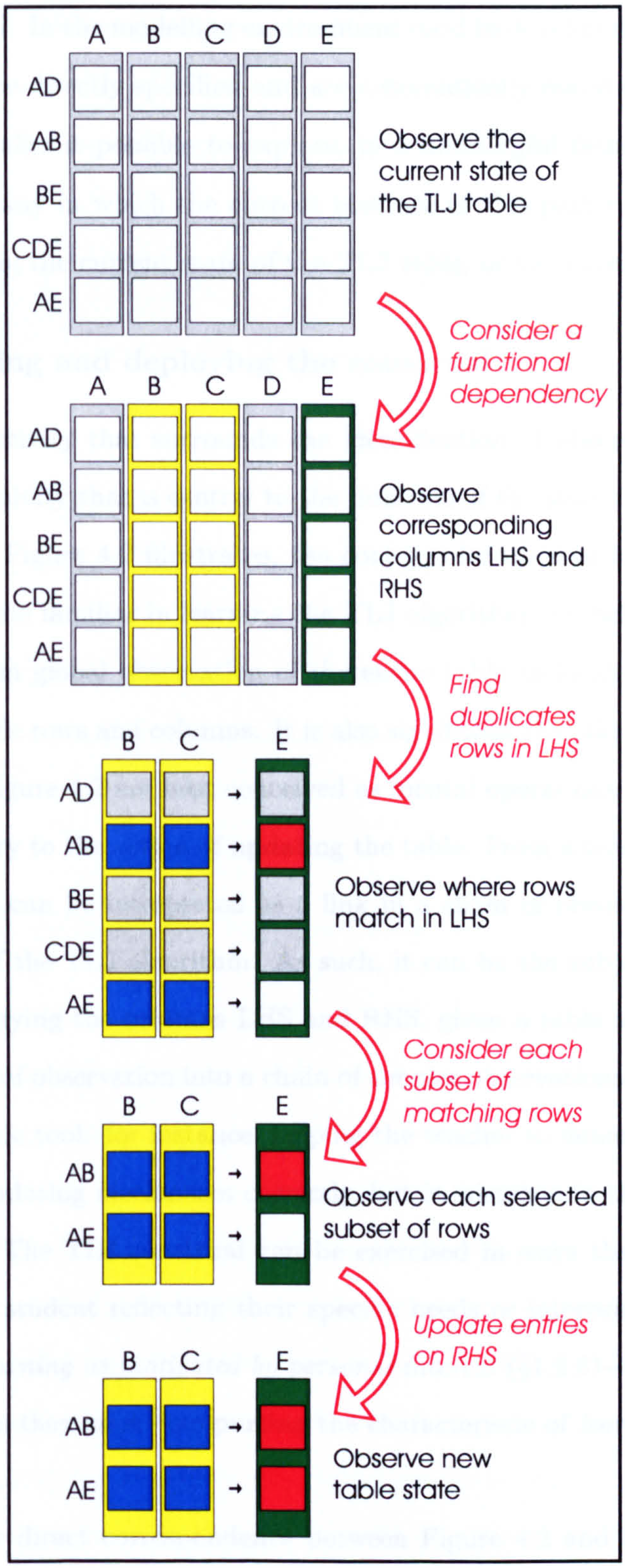


Figure 4.2: Steps in the TLJ algorithm.

table, and a given FD determines the set of columns LHS and a column RHS, and how the duplicate rows in the set of columns LHS then determine the updated entries in the column RHS. In the modelling environment used to develop the construal, these dependencies can be directly specified and are automatically maintained (as discussed in §2.2.2). This makes it possible to explore, in a meaningful fashion by tracing the dependencies, the way in which the current instance of this pattern of observation is affected by changing the current state of the TLJ table, or the current FD.

4.1.4 Developing and deploying the construal

The exploratory activity that surrounds the identification of observables and dependencies is a core activity that is central to the interests of the student, the teacher and the developer. As Figure 4.2 illustrates, the contexts for observation with which the student must become familiar in learning the TLJ algorithm are rich and subtle: they involve moving from global observation of the entire table to localised observation of the entries in specific rows and columns. It is also significant that the activities denoted by the arrows in Figure 4.2 are best conceived as mental operations on the part of the student, preparatory to the action of updating the table. From a teacher's perspective, each of the arrows can be interpreted as a link in a chain of observation involved in executing a step of the TLJ algorithm. As such, it can be the subject of an exercise: for instance, identifying the columns LHS and RHS, given a table and a FD. Decomposing the pattern of observation into a chain of simpler observations also has potential value as a diagnostic tool: for instance, helping the teacher to detect where a student understands the updating mechanism correctly, but is mistaken in their interpretation of a FD relation. The TLJ construal can be exercised in ways that are personal to the teacher or the student reflecting their specific needs or interests—supporting the characteristic of *learning as motivated by personal interest* (§1.2.6)—and in new situations or contexts as they arise—supporting the characteristic of *learning as a situated experience* (§1.2.7).

There is a very direct correspondence between Figure 4.2 and the EM construal for the TLJ as that depicted in Figure 4.3. This correspondence is best appreciated by interacting with the `tkeden` environment, but it is to some extent apparent from the relationship between Figure 4.2 and Figure 4.4. Just as the pattern of observation

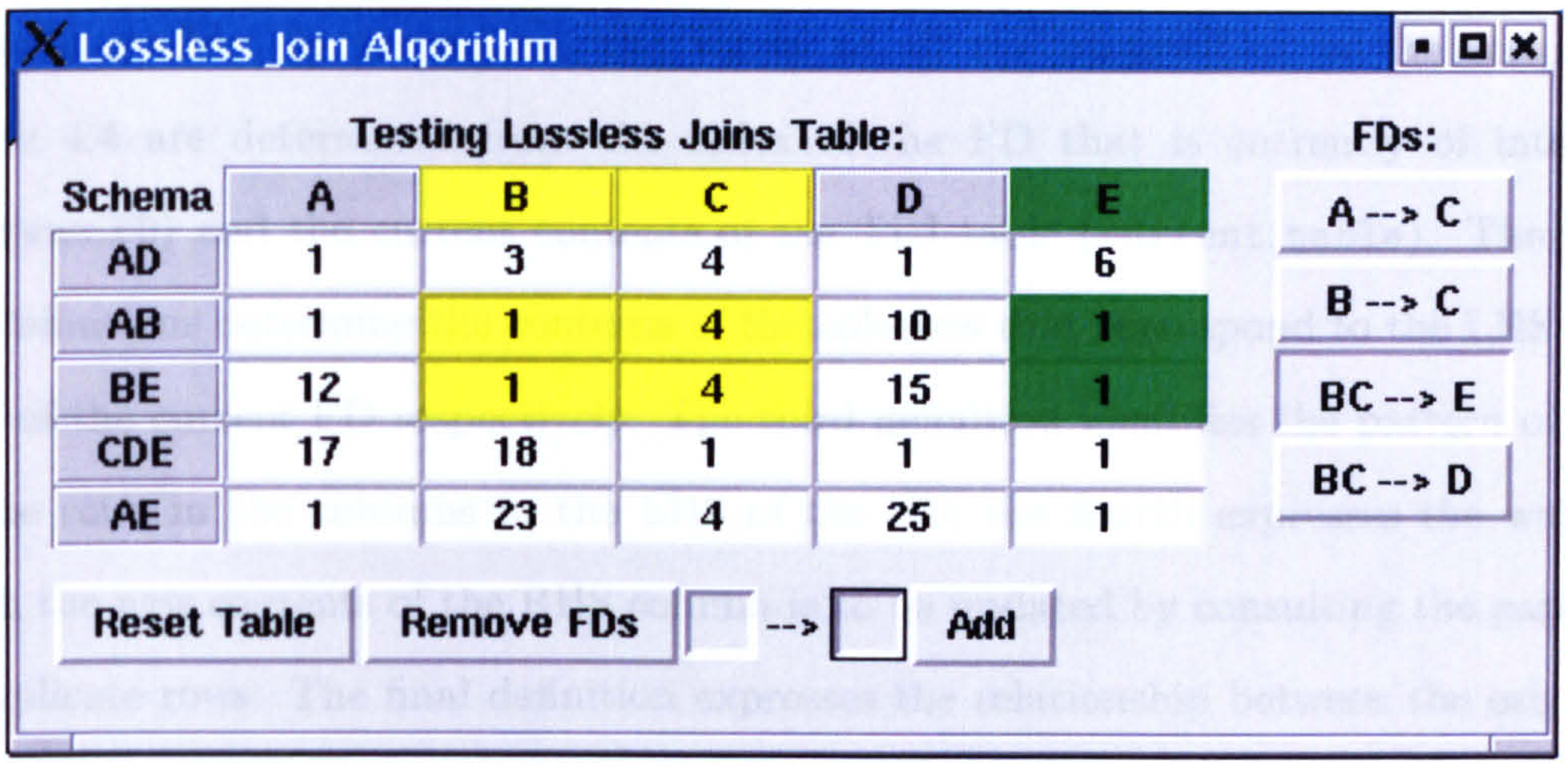


Figure 4.3: The TLJ construal.

```
project_table_LHS_FD is project(current_table,
    makestrlist(FDs[current_FD][1]));

project_table_RHS_FD is project(current_table,
    [FDs[current_FD][2]]);

pattern_duplicate_rows is index_duplicated(
    tail(project_table_LHS_FD));

newcol is transformcol(makelistcol(project_table_RHS_FD),
    pattern_duplicate_rows);

newtable is apply_current_FD_current_table(current_table,
    newcol);
```

Figure 4.4: Observables and dependencies in the TLJ construal.

depicted in Figure 4.2 is the core of the TLJ algorithm, so the script of five definitions linking observables and dependencies in Figure 4.4 is the core of the TLJ construal. The names of the observables in Figure 4.4 have been made more expressive, and the code for operators (such as index_duplicated, and makelistcol) has been omitted, but the definitions are essentially as they appear in the **tkeden** script. The correspondence between Figure 4.2 and Figure 4.4 demonstrates the extent to which an experience of the algorithm can be loosely connected to an experience of observing the dependencies in the TLJ construal. EM's support for *learning as a continuous experience* (§1.2.8) can be recognised from the necessary close correspondence between observations in the world and observations in the construal. In order to further illustrate the use of EM for liberating the *meaningful* characteristics of learning, a brief explanation of how this script was developed, and relates to the pattern of observation in Figure 4.2, is appropriate.

As is evident by inspection, the values of all the observables in the script in Figure 4.4 are determined from the index of the FD that is currently of interest (`current_FD`) and the current contents of the TLJ table (`current_table`). The first two definitions determine the contents of the columns that correspond to the LHS and RHS of the current FD respectively. The third definition identifies the pattern of duplicate rows in the columns in the LHS of the FD; the fourth expresses the way in which the new contents of the RHS column is to be updated by consulting the pattern of duplicate rows. The final definition expresses the relationship between the original value of the table and the value that it takes after the FD has been processed. These definitions correspond closely to the links in the pattern of observation in Figure 4.2: in establishing the definitions using the `tkeden` interpreter, the operators introduced to specify the relationship associated with each link are tested in isolation by supplying different test values for the parameters in much the same way that the student might confirm that they have understood each observational link in mastering the algorithm. Though the development of a script may seem to have more of the characteristic flavour of conventional programming, it remains anchored to the learning domain. The missing elements of the `tkeden` script are the specifications of the operators themselves, which take the form of rather straightforward procedural code to compute an output from an input without side-effect. The script illustrates other features that are of interest from a computational perspective. These include:

- the re-use and adaptation of standard operators (such as the relational operator project, borrowed from the relational database extension of `tkeden`).

- the use of definitions to maintain dependencies between different modes of observation that are a common concern for traditional programmers, namely those that are associated with two or more data structures for a particular application (such as the conversion function `makelistcol`).

For the experienced developer using `tkeden`, the model-building task is greatly simplified by a combination of these three techniques: creation of relatively simple functions without side-effects; re-use of existing functions and scripts; and the use of definitions to maintain many different consistent concurrent representations of a given family of observables. The interface shown in Figure 4.3 was constructed using observables and dependencies in much the same way as the explicit TLJ-related observables in Figure 4.4. The GEL (Graphical Environment Language) notation [EMP:gelHarfield2007], developed by the author, was used to construct the buttons and labels which for the purpose of this model were dependent on the FDs and the current_table observables. The buttons semi-automate the actions that a student may be interested in performing, such as changing the current FD or changing the set of FDs (corresponding to a modification of the current_FD or FDs observables that are used in Figure 4.4). Figure 4.5 introduces the basic principles of GEL, and further information can be found in the GEL documentation [EMP:gelHarfield2007].

4.1.5 Implications for learning

In the above discussion of the TLJ construal, it has been shown that EM offers a suitable approach to technology-enhanced learning according to the three characteristics of the *meaningful* strand of learning as set out in Chapter 1, that is:

- learning is motivated by personal interest (§1.2.6);
- learning is a situated experience (§1.2.7);
- learning is a continuous experience (§1.2.8).

The teaching of a specific topic in databases such as the TLJ algorithm requires support that is sensitive to the personal needs or interests of the teacher or the student, and to the variety of situations in which the algorithm can be used for learning. In some cases the teacher may be interested in exercising the algorithm very general ways as a device in a lecturer, and in others the teacher may exercise a very specific element of the model for the benefit of a particular student to rectify a misunderstanding. Furthermore, the TLJ construal can be seen as a device for navigating experience according to the EFL as described in §2.2.8. The difficulties which learners encounter with the TLJ algorithm involve a mismatch between formal knowledge of the algorithm (the bottom of the EFL) and practical experience of the algorithm (the top of the EFL). EM supports *learning as a continuous experience* (from the top of the EFL to the bottom) because artefacts such as the TLJ construal enable learners to exercise the algorithm on a practical level—personally making the interactions involved—and on a formal level—observing the dependencies leading to state-change.

The Graphical Environment Language: GEL

One of major technical contributions to this thesis is the creation of the GEL notation [EMP:gelHarfield2007]. GEL is a notation that has been added to the `tkeden` environment enabling the creation of new graphical environments or graphical user interfaces. GEL itself is a model that was created using the agent-oriented parser (AOP) inside `tkeden`. The AOP is another of my contributions to the EM project [EMP:agentparserHarfield2003] that can be used to create new notations. The GEL notation is open to the same exploration and experimentation as other models, including models created using GEL. Much of the development of the GEL notation occurred on-the-fly whilst I was using the GEL notation.

GEL can be used to create graphical environments that contain general GUI components such as windows, labels, buttons, list boxes, option buttons, radio buttons, scroll bars and scale bars, and `tkeden` specific components such as SCOUT or DoNaLD windows. GEL is a dependency-based notation, enabling complex relationships to be modelled and maintained among elements of the model. The example of a GEL script below demonstrates some simple components and the use of dependency.


```
myexample = window {
        title = "Listbox Example";
        content = [mylistbox] };

mylistbox = listbox {
        selectmode = "browse";
        items = [ "blue", "red", "green", "yellow" ];
        selecteditems = [ "red" ];
        height = mylistbox_items#;
        background = mylistbox_selecteditems[1] };
```

The first definition describes `myexample` as a window with a title and containing something called `mylistbox`. Entering this definition into the `tkeden` interpreter will create a new window on screen. By dependency, when `mylistbox` changes, the window will be updated. The second definition describes a listbox with initially four items. Entering this definition will create the listbox inside the window. The height of `mylistbox` is dependent on the current number of items, therefore if another item is added by redefining `mylistbox_items` then the listbox height will change and also the window will be updated. The final line in the definition of the listbox describes the background as dependent on the first selected item in the listbox. Thus, if the modeller selects the 'yellow' item then the background of the listbox becomes yellow, as shown above.

The GEL notation has been used to construct a number of models in this thesis. Further information on GEL can be found in the documentation and in an interactive introductory model [EMP:gelHarfield2007].

Figure 4.5: Description of GEL.

The difficulty of unifying the roles of *student*, *teacher* and *developer*, as discussed in §3.3, is one of the obstacles to a constructionist approach to technology-enhanced learning. The technical problems of supporting the degree of openness in interaction that constructionism ideally presumes are so acute that Ehrmann [Ehr00] has been led to question whether the vision of learners constructing their own learning environments is a mirage. It is clear that in activities such as developing micro-worlds for children–at any rate with current software tools—there is little prospect that the learners can themselves carry out the model construction. The TLJ case study is of interest because it proves that in principle there can be a high degree of synergy between interactions that are demanded of the learner in the roles of student, teacher and developer. For the target group of learners (viz. undergraduates with high levels of programming skill following an advanced module in database theory), there is no great conceptual or practically significant distinction between the kind of activity involved in learning about the lossless join algorithm and that involved in constructing the associated EM construal. It remains to be seen to what extent, subject to appropriate tool refinement and suitable training in the application of EM principles and tools, the same synergy between learning and model-building can be demonstrated in other learning contexts.

4.2 Flexible learning in computer graphics

In this section I will show how EM tools and principles can offer support for the *flexible* characteristics of learning discussed in Chapter 1, in the context of teaching a specific topic in computer graphics. The characteristics of 'learning occurring without a prescribed outcome' and 'learning not following a preconceived path' are evident in EM because the learner is a model-builder who can: reuse models in different contexts; combine models together; take on the unified role of student, teacher and developer. The models explored in this chapter have led to one of my main technical contributions which is the construction of a generic presentation environment for EM.

4.2.1 Flexibility through model reuse

Code reuse is important in programming when you have two or more tasks that are similar or the same and therefore a part of the code can be shared or reused. In model-building, reuse can have a slightly different meaning. Model reuse involves taking an existing model and fitting it to a new context (as introduced in §2.2.7). The fitting often entails the extension or reduction of the model. Similarities can be drawn with Piaget's theory of learning [Pia50], discussed in §1.2.2, as necessarily involving assimilation or accommodation, where a new experience is fitted to existing knowledge, or knowledge is modified to fit a new experience. In the case of model-building, either the model can be modified to fit a new context or the model-builder's imagination may have to be stretched or modified to fit the limitations of an existing model. A good example of reuse from the EM archive is the jugs model [EMP:jugsBeynon1988] which has been reused in a wide variety of contexts. Initially developed as an artefact for explaining the concept of dependency, it has subsequently been used, as discussed in §3.2, for modelling queues at a bar, learning a language, and displaying chords on a violin [EMP:kaleidoscopeBeynon2005], as well as the basis for a small concretisation case study[†].

4.2.1.1 The 3D room viewer

A group of models relating to a simple room demonstrate an example of model reuse over a long period of time. Each of the models is based on the idea of modelling a room containing a table, a lamp on the table, and various other objects. The model demonstrates dependencies such as the position of the lamp being related the position of the table. The original room was the first model that used the line drawing notation DoNaLD by Edward Yung [Yun90] [EMP:roomYung1989] in 1989 and remains a popular model for introducing dependency. It was further refined in 1991 by Simon Yung and integrated with the SCOUT notation which added viewport capabilities that enabled the room to be embedded in a viewer model [Yun93] [EMP:roomviewerYung1991]. Both models are 2D top-down views of a room. The focus of this chapter is the 3D room viewer which enables the room to be viewed from any angle as a 3D line drawing with perspective (see Figure 4.6). The 3D room viewer was a 3rd year project by Andy MacDonald in 1997/8 [Mac98] [EMP:room3dMacDonald1998]. Beyond the room viewing aspect, the model gives the user the ability to apply forces to the ob-

[†]This was part of a joint paper entitled *Varieties of concretisation: an illustrative case-study* that I presented at the Baltic Sea Conference on Computer Science Education (Koli Calling 2005) [BHJ05], but it is not discussed further in this thesis.

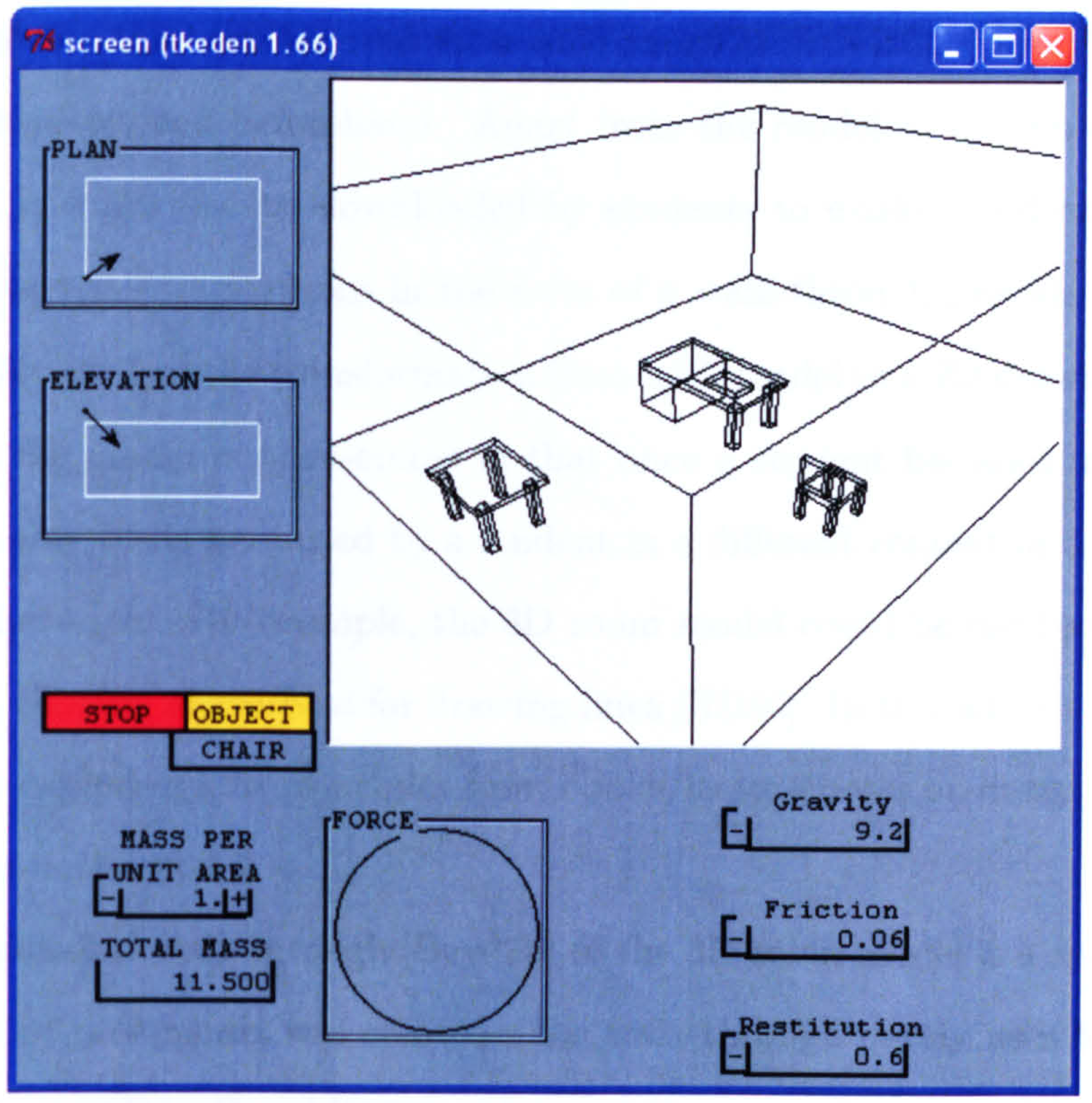


Figure 4.6: The original 3D room viewer model.

jects in the room using the circular force control in Figure 4.6. The 3D room viewer also drawns on an earlier model by Richard Cartwright for creating 3D line drawings in DoNaLD [BCC96]. When the SASAMI notation was developed for creating 3D artefacts in OpenGL, another 3D room model was constructed by Carter [Car99] [EMP:room3dsasamiCarter1999]. Figure 4.10 on page 105 illustrates the history of the room model to be explored further in this section.

In 2007, the 3D room viewer [EMP:room3dMacDonald1998] was used by Meurig Beynon for demonstration purposes in a computer graphics module for undergraduate computer scientists [Bey07b]. Apart from the reuse of the previous room viewer models, this shows a concrete example of a reused model that can perform a function in a completely different context to that initially envisaged by MacDonald in his 3rd year project [Mac98]. MacDonald was interested in applying forces to furniture in the room, whereas for the purposes of Beynon's lectures such facility was less relevant than the part of the model concerned with presenting a 3D scene. In the lectures, the model was used to explore issues surrounding the transformation of a 3D scene into a 2D view through experiments with definitions connected to the projection function. Such reuse is behind the characteristic of EM activity as not being concerned with prescribed outcomes for models—they can be used for a wide range of activities leading to outcomes that were not prescribed beforehand. Apart from the model's function as a demonstration tool, it could also be downloaded by students to explore and experiment by themselves. Beynon's interactions in the form of a walk-through provide a basis from which to flexibly explore the transformation from a 3D model to a 2D drawing [Bey07b]. Furthermore, the model is open-ended in that once a student becomes more familiar with it, the model could be reused by a student in a different context in order to make sense of another topic. For example, the 3D room model could be used to experiment with different clipping algorithms for drawing lines [HB86]. In this way the student is a model-builder exploring how principles from books/lecture notes on computer graphics relate practice in the world.

Beynon's detailed walk-through [Bey07b] of the 3D room model is a special type of reuse because no new model was created—the walk-through purely uses MacDonald's model [EMP:room3dMacDonald1998]. Many of the examples of model reuse in EM have involved fairly trivial models (e.g. [EMP:jugsBeynon1988], [EMP:roomYung1989]) and possible reasons for this may be that complicated models require too much effort on the part of the new model-builder in order to understand the model. Beynon's walk-through is unique in that it provides material, in the form of example interactions, to prompt the discovery of interesting aspects of the model, specifically in this case for understanding a topic in computer graphics. In some respects this has enabled the type of learning that *need not have a prescribed outcome* and *does not follow a preconceived path.*

4.2.1.2 Contrast with the 3D Dino Viewer tool

Previously, another tool was used for demonstrating 3D to 2D viewing transformations in the computer graphics module. The 3D Dino Viewer tool, developed by Abhir Bhalerao [Bha06] using C/C++/OpenGL, was designed to demonstrate the transformation from a 3D scene (of a simple dinosaur) to a 2D view. The tool allows the learner to explore the effects of changing the eye position, the eye direction and the front and back clipping planes (see Figure 4.7). The results of these changes are displayed in two 3D views: from the perspective of the eye, and from the perspective of an external observer. The tool makes use of dependency to maintain a consistent state for example

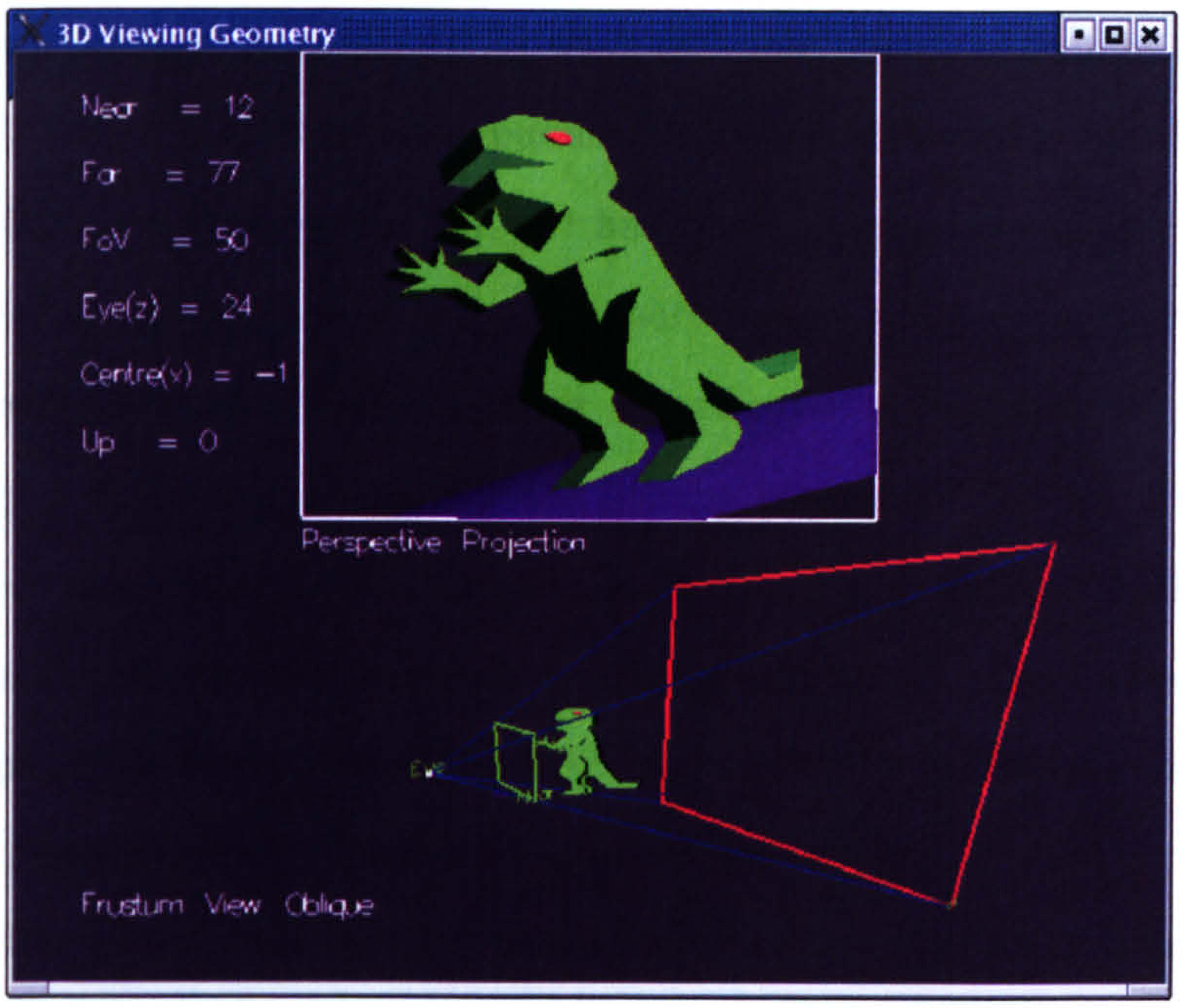


Figure 4.7: A C/C++/OpenGL program for exploring 3D to 2D transformation.

between the values of the viewing plane vector and the 3D view of the scene. In some ways, this is an example of Empirical Modelling principles as there is a direct dependency between the values of the transformation parameters and the two OpenGL views. The tool has some of the qualities of a spreadsheet discussed in §3.4 where changing the values in a cell causes related cells and graphs to be updated simultaneously and automatically.

The 3D Dino Viewer is well aligned to the material in the module and the specific topic of 3D to 2D transformation in computer graphics [HB86]. In terms of features and usability, the artefact is an excellent example of educational technology for communication purposes. It can be used by a lecturer for demonstration and by students for post-lecture experimentation. It is particularly easy-to-use in terms of exercising functionality, much easier at least than the 3D room viewer models described above. Part of the reason for this is in the way that it was implemented. Bhalerao developed the 3D Dino Viewer with one particular task in mind: demonstrating 3D to 2D transformations. As with most programming tasks, the closed-world approach to the implementation allows the programmer to tune the application for efficiency and ease of use. This results in an easy-to-use artefact with a very specific use. Herein lies a particular issue that stresses the paradigm conflict between learning in the world and learning through computers: the artefact is so easy to *use* that there is no motivation for learners to *experiment* with it in ways that may lead to learning spanning the depth of the EFL (e.g. a continuous experience from practical understanding to formal knowledge). Moreover, because the artefact has been made so easy-to-use with specific functionality, it has certain prescribed behaviours that would be *too difficult* for most learners to flexibly *extend* or *reuse* without an extensive knowledge of C/C++. This means that there is little potential for the learner to explore the topic of computer graphics further using the 3D Dino Viewer as it forces a predefined path of interaction and a specific outcome for learning. Learning with the 3D Dino Viewer follows a preconceived path and has a prescribed outcome, and therefore it does not qualify for satisfying the *flexible* strand of characteristics set out in Chapter 1.

The idea that a computer-based artefact might be *too easy-to-use* seems contrary to the common 'push' to make computers more accessible. I am not advocating that computers should be more difficult to use, but that we should not sacrifice imagination and flexibility for the sake of usability. In many cases computers can be made more user friendly, and this should be encouraged, but not to the detriment of what can be achieved with computers. In terms of learning, the idea of making things 'easier to use' in computing often gives rise to the idea that educational technology can make subjects and skills 'easier to learn'. As Jonassen [Jon06] argues, good communication of information does not necessarily lead to learning, and 'easy-to-use' artefacts do not readily imply 'easy to learn' subjects and skills. Can you imagine an easy-to-use piano that you could learn to play adeptly after using on your first time? (This seems to be what some educational technology aspires to do!) The truth is that learning to play any musical instrument is difficult and requires skill, effort, perseverance, and practice. The motivation and satisfaction comes from progression, it comes from getting it wrong many many times, being flexible in ways of use, and discovering how to get it right. When learning a musical instrument, there is this depth to the learning. If something is too easy then it does not offer the same learning experience as an experience that has involved a deeper understanding.

If we 'let go' of the technologist's agenda of making computer-based artefacts easy-to-use then learners might progress to a deeper learning. For educational technology that supports *experimental*, *flexible* and *meaningful* learning technologists should be striving for environments where models can be constructed and manipulated. Such

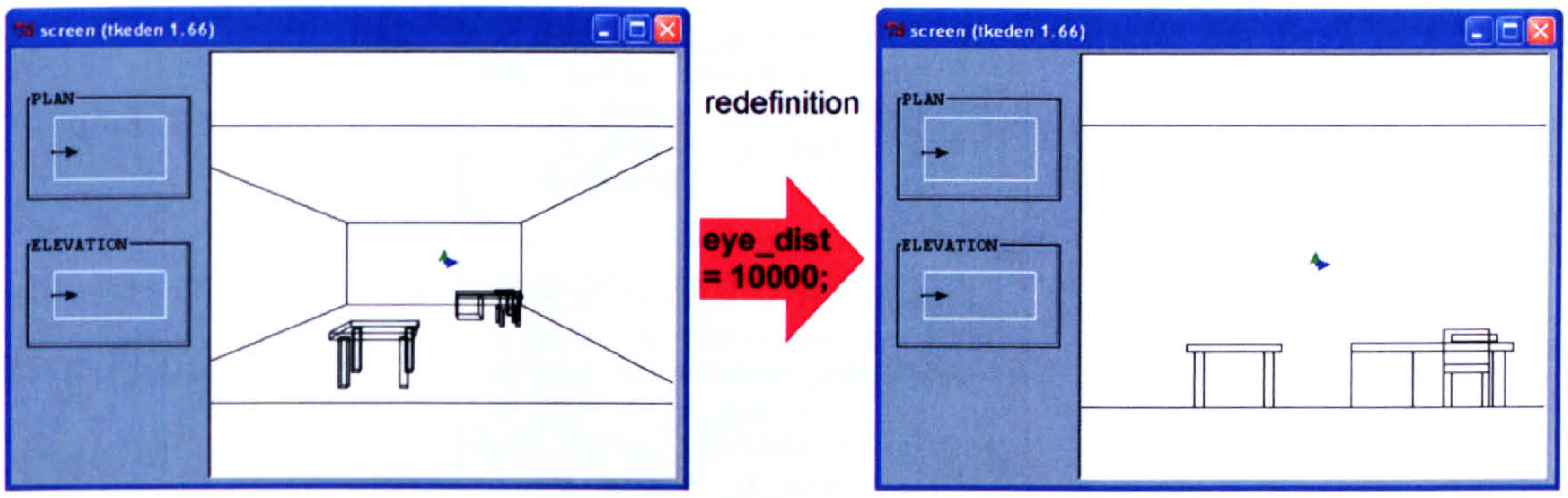


Figure 4.8: Redefinition to change the projection from perspective to orthogonal.

sentiments are expressed by Jonassen that if we cannot construct a model of something then we do not understand it [Jon06]. In the general terms of modelling, as Morgan & Morrison point out in *Models as mediators* [MM99], model use is much less helpful than model construction and experimentation.

Although the EM 3D room viewer may be more difficult to use, if the learner has the motivation to attempt to make sense of the model then it provides a much richer environment for learning about viewing transformations. The evidence for this is in Beynon's walk-through of the model [Bey07b], that shows how the model can be exercised in many different ways to highlight issues relevant to geometric constructions. In one part of the walk-through, the 'project' function is dissected to try to better understand what it does. Small changes to different definitions related to the 'project' function are suggested to the model-builder in order to observe the effect. For example, by changing the eye distance in the project function to a negative number, the camera view can be transformed so that it is inverted [Bey07b]. A more useful redefinition, as depicted in Figure 4.8, would be to make the eye distance very large which imitates orthogonal projection, as opposed to the original perspective projection [Bey07b]. Figure 4.8 demonstrates that such changes are not necessarily easy, a redefinition typically requires: a knowledge of the observables in the model (e.g. `eye_dist`), an appreciation of the current state of the model (e.g. `eye_dist = 100`), an understanding of the perceived dependencies in the model (e.g. `eye_dist` is related to the 3D drawing of the room), and an ability to modify state in the required notation.

The 3D Dino Viewer tool designed by Bhalerao already includes a feature to switch between perspective and orthogonal projection. It is a simple button press, but it does not show the relationship between perspective and orthogonal in terms of eye distance

```
_x_pos = _y_pos = 0;
for (i=1; i<=100; i++) {
    _x_pos = _x_pos + 3.7;
    _y_pos = _y_pos + 1.5;
    eager();
}
```

Panning

```
_x_dir is 500-view_cen[1];
_y_dir is 400-view_cen[2];
_z_dir is 0-view_cen[3];
_x_pos = _y_pos = 0;
for (i=1; i<=100; i++) {
    _x_pos = _x_pos + 3.7;
    _y_pos = _y_pos + 1.5;
    eager();
}
```

Tracking

```
_xpos=370; _y_pos=150;
_z_pos = 100; _z_dir = 0;
_y_dir is sin(swivel_angle);
_x_dir is cos(swivel_angle);
for (i=1; i<=100; i++) {
    swivel_angle = i*2*PI/100;
    eager();
}
```

Scanning

Figure 4.9: Three possible interactions with the 3D room viewer.

as the 3D room viewer model does—that orthogonal is like perspective at a large eye distance. This was not a feature that was consciously built into any of the room viewer models, but a consequence of the model being constructed with attention on observables, dependency and agency (as described in §2.2.2). The distinction between conventional software development and EM (as demonstrated in Chapter 3) is evident in comparing the 3D Dino Viewer and 3D Room Viewer—the former is a black-box with a closed set of prescribed behaviours and the latter is an artefact that can be taken apart to explore the relationships that lead to observed behaviours. As a result, the EM model can be explored in all manner of ways that have not been preconceived. One such example is animation which, as Beynon suggests in the walk-through, can be achieved by redefining the view plane parameters as in Figure 4.9. The model-builder can simulate actions that can be performed with a camera, including: panning around a scene; moving the camera whilst directing it at a fixed position; and scanning in every direction from a fixed camera location. The scripts, taken from Beynon's walk-through [Bey07b], for each of these interactions are shown in Figure 4.9.

In summary, the 3D Dino Viewer is a tool with a particular job in mind: communicating the effects of pre-specified parameters for transforming a 3D scene into a 2D view. It is easy-to-use, but not at all flexible beyond its preconceived behaviour. The

3D room viewer model is an artefact that is open to any manipulation limited only by the imagination of the learner or model-builder. Although it might require more effort on the part of the learner, it can be used to exercise preconceived uses and explore new contexts that fit the needs of the learner. The 3D room viewer model is both a teaching instrument that can be used to give demonstrations in the lecture, and also as a learning environment that the student or teacher can experiment with in his or her own time.

4.2.2 Flexibility through model combination

Another relevant feature of EM, closely related to model reuse, is model combination—reusing two or more models together in combination. In this section I am going to show how this technique can be used effectively to extend the 3D room viewer model and make it more accessible to students and teachers. I will show how several models that I have developed can be combined to create a presentation environment, and later I will combine this with the 3D room viewer. This section briefly explores some exercises in combining and extending models that have resulted in extensions to the `tkeden` tool (i.e. the Agent-Oriented Parser [Har03], the Graphical Environment Language [EMP:gelHarfield2007], the `%html` notation and environment). This eventually leads to the combination of all these models which results in the EM presentation model. Later I shall explore how the presentation model was combined with the 3D room viewer by MacDonald [EMP:room3dMacDonald1998] and the 3D room viewer walk-through by Beynon [Bey07b]. Figure 4.10 illustrates the stages of model-building activity relating to the room viewer models.

4.2.2.1 The `%html` notation and environment

The GEL notation (see Figure 4.5 on page 95) is an example of a notation that I developed as part of this thesis, using the agent-oriented parser (AOP) [EMP:agentparserHarfield2003]. Another notation I have developed using the AOP is the `%html` notation, which forms the basis of the HTML Environment [EMP:htmlenvHarfield2007] that is constructed with a combination of both the AOP and GEL. The motivation for developing the `%html` notation and environment was itself as a tool for model combination. A problematic but essential characteristic of EM tools is that they are used to construct models which

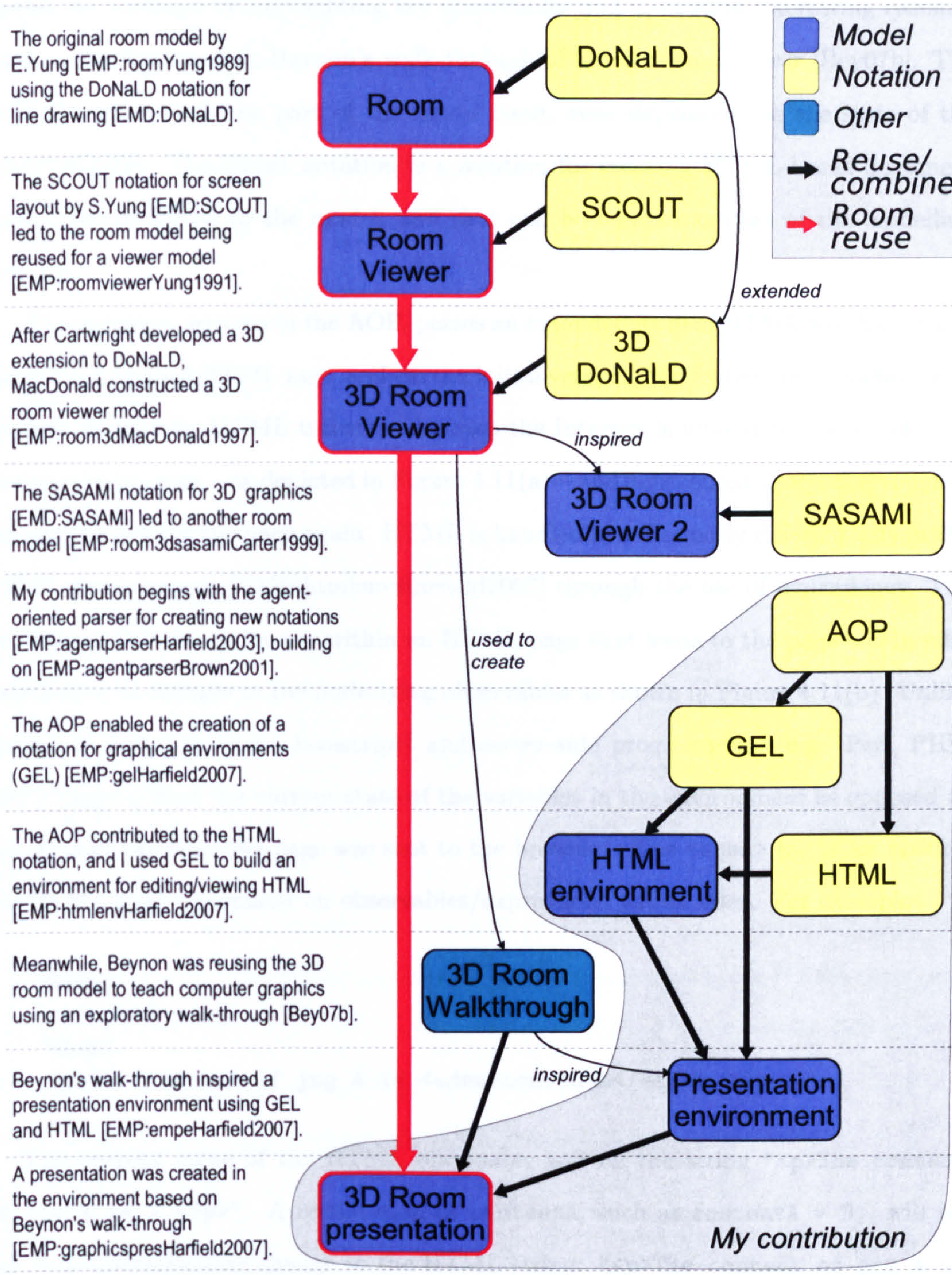


Figure 4.10: The ancestry of the 3D room graphics presention.

are personal and unique to the model-builder (as discussed in §2.2.6). This makes it difficult for other model-builders to exercise, explore and extend these models [Kin07]. To improve this situation, it is useful for a model-builder to document and explain the model, for instance by highlighting key observables and indicating interesting redefinitions as can be found in Beynon's walk-through of the 3D room viewer [Bey07b]. The documentation could be part of the model itself, even dependent on the state of the model [Kin07]. The %html notation is a solution for creating HTML-based documentation that is linked to the model, and that can be created as part of the modelling activity.

The notation, written in the AOP, parses an extended form of HTML which contains the usual standard HTML tags, and (in the initial version) two further tags: <eden> and <page>. Generally, HTML transferred across the Internet is interpreted once before it reaches the browser—as depicted in Figure 4.11(a)—and subsequent processing requires the user to request the page again. HTML is handled in a distinctly different way in the HTML Environment [EMP:htmlenvHarfield2007] through the use of dependency. The two tags enable dependencies within an HTML page that leads to the page continually responding to changes in the underlying observables as shown in Figure 4.11(b). Unlike client-side scripting (e.g. Javascript) and server-side programming (e.g. Perl, PHP, ASP), pages reflect the current state of the variables in the environment as opposed to the state at the time the page was sent to the browser[†]. The <eden> tag is for making the HTML page dependent on observables/expressions within eden. For example:

```
%eden
contentA = 2;
%html
<p>The content of jug A is <eden>contentA</eden>.</p>
```

The current value of the HTML observable will be the string "<p>The content of jug A is 2.</p>". A redefinition of contentA, such as contentA = 3;, will result in a corresponding change to the HTML string: "<p>The content of jug A is 3.</p>". Although the above syntax (for the <eden> tag) may look similar to server-side programming, it is very different in its semantics. When including a variable in PHP (see Figure 4.11(a)) using <?php echo contentA;>, it is an instruction for the

[†]Network issues are irrelevant to the discussion as the server could be on the client, or equally the HTML Environment could be run over a network using dtkedenas discussed in Chapter 5.

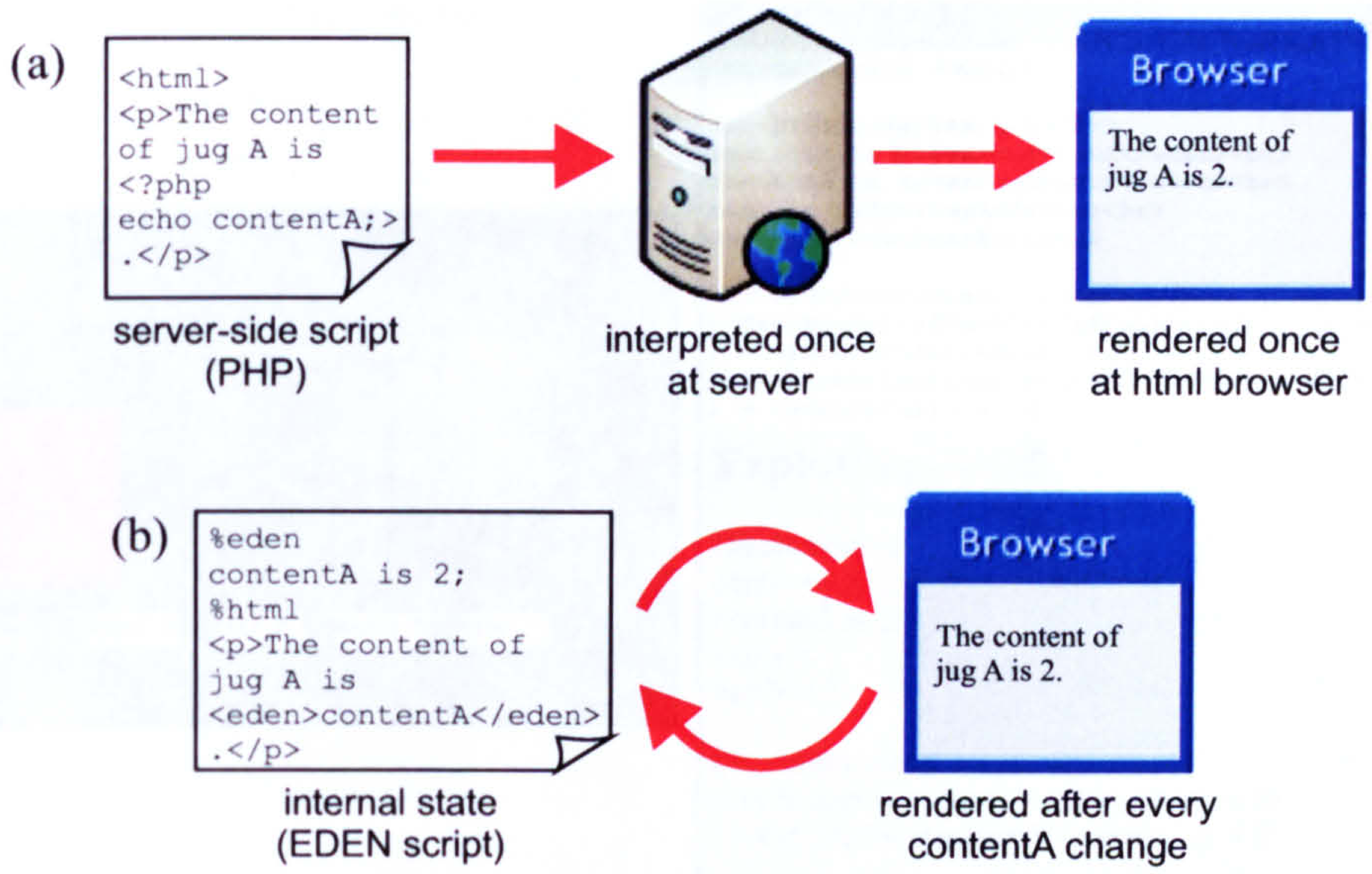


Figure 4.11: Comparing (a) traditional HTML scripting across the Internet with (b) HTML in **tkeden** using the HTML Environment.

translator (when requested) to write out the value of contentA at that particular point in the HTML code. Figure 4.11(a) shows how a script is interpreted once before it is rendered on-screen. With the EDEN script, the HTML is always consistent with the `contentA` observable; the HTML will [indivisibly] always shows the current value of `contentA` (see Figure 4.11(b)). An update to `contentA` will cause the HTML script to be updated, hence we say that the HTML script is dependent on `contentA`.

In GEL, a new component was added to parse standard HTML and render it within the component. As explained in Figure 4.5 on page 95 (and in the GEL documentation [EMD]), GEL components have some standard properties (i.e. border, background, relief, etc) and some special properties unique to the component. A HTML component has a property for setting the HTML script, and any changes to this observable cause the component to be updated with the up-to-date rendering of the HTML text. Therefore, text, formatting and graphics within a HTML component are dependent on its HTML script, and the HTML script may be further dependent on other observables within **tkeden**. Thus, if the HTML script is defined as above (`"<p>The content of jug A is <eden>contentA</eden>.</p>"`) then the HTML component will show `"The content of jug A is 2."` as paragraph text. A change to the HTML script will cause the component to re-render the HTML, and likewise, as the HTML script is dependent on `contentA`, a change or redefinition to `contentA` will cause the HTML script to update and hence update the HTML component displayed on screen.

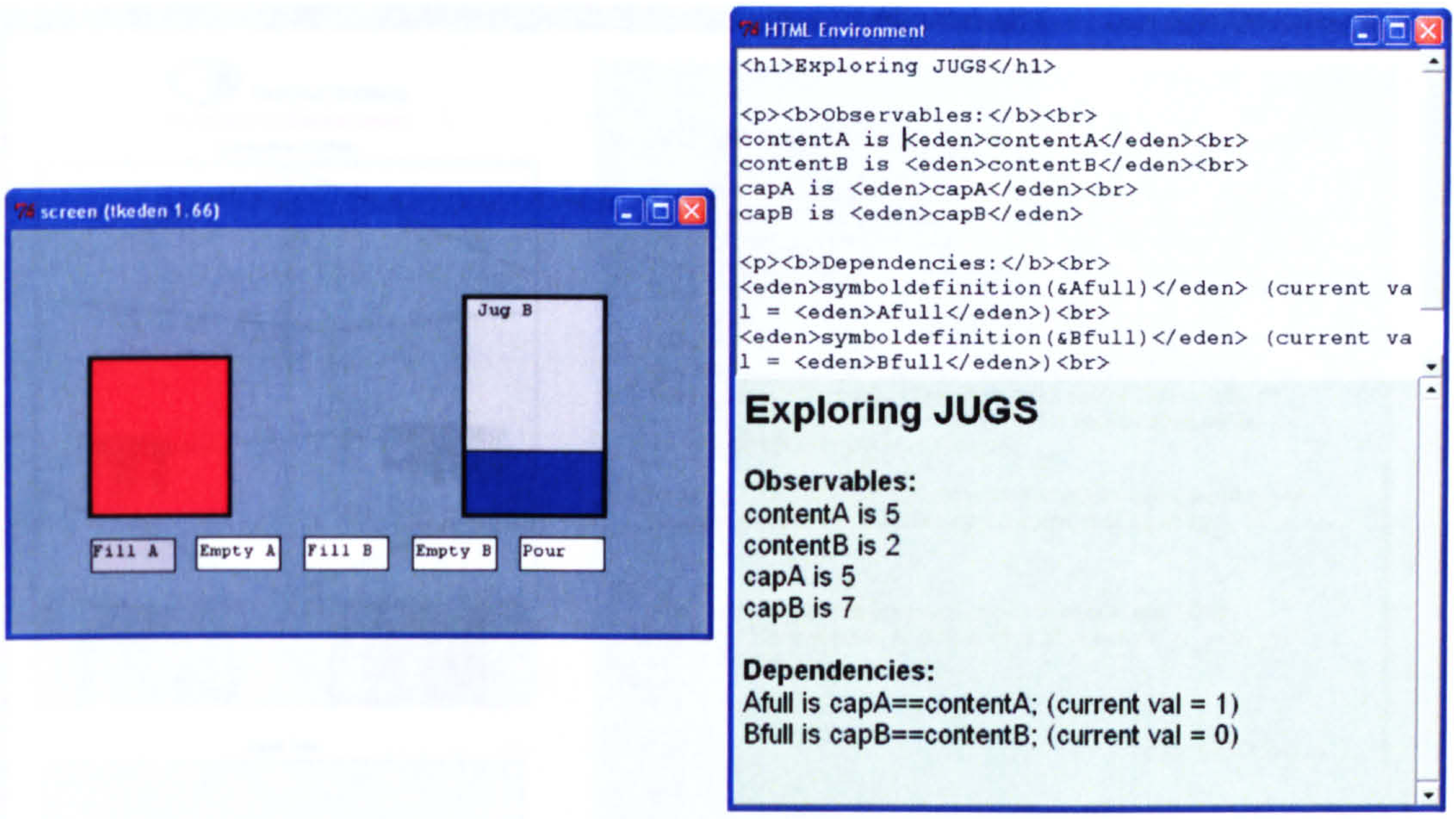


Figure 4.12: The jugs model augmented with an HTML window displaying the current state.

A combined model was created, utilising the HTML notation and the HTML component in GEL, which I have called the HTML Environment [EMP:htmlenvHarfield2007]. This model enables a model-builder to create and view HTML pages that can be used during their modelling activity. Although this model is not very useful as a stand-alone model, it is useful for model-builders to combine with their own models. It's main motivation is to provide support for drawing attention to aspects of a model so that the model is more accessible to other model-builders. Figure 4.12 shows an example of the jugs model [EMP:jugsBeynon1988] with an HTML script displaying the current state of the observables and dependencies. The HTML component, in the lower half of the HTML Environment window, is dependent on the content of the jugs in the left-hand window. King's *Introduction to dependency* presentation [EMP:jugspresentationKing2005] has similar functionality specifically for displaying the state of the jugs model [Kin07]. The benefit of the HTML Environment is that the presentation can be changed on-the-fly by redefining the HTML script.

4.2.2.2 The Presentation Environment

The Presentation Environment (EMPE) [EMP:empeHarfield2007] is an example of using the HTML Environment in a different context. The HTML Environment was created for building HTML pages to document models. The environment was modified

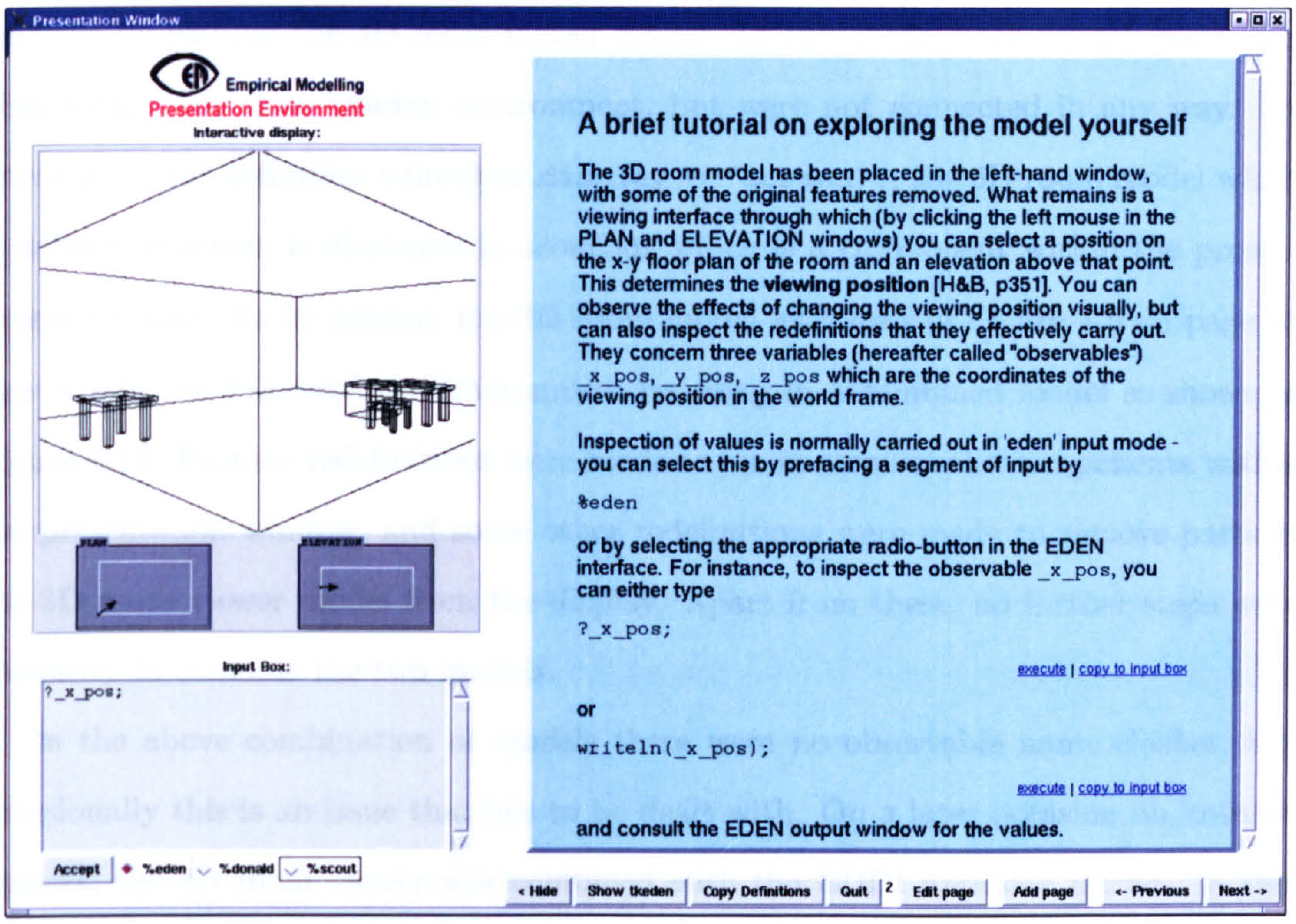


Figure 4.13: The EM Presentation Environment (EMPE).

to be a presentation tool combining slides with models. The HTML pages became slides that fill most of the window, and other components were added for displaying and interacting with models within the presentation window. Figure 4.13 shows the presentation window containing: a) slides; b) an input box; and c) a SCOUT/DoNaLD interaction box.

4.2.2.3 The 3D room presentation

Before I even thought of creating a 3D room viewer presentation, I had already performed some serious model reuse, extension, and combination: a) the AOP and EDEN to construct GEL; b) GEL and the AOP to construct an HTML Environment; c) the HTML Environment with GEL to construct the Presentation Environment. Figure 4.10 illustrates the history of these activities, and my contribution to the evolution of the room models. In the last activity, I combined the Presentation Environment with the 3D room viewer to construct a 3D room presentation [Har07a] (to be further used for the computer graphics module). The following description gives some insight into the activity of combining models.

The first action to be taken was to load the Presentation Environment into `tkeden`, followed by loading the 3D room viewer model. In this state, both models were running within the same tkeden environment, but were not connected in any way. By introducing the definition `scoutbox_display = "screen";`, the 3D room model which is defined in screen is displayed in scoutbox which is a component within the presentation window. To be precise, the 3D room model as shown in Figure 4.6 on page 98 moved into the Presentation Environment resulting in a combined model as shown in Figure 4.13. Further redefinitions were made to resize some of the components within the presentation window, and some other redefinitions were made to remove parts of the 3D room viewer model from the display. Apart from these, no further steps were necessary to combine the two models.

In the above combination of models there were no observable name clashes, but occasionally this is an issue that has to be dealt with. On a later occasion, in another context, the 3D room viewer was combined with the EDDI database model. In this case there was a clash of names as the 3D room viewer has a project function for transforming 3D to 2D and the EDDI model has a project function for selecting columns in a relation/table. To get around this problem, it was necessary to replace one of the functions under a new name and rename every usage of the function. It is also not obvious in tkeden when two observables clash from different models because redefining observables on-the-fly is a perfectly natural activity for a model-builder. This is one of the uncertain aspects of combining models, although one that 'virtual agency' (as partially implemented in current versions of tkeden) [Sun99] aims to address.

The next action was to create presentation slides to support a model-builder in exploring parts of the 3D room viewer model. This was achieved within the presentation window with the help of the HTML Environment and notation. Slides were created from the HTML given in Beynon's notes on using the 3D room viewer, some of which were discussed above, such as changing the 3D to 2D project function from perspective to orthogonal by redefining the eye distance. At this point I discovered that I could make use of the %html notation to display the state of key observables in the model. For example to display the current eye distance in a slide, the following HTML was used: `<p>The current eye distance is <eden>eye_dist</eden>.</p>`. It is worth pointing out that all of the above was achieved in a single session; at no time was there a need to close and reopen the tkeden environment, or recompile as you would a Java program.

In combining the models and using the Presentation Environment to create slides to support exercising the 3D room viewer, the resulting model is suitable not only for giving lectures but also for students to explore on their own or in groups. This environment is much more accessible than the 3D room viewer on its own as using the `%html` notation has allowed parts of the model to be uncovered for the learner. By highlighting key observables and possible interactions the model uncovers aspects that are relevant to learning about computer graphics. This 'making the model more accessible' has not been at the expense of flexibility as in the 3D Dino Viewer [Bha06]. The 3D room viewer model is there in its entirety (and it is quite possible that another model-builder might use it for a different activity, such as designing a room). However, scaffolding around the model has been added that points to parts of the model that might be worth investigating for someone learning about computer graphics. The learner must still be inclined to experiment themselves, but the learning curve has been smoothed out somewhat through the addition of help with some basic experiments.

The value of combining models is that: models can be reused in different contexts (e.g. HTML Environment used for presentations); models can be extended and given greater depth (e.g. using GEL components instead of creating them from scratch); scaffolding constructed around the model can create an environment that is accessible to other model-builders (e.g. combining the Presentation Environment with a model such as the 3D room viewer to provide some support for other model-builders to explore and develop their own models).

4.2.3 Flexibility through marrying the roles of student, teacher and developer

As discussed in Chapter 3, traditional software development techniques draw a clear distinction between development and use. In terms of educational technology, the roles of student, teacher and developer are represented by activities involving use, specification and implementation respectively (see Figure 3.10 on page 72). In EM there is no distinction between development and use because all interactions with a model are deemed to be of the same nature (i.e. redefinitions invoking state-change). Hence, distinctions between activities representative of students, teachers and developers are blurred when model-building using EM (e.g. the teacher is not concerned solely with specification). The model-builder can be seen as *marrying*, *integrating*, or *unifying* the roles of student, teacher and developer.

The way in which model-building took place in combining the Presentation Environment and the 3D room viewer demonstrated more than model reuse and combination. Using the 3D room viewer for a presentation might typically be considered as a teacher activity (e.g. in preparing a lecture), but during the model-building there were other activities taking place. As I created and used the 3D room presentation, I found that there were things that the Presentation Environment could not do. So I began to make some adjustments to the underlying models. When I delved into this further, I found that some of these limitations of the Presentation Environment were due to the HTML Environment. So I began to extend the HTML Environment also. Furthermore, I ended up changing the underlying GEL notation. What follows is a detailed account of the marrying of student, teacher and developer roles in the evolution of the 3D room graphics presentation [EMP:graphicspresHarfield2007] (as depicted in Figure 4.10 on page 105) and the implications for the flexible characteristics of learning as described in Chapter 1.

Specifically, as I began to use the 3D room presentation, I noticed that I was typing in redefinitions that were already displayed in the slides. I spent time copying what was already on the slide, occasionally making some small modifications to get different effects (e.g. changing the eye distance). I realised that I needed a way of executing the scripts and variations of the scripts that were displayed on the slide. The slides in the Presentation Environment, as it stood, lacked the potential for human interaction to be automated—it required what EDEN already has, that is agent actions (described in Chapter 2).

One of the most basic elements of agency in a webpage is the hyperlink, which usually performs an action that changes the current webpage in the browser. This action was not necessary in the Presentation Environment because the webpages had become a sequence of slides. It would be useful however, if the hyperlink could be used to trigger the execution of a script. I created an example page containing the line `<a href=writeln("hello");>click here</a>` and then I set about changing the GEL notation so that it would execute the EDEN script when the hyperlink was clicked.

```
/* redefining an existing parse rule */
%aop
<html_statement2> = <html_statement3> "<page>" <html_page>
                    : do { $v is $p1 // $p2; } now
                    | <html_statement3>;

/* defining two parse rules to deal with the <script> tag */
<html_statement3> = <html_text> "<script>" <html_script>
                    : do { $v is $p1 // $p2; } now
                    | <html_text>;
<html_script> = <html_readall> "</script>" <html_statement>
              : do { $v is "format("//$p1//")//"//$p2; } now
              | <html_error2>;

/* defining a new function for displaying the script */
%eden
func format {
   para script;
   return "<pre>"//script//"</pre>"
      //"<p align=right><font size=-3>"
      //"<a href=execute(\""//script//"\");>execute</a> | "
      //"<a href=inputbox_text=\""//script//"\";>"
      //"copy to input box</a>"
      //"</font></p>";
}
```

Figure 4.14: Adding the 'script' tag and complementary HTML formatting.

This required a change to the underlying **Tcl/Tk** scripts that deals with the HTML component, but this was achieved on-the-fly without restarting the model or the environment.

In the course of making this small change, I had moved from the role of student who was exercising the presentation, to the role of teacher who was creating the presentation slides, to the role of developer who was modifying the functionality in the Presentation Environment. All of these roles were played in the same environment, there was no switching of modes or views that is a common feature of many learning environments. As a learner, I had *married*, *integrated* or *unified* the roles of student, teacher and developer, as shown in Figure 3.11 on page 73.

After the above changes, it was possible to place scripts within a hyperlink on a slide that could be triggered by the model-builder. In terms of ease of use, this was an improvement, but it encouraged the 'just looking at the artefact' type of interaction that was disapproved of earlier. I was not satisfied that this exposed the model to manipulation in a flexible manner.

The next extension I made to the environment involved adding a new tag to the **%html** notation that dealt with scripts. This new tag would contain EDEN or other

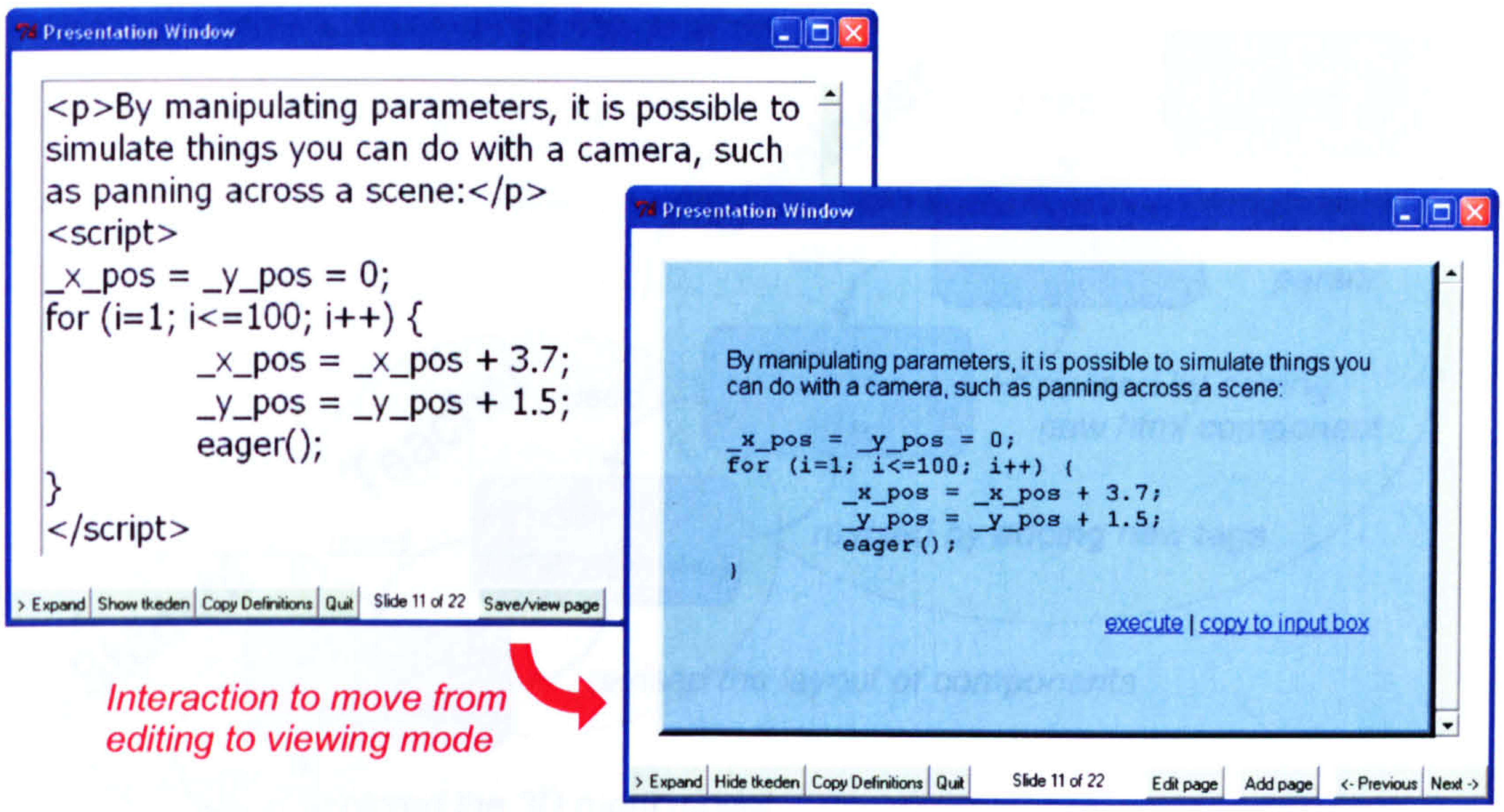


Figure 4.15: Using the 'script' tag immediately in the Presentation Environment.

scripts, and once rendered in a GEL HTML component, allow the model-builder to execute them. It would also open up the possibility of making changes to the scripts before execution by copying the to the input box. This would be a further extension of the technique I had moments ago constructed. Figure 4.14 contains the EDEN script showing how I changed the `%html` notation to parse a `<script>` tag that transformed the contained script into a formatted script with some hyperlinks for actions on the script. This required some significant changes to the `%html` notation, adding some new parse rules and changing some existing ones (see Figure 4.14). The first line of Figure 4.14 changes the existing rule to inform the AOP of the new statement, the second and third lines describe what to do with script tags, and the EDEN function converts a script into some nicely formatted HTML. After these changes to the `%html` notation, the `<script>` tag was immediately available for use in the presentation slides. Figure 4.15 shows a slide being edited to make use of the `<script>` tag and the slide being viewed with the formatted script. This second extension again shows how a model-builder, from the role of a teacher creating a presentation, can take on the role of a developer to modify the environment in which the presentation is created. The modifications are more than superficial changes such as changing the layout or design of the slides—they are fundamental changes to the notations and state of the environment. Figure 4.16 highlights the models that were involved in the interaction and shows that some interactions could be typified as teacher activities, whereas others—such as the

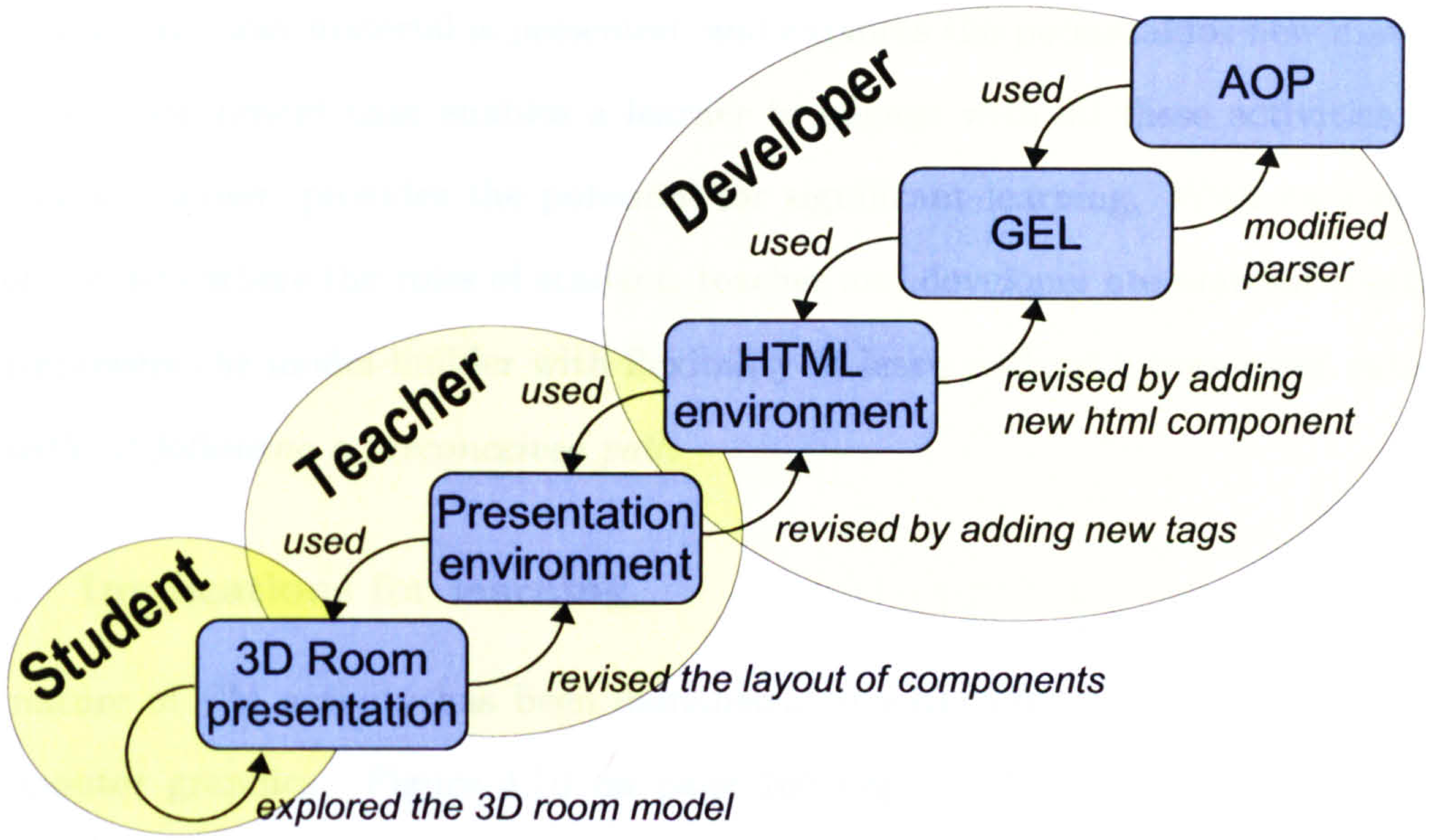


Figure 4.16: Interactions demonstrating the marrying of the roles of student, teacher and developer.

addition of the `<script>` tag—could be seen as developer activities.

The account of EM in this section shows that a teacher creating a presentation is a student exercising the presentation for themselves, and the student or teacher can also be a developer modifying the Presentation Environment as depicted in Figure 4.16. To some extent the words 'student', 'teacher' and 'developer' are inappropriate to refer to each of the roles because their activities are not constrained in the same way as they are with ET as discussed in §3.3. Figure 4.16 illustrates how a model-builder is likely to play the roles of 'student', 'teacher' and 'developer' at different times when exercising, manipulating and constructing the model. A model-builder's experience with the model is likely to determine the extent to which it is exercised, manipulated and constructed. As a model-builder becomes more experienced, deeper changes will no doubt be made to the underlying observables, dependencies and agency that form the model, from the surface level of key observables in the model to a deeper level of observables uncovered from significant exploration. In this way a model-builder moves from a role of user to a role of developer within the same session and environment.

It has been shown how EM (using the `tkeden` tool) enables the model-builder to play the role not only of the student and the teacher, but also of the developer. In some respects, a learner with the characteristics set out in Chapter 1 is constantly involved in activities in each these roles; a learner studies the given material, questions and changes the way material is presented, and explores the potential for new material. Thus, an environment that enables a learner to engage with all these activities, in a continuous manner, provides the potential for significant learning. With its focus on model-builders where the roles of student, teacher and developer are married together, EM empowers the model-builder with flexibility to learn *without a prescribed outcome* and *without following a preconceived path.*

4.2.4 Implications for learning

The nature of EM activity has been demonstrated with reference to a specific topic in computer graphics. Figure 4.10 on page 105 depicts the extent to which model reuse and combination can be useful in EM. With reference to learning, model reuse and combination can lead to *flexibility* in learning on two levels. On the level of interactions, the nature of EM allows the model-builder to be concerned with observing and exploring states, not developing and using prescribed behaviours, as differentiated in §3.1. Figure 4.16 demonstrates that a model-builder can flexibly take on the role of a teacher at one moment and of a developer at the next. In EM, exercising a model need not have a *prescribed outcome* for the learner, and exploring a model need not follow a *preconceived path.* Similarly on the level of models, the potential for model reuse and combination ensures that the possible outcomes and paths that models can take are endless. Considering the evolution of the room model in Figure 4.10, it is impossible that Yung could have envisaged the potential for teaching computer graphics, or the path that his model would follow in eventually contributing to the teaching of computer graphics. The nature of EM as explored in this section (i.e. prior to prescription and ritualisation) match the flexible strand of characteristics of learning described in Chapter 1, that is learning *occurs without a prescribed outcome* and *need not follow a preconceived path.*

4.3 Experimental learning in artificial intelligence

The final section of this investigation into particular areas of computer science education examines the experimental characteristics supported by EM in the context of learning about a specific topic in artificial intelligence (AI). As explained in §1.2.1–

§1.2.3, the three characteristics that form the experimental strand of eight significant characteristics of learning are:

- learning occurs when constructing artefacts in the world;
- learning involves an active construction of understanding;
- learning results from realising the unknown.

While the previous two sections on databases and graphics demonstrate EM's suitability for teaching some standard components of a computer science course, this section examines the potential for learning in a more experimental fashion, as might be expected in research or in the primitive stages of developing teaching material. The subject of AI is chosen because a number of student projects have been undertaken relating to AI (e.g. the Wumpus model [EMP:wumpusCole2005] and the Ant Navigation model [EMP:antnavigationKeer2005] to be discussed further in Chapter 6) and my research has previously applied EM experimentally to explore AI and social behaviour in studying human-robot interaction (HRI)†. Therefore, this section relates to computer science education primarily from a research perspective. However, this is not to suggest that the experimental characteristics of EM are only applicable to research—the case studies in databases and computer graphics demonstrate experimental qualities not primarily drawn on in the discussion.

4.3.1 Social behaviour, AI and human-robot interaction

Human-robot interaction (HRI) is the study of interaction between people and robots, and is concerned with developing robots that react in ways that are acceptable to people. Robots are already being used in areas such as search and rescue, bomb disposal and space exploration, and in the future it is likely that robots will be found in everyday environments. Dautenhahn, a prominent researcher in the growing field of HRI, discusses the need to consider how robots can coordinate their actions with people in a suitable manner [Dau07]. In order to develop such socially-aware robots, designers must attend to details of social behaviour—as studied by social scientists—and develop

†The application of EM to AI drawn on in this section is a development of ideas from a joint paper presented at the Artificial Intelligence and Simulation of Behaviour (AISB) Annual Convention, University of Herfordshire, in April 2005 [BHC05].

AI—as studied by computer scientists—that imitates *some* aspects of social behaviour [Dau07].

The difficulty of dealing effectively with issues relating to social intelligence in the design of robots is widely recognised. In discussing this challenge, Fong et al [FND02] identify two approaches to the design of socially intelligent agents, the 'biological' and the 'functional'. The biological approach aims to draw on understanding of animals and their behaviour to design robots which exhibit similar properties to their biological counterparts. The functional approach only takes the functionality of such robots into account and is not concerned with the mechanisms by which this is achieved. Traditional AI generally takes a functional approach. The biological approach is favoured by those interested in the social sciences and biology.

Whatever the orientation of the robot design, there are major technical and conceptual issues to be addressed in developing robots that are socially responsive. It is implausible that such issues can be resolved by implementing behaviours that are comprehensively pre-specified by abstract analysis of the operating context as would be expected with a conventional program.

There are a number of relationships that may have an impact upon social interaction. Three significant type of relations, as described in [BHC05], are:

- *Spatial relations*—An agent's physical location and the surrounding space are likely to affect the behaviour of the agent. Actions in small confined spaces are usually different from those in large open spaces.

- *Temporal relations*—Time plays a significant role in human behaviour. When time is at a premium humans are likely to perform tasks differently from when they have plenty of time.

- *Status relations*—The status of human agents affects their interaction and expectations. Interaction with those with whom we are familiar differs from interaction with strangers. Interaction within the working environment, families and cultural contexts is likewise differentiated according to the status of the agents with whom we are interacting.

Human beings take account of these relations by gradually learning from experience, whereas a conventional programming approach to robots views these relations as being predefined up-front. A key issue in HRI is to what extent should robots be able to learn from experience.

4.3.2 EM for learning about HRI

In the joint paper with Beynon & Chang [BHC05], it is argued that EM is a suitable approach for learning about these issues that may then inform actual robot development. The evidence is based on learning about HRI through the construction of personal models developed by myself and other model-builders. Three models are used as examples of attending to spatial, temporal and status relations.

4.3.2.1 Spatial relations

The first model of interest relates to the study of crowd behaviour and was developed by Martin as part of a final year undergraduate project. The model contains a graphical representation (2D line drawing using the DoNaLD notation [EMD]) of two people walking towards each other in a corridor—it will be referred to in this thesis as the corridor model†. Martin used the model for experimentation to learn about how people avoid each other in a corridor, and therefore, it satisfies the experimental characteristic that EM can be used, in the spirit of constructionism, to construct artefacts for learning as discussed in Chapter 2.

The artefact takes account of many spatial elements: heading, direction of focus, field of attention, and personal space [Mar04:p35]. Martin experimented with different mechanisms for collision avoidance by varying the emphasis of these elements. This resulted in four types of avoidance strategy [Mar04:p37] which were compared and examined as possible strategies that could be implemented in robots. As strategies developed, observations led to changes in the model. For example, when experimenting with avoiding by focussing on a point in the distance, the agent required an observable for the re-target direction and distance [Mar04:p41]. Martin also developed an understanding of what sort of strategies for avoidance are realistically used by humans, and therefore he experimentally formulated possible theories about a specific area of human behaviour. Martin describes the benefits for learning of EM artefacts is that they are "experimental testbeds for formulating theories and possible implementations"

†Martin's project contains three variations of corridor model [Mar04:p30], but only the first and simplest will be covered here (the third and most complicated contains 25 agents).

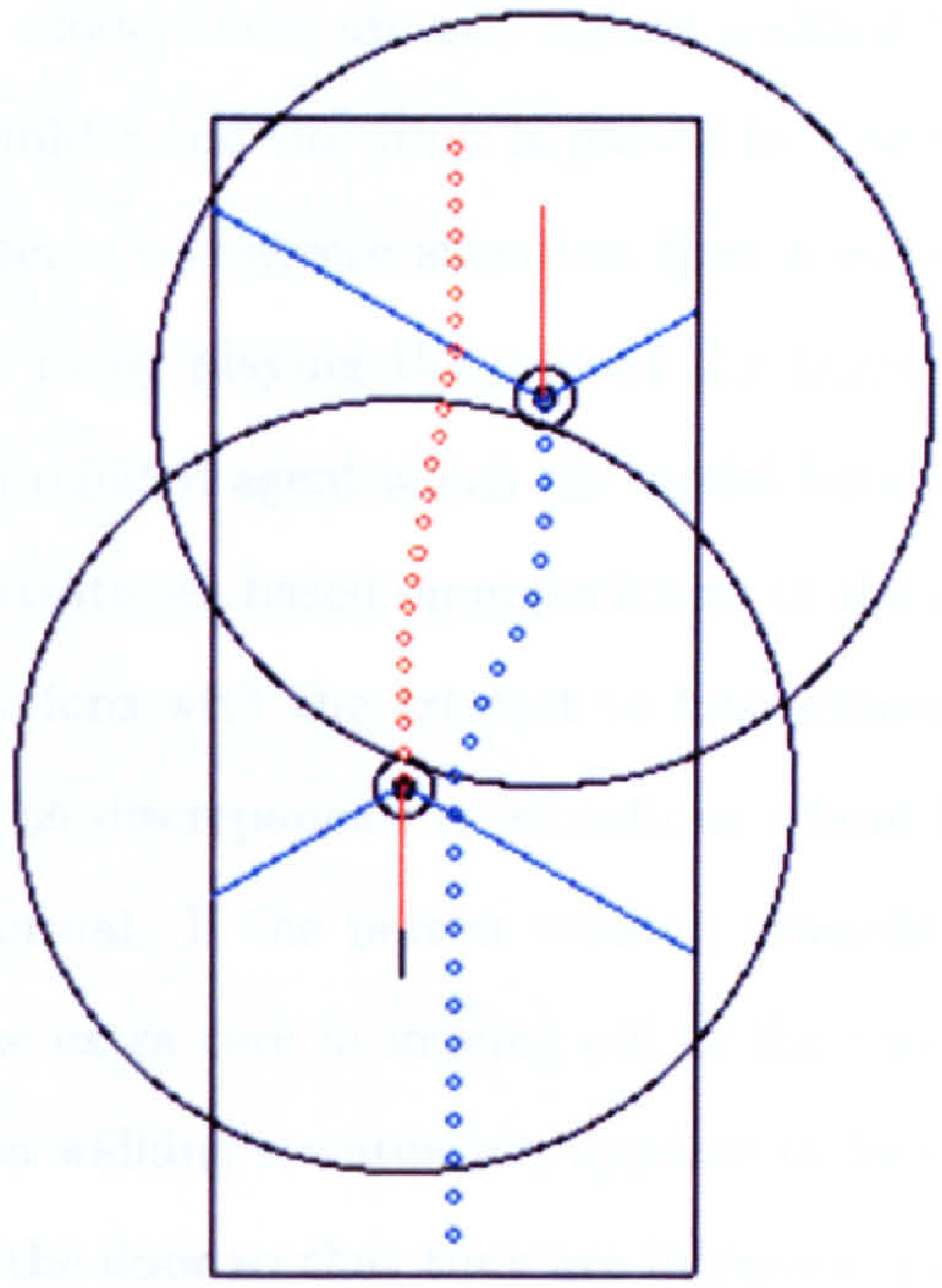


Figure 4.17: The corridor model by Martin [Mar04].

[Mar04:p78]. This reflects Jonassen's idea of learners as theory builders where building theories contributes at a fundamental level to conceptual change [Jon06].

Martin says that model-building "stimulates ideas and discussion" [Mar04:p80]. Many of the issues that his model-building brought up related to spatial relations. Examples given include the introduction of a stairway into the corridor model that could have ramifications for the agent's field of vision, a doorway that might force agents to stop and wait or even hold the door for another, or how an agent walking side-by-side with a stranger might change their pace to avoid uncomfortable synchronisation.

Learning occurs when constructing artefacts in the world because construction leads to experiences on all levels of the EFL (see §2.2.8). By constructing models, learners have to think about the formal definition of a phenomena, and by using artefacts there is a practical experience of the phenomena. In everyday life we have the concrete experiences but not the abstract ideas about how we avoid people in a corridor (i.e. we don't think about it, we just do it). At the other extreme there is modelling in mathematics which may force us to think in great detail about the formal definition of a phenomena but often without being able to test the abstract constructions in concrete situations. The benefit of model-building, specifically with reference to observation, dependency and agency, is that we can construct models and experience them. For example, in the corridor model there are two agents walking towards each other, one is played by the model-builder and the other is played by a semi-automated agent. On the one hand this represents a concrete situation that a learner studying HRI could relate to and participate in by playing the part of the human walker. On the other hand there is the semi-automated agent which the model-builder can use to experiment with possible avoidance strategies based on experiences in the model and in the world.

By comparing interactions with the artefact to interactions with the referent it is common that there will be discrepancies or situations where behaviour in a corridor may deviate from the normal. If the person walking towards you is holding a white stick then you might take extra care in moving out of the way. Or if moving through a doorway and the person walking towards you appears to be elderly then a respectful action would be to hold the door so that they are inconvenienced as little as possible. The fluid nature of EM activity means that the personal, particular or subtle needs can incorporated in the stream of model-building—this is a significant characteristic of EM that is beneficial in learning.

4.3.2.2 Temporal relations

The emergency egress model was constructed by Howard Perrin as part of a submission to WEB-EM-2 [WEBEM2:Emergency Egress Simulation]. It was inspired by Martin's corridor model and builds on Martin's initial contribution to understanding crowd behaviour. In particular, the model considers the situation of a large room (e.g. seminar room) or building (e.g. Department of Computer Science) containing many agents (e.g. students) and the event of a fire alarm.

The model includes many spatial aspects similar to the corridor model: people in the model have a physical or personal space that cannot be penetrated by other people, people avoid walking into walls and cannot see around or through walls, people cannot walk over desks and other obstacles, although they can see around them. Spatial relations might change depending on the status or time (e.g. during a fire alarm it might be that people are quite comfortable moving through a doorway more than one at a time).

The main temporal relation in this model involves the fire alarm. When the time comes for an emergency exit, the agents in the model all change their focus to exiting

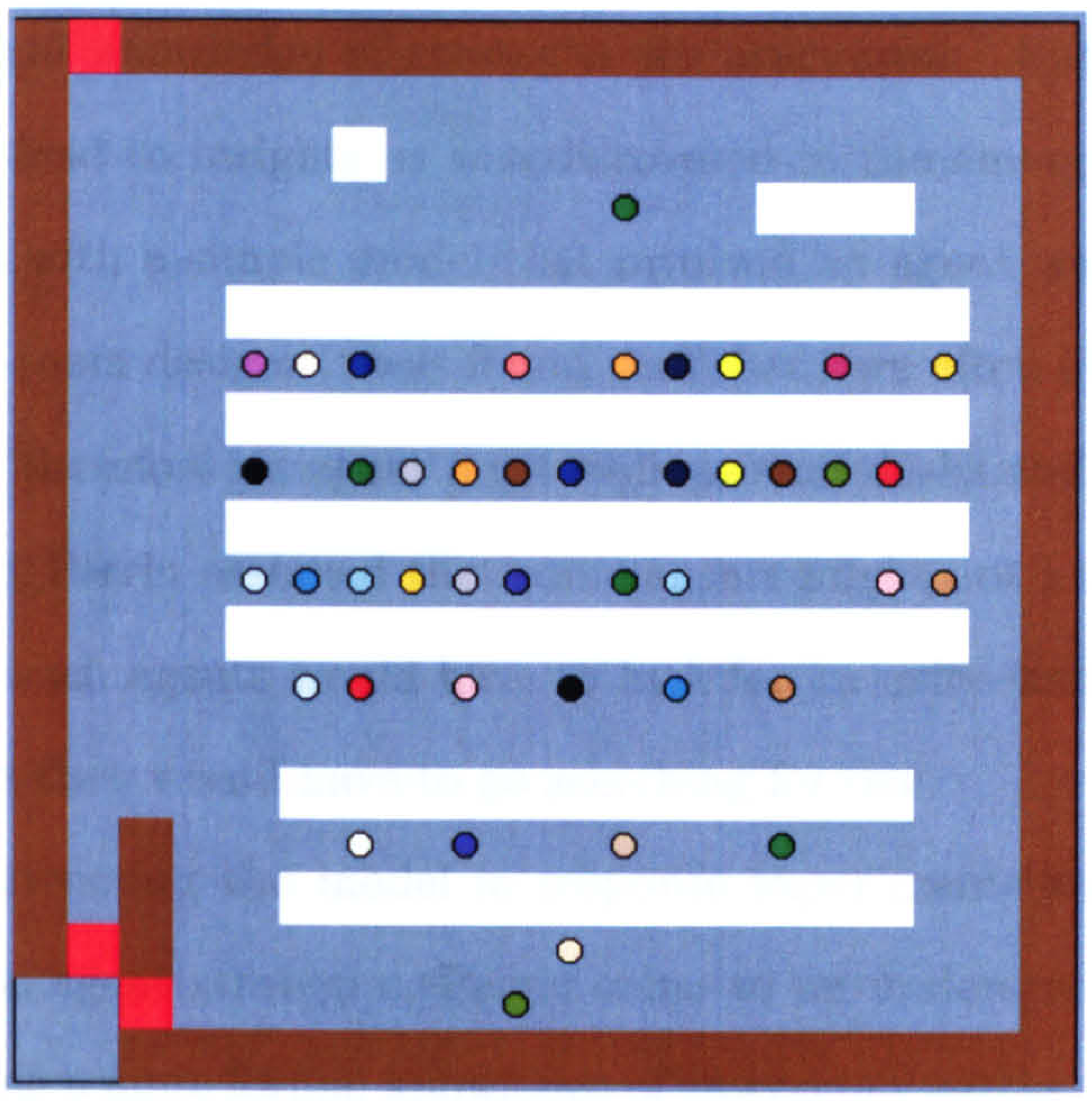


Figure 4.18: The emergency egress model by Perrin [WEBEM2:Emergency Egress Simulation].

from the room. This might involve searching for an emergency exit sign or door, observing other agents head towards an exit point, moving to an exit, or queuing to get to an exit.

The emergency egress model can be used to learn about the design of rooms in terms of where to place exits and furniture, or to test out emergency situations to find out the time required to vacate a building. Relevant to learning in this section is the idea that such models can be used to learn about human behaviour so that it is possible to build robots that exhibit similar characteristics to people.

One of the key elements to constructing models that mimic human behaviour is to construct the model from the viewpoint of the person or agent inside the model. In the emergency exit situation, the agent is unlikely to have a complete knowledge of the layout of the building including all the fire exits or a precise knowledge of the closest fire exit from any given point in the building. A computer model may be useful for calculating these things, but they are not necessary for successful evacuation. It is possible that a person in a familiar building might think through the best way to escape from the building in an emergency, but sometimes all that is needed is a simple strategy like 'follow your neighbour'. By the model-building putting themselves in the position of someone trying to exit a room in an emergency is useful because not only might it lead to more realistic agents, but also it connects practical knowledge of emergency situations with formal knowledge of strategies for evacuation. Such an approach to model-building can lead to insights as was discovered in the emergency egress model where Perrin began with a simple model that involved an agent walking towards any exit. With different room designs it was found that there are often obstacles to moving directly to the exit, therefore an agent must walk around desks and chairs in order to get to the exit. Next Perrin assumed that some agents might not know where the exit is in the room, and such agents would have to look for an exit—and if they could not observe an exit then they would have to go searching for one.

By actively constructing the model in response experiments with different room designs and different agent strategies, Perrin came to an understanding of the issues involved in developing a more formal knowledge of emergency egress—particularly from the viewpoint of an agent. Such knowledge is useful when considering HRI, but not necessarily in terms of implementing robots that are able to successfully vacate a building in an emergency. More important to HRI is the temporal relations that may impact on social interaction. In an emergency it is important that the robot does not hinder the evacuation process, therefore it might be useful for the robot to have a model of human behaviour in order for it to avoid obstructing people exiting the building—it is probably not desirable for the robot to join in the rush for the exit. As time progresses, it might be suitable for the robot to exit the building, or for it to take on other responsibilities such as checking the building for stranded people. Perrin's model can take account of such situations as he explains, by for example, making an agent move very slowly to simulate a person with a broken leg. A possible next experiment for a model-builder interested in HRI would be to see if one of the agents could act as an observer looking for peculiarities such as slow moving people—such agents may be useful in an emergency situation†.

In order to explore such relationships in HRI, the EM approach to model-building engages the learner in active construction. EM's approach of identifying patterns of observables, dependencies and agency enables a close relationship between experiences in the world and in the model. Perrin's model is connected to issues in crowd behaviour that Martin was investigating in the corridor model, but Perrin started from scratch in the construction of his model. In this way, experimentation was not constrained to ex-

†A detailed description of HRI issues in urban search and rescue is given by Murphy [Mur04].

ercising an existing model—like experimentation in microworlds according to Jonassen [Jon06]— but actually involved constructing a model to develop some understanding. Therefore the emergency egress model shows support for the characteristic 'learning involves an active construction of understanding' described in §1.2.2. A key factor in EM's support for active construction is the well-conceived foundation of modelling with dependency as described in §2.2.2.

Further refinements relating to temporal aspects of emergencies might involve a 'panicky' aspect, whereby the longer a person has been waiting to escape the room the more they become inpatient and the less likely they are to act in an orderly fashion. In such cases observing social distances will be less of a priority than getting through the fire exit as quickly as possible. It is through experimentation with observables, dependencies and agency that the model-builder will be able to recognise and address subtle features of the problems in HRI.

4.3.2.3 Status relations

The first objective in applying EM to HRI would be to better understand how human capabilities and behaviours and robot capabilities can be most effectively elaborated and integrated. As illustrated in the corridor and emergency egress models, EM can help explore the factors that are significant in determining human behaviour in relation to such tasks as collision avoidance. It can also enable the construction of prototype behaviours expressed in observables, dependencies and agency that serve as useful models for devising and analysing robot behaviour. A more ambitious goal involves demonstrating that EM can used in programming robots, especially where low-level interface issues such as image recognition are a concern. Such questions are not elaborated in this section where the focus is on learning about HRI—for a detailed discussion of the application of EM to HRI refer to Beynon et al [BHC05].

The third model to be considered explores the situation of a domestic robot interacting with people in a house. It could serve as a useful model for learning about the issues in HRI, and particularly for experimenting with various configurations of observables, dependencies and agency for robots. The model uses elements from a model developed by Sunny Chang in her final year undergraduate project in modelling an intelligent house environment [Cha04]. The house robot model that I created in

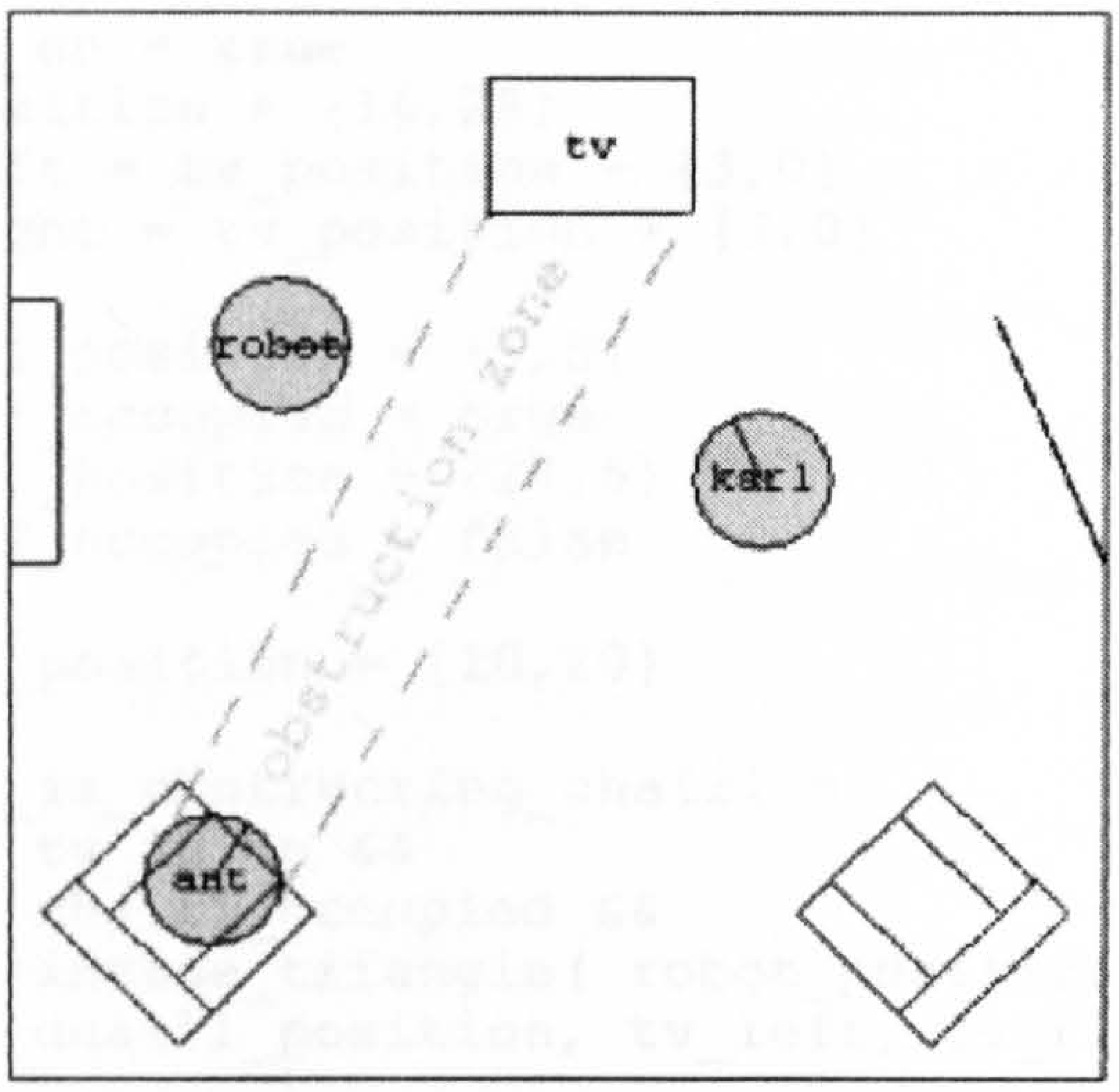


Figure 4.19: The living room in the house robot model.

tkeden was used to experiment with various familiar situations in the house where a robot might feature or be considered useful in the future†.

Dependency plays a key role in all forms of HRI and experimentation with dependencies can inform possible robot behaviour. By way of illustration, consider the dependency involved in a living room scenario where there are people watching television, as depicted in the screenshot shown in Figure 4.19 taken from the house robot model. Clearly it would be undesirable for a house robot to obstruct someone while they are watching television. In a similar way to the use of dependency to detect collisions in the corridor model, a dependency (or set of dependencies) can represent a potential obstruction by the robot. Figure 4.19 designates a potential obstruction zone for the robot. Using dependency, these areas can be defined in terms of the position of other agents and the television, so that if a person moves then the robots awareness of an obstruction is updated. Another issue that should be considered is if there is actually an obstruction. The *status* of the television also determines whether there is an obstruction—if the television is switched off then the robot can be fairly sure that it is not being watched. The obstruction is then dependent on: the robot being inside the area between the person and the extremities of the television, and the television being switched on. Thus, there are *spatial* and *status* relations to be consider in this situation, as well as a *temporal* aspect in that the status and position of the agents may change over time. The way in which these dependencies can be directly manip-

†It has been suggested that most people like the idea of robots taking over the menial jobs around the house—whether it is possible, and when it is likely to happen, is yet to be seen.

```
tv_is_on = true
tv_position = {16,28}
tv_left = tv_position - {3,0}
tv_right = tv_position + {3,0}

chair1_position = {5,5}
chair1_occupied = true
chair2_position = {27,5}
chair2_occupied = false

robot_position = {10,20}

robot_is_obstructing_chair1 =
      tv_is_on &&
      chair1_occupied &&
      inside_triangle( robot_position,
      chair1_position, tv_left, tv_right )

robot_is_obstructing =
      robot_is_obstructing_chair1 ||
      robot_is_obstructing_chair2
```

Figure 4.20: An extract from the house robot model showing that an obstruction is dependent on the position and status of other agents.

ulated in tkeden is illustrated in a small extract of script in Figure 4.20, selected for this example from the house robot model shown in Figure 4.19. Further dependencies could be introduced at any time, for example some people might be particularly sensitive about how much space is occupied around or behind the television—therefore the dependencies relating to robot_is_obstructing could be modified so that the robot is deemed to be obstructing if it is behind the television.

The house robot model can support learning about HRI due to the focus on experimenting with a situation in which we are generally familiar. By taking a familiar scenario such as the living room of a house and introducing a new agent (e.g. a robot agent) into the picture we are 'building up' the model incrementally, situating the unfamiliar new element in a familiar environment, from the perspective of an external observer and also from the perspective of the new agent. Model-building is not a one-off exercise, as is typical in programming to create a complete program with specific outputs; it is focussed on building up or evolving a model. The activity of model-building establishes—through experimentation—an intimate relation between the artefact itself and the mental model of the model-builder. Thus, the activity is a support for learning. Experimentation is the key to the evolution of this relationship, and very much involves considering and exercising familiar aspects of the artefact in order to discover what is not familiar in our mental models.

Consider a possible evolution of the house robot model, in a larger living room scenario with more agents, where the robot is given the task of moving from one side of the room to the other—in order to reach the cocktail cabinet to replenish it's drinks tray for example. The robot now has to combine the avoidance of obstruction zones with the avoidance of people moving around the room whilst completing a task. One way to build up the model would be to combine it with the corridor model, and experiment with the effects of this conjunction. That modelling social interaction should be evolutionary and open-ended is completely plausible. As we do not fully understand the nature of social conventions [Gil95]—even our own—it is unlikely that we will ever want to finalise (or completely formalise) a behavioural model.

In another experimentation session I considered how a robot might learn or adapt its dependencies associated with how much space a person is comfortable with before an obstruction or intrusion is felt. When two people are talking together they assume a comfortable distance with which to communicate, when standing in a queue people have a natural feeling for how close they should stand, when walking past someone it is not considered natural to brush shoulders or to take particularly evasive action to avoid the person. In *The Hidden Dimension*, Edward Hall devotes an entire book to these topics and he describes in detail the subjective personal spaces that people keep with one another [Hal66]. From Hall's descriptions, and from personal inspection, it is obvious that such personal spaces for an individual develop over time from experiences, and cannot be formalised even within a particular culture. My concerns were related to how we might become aware of personal spaces and learn to respect them, from the position of an agent like the house robot. My strategy involved giving the robot agent a rough distance to keep between itself and other agents and then modifying this when the robot noticed that other agents moved away when close to them. Experimenting with this model led me to realise that there are many other issues that were initially unknown to me. For example, if a person walks towards the robot there might be expectation that the robot would move out of the way, and not necessarily adapt to a new closer social distance.

By exploring these HRI issues using the house robot model, EM's potential for supporting experimentation in learning has been demonstrated. Experimentation is an incremental activity which involves building models of what is familiar and using them to highlight the unfamiliar—it is a particularly fluid activity to which EM is well-suited. As models are built up, the model-builder becomes aware of relevant 'residue' issues and can begin to experiment with patterns of ODA to develop a deeper understanding of the issues. In this way, EM offers support for *learning resulting from realising the unknown* as characterised in §1.2.3.

4.3.3 Implications for learning

The EM approach offers potential for learning about issues in HRI by building on previous work [BHC05]. The above discussion highlights the importance of EM's support for the experimental characteristics of learning as introduced in Chapter 1. The introduction to EM in §2.1 highlights the nature of EM construals that in some sense 'embody knowledge' [BHC05]. Construals are particularly connected to personal interests, situations and experience, and this makes them ideal for experimentation that involves a necessary human element [BR07].

The construction of construals using EM tools is an active process of incrementally building up and continually refining the artefact in response to experimentation. As highlighted by the house robot model, experimentation and construction are open-ended—it is the model-builder who actively chooses in which direction to explore. Artefacts start simple: observables and dependencies are created from scratch, or an existing model forms the basis of further model-building. In the context of HRI, as spatial, temporal or other relations become apparent, observables and dependencies are elaborated that embody more complex ideas. Where interactions and observations with the model do not reliably correspond to interactions and observations with the referent, the model-builder engages in an experimental sense-making activity in order to develop a deeper understanding of the model and the referent. Experimentation is the key activity that enables a correspondence to develop between the model and the referent as depicted in Figure 2.5 on page 39. Experimentation promotes the development of understanding and plays a significant role in learning about specific topics in HRI, graphics and databases as demonstrated in this chapter.

Chapter 5

EM for education in a broader context

The previous chapter illustrated EM as an educational technology from the perspective of teaching and learning about computer science. However this thesis argues that EM has potential beyond computer science education. The aim of this chapter is to demonstrate that EM is well-aligned with education in other subjects, adult education or lifelong learning, and collaborative learning.

5.1 Learning in other subjects

5.1.1 A selection of EM models and activity

EM has been used to build models and learn about a wide range of subjects, as can be seen from the EM project archive [EMP]. The breadth of the modelling activity spans mathematics, engineering, science, humanities, arts and business.

Mathematics education is a well established field and it is no surprise that many postgraduate and undergraduate projects have explored this area in relation to EM. Roe has investigated the use of EM for mathematics education [Roe99] [Roe03], starting with model-building in the school curriculum with coordinate geometry [EMP:cogRoe1999], and later in terms of exploratory modelling to understand monotone boolean functions [EMP:fdl4Beynon2002]. Roe also discusses the merits of open-ended environments—in particular using Imagine Logo to develop models of shot put and discus throwing—for putting the learning back into e-learning for mathematics [RPJ05]. Other topics that have been explored by other authors using EM tools can be found for example in

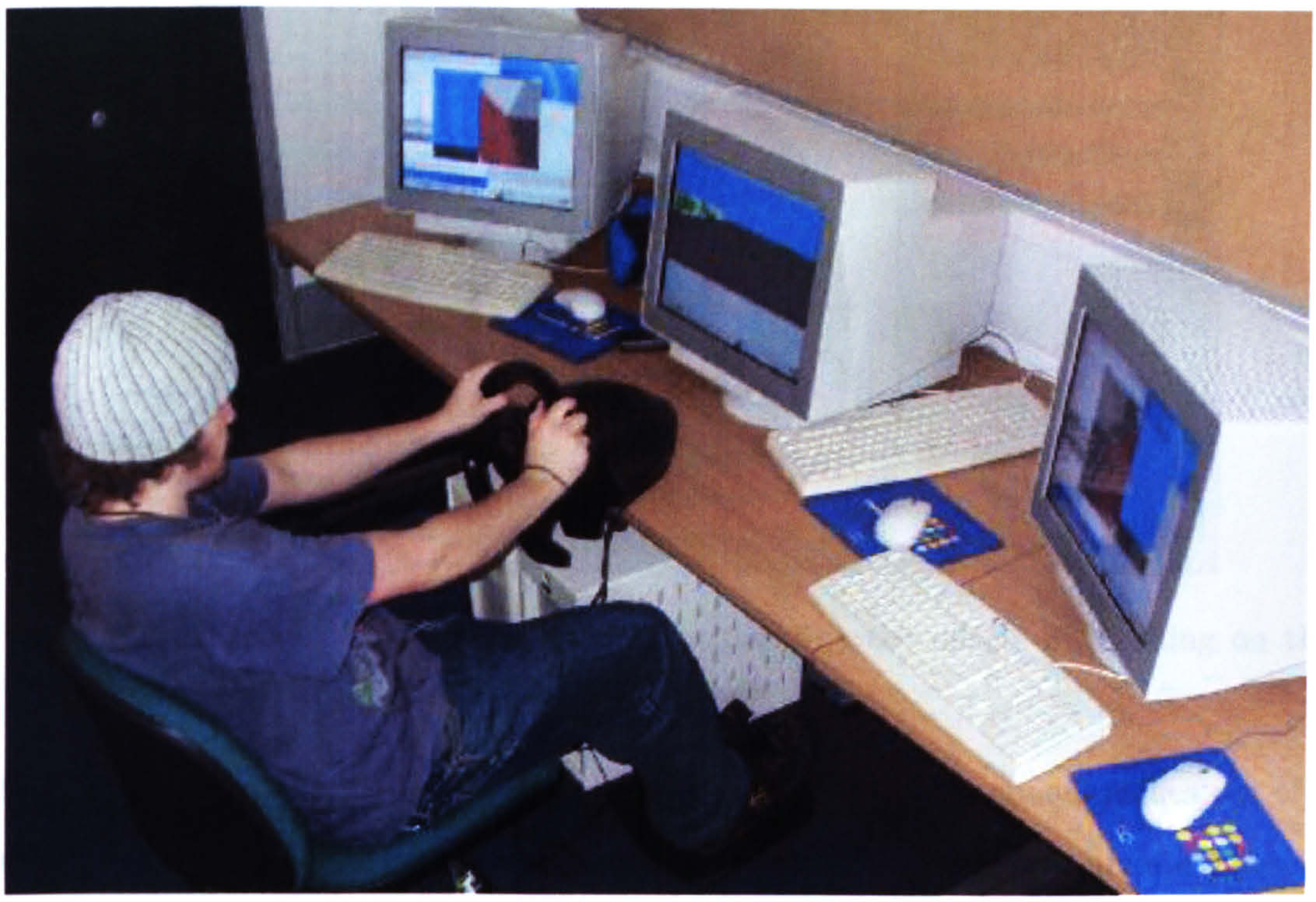


Figure 5.1: Car parking simulator [EMP:carparkingMcHale2003].

a complex number model [EMP:complexGardner1999] and the fractional relationship and equivalence helper [EMP:fractionsCronick2003]. The latter model shows external similarities to Visual Fractions from Logotron† in respect of the use of dependency to experiment with fractional equivalence. However, behind the interface, dependency in Visual Fractions is made possible by some significant procedural programming using Imagine Logo, whereas the fractions model by Cronick is relatively simple to comprehend as a set of observables and dependencies. Another relevant EM environment was created during an undergraduate final year project by Guillou which examined the use of adventure games for mathematics education [Gui03].

Engineering has been an active area of research for EM [FB01] [BNY94], although not primarily in terms of education. One of the most relevant projects to education is the car parking simulator which was developed by McHale for his undergraduate final year project [EMP:carparkingMcHale2003]. The model was designed to aid learner drivers develop their parking skills. The car parking simulator is closely related to engineering—as well as applied physics—because a large part of the project was concerned with the mechanics of cars, from issues like steering and turning circles, to the

†Visual Fractions (`http://www.logo.com/cat/view/logotron-visual-fractions.html`) is an educational program created using Imagine Logo.

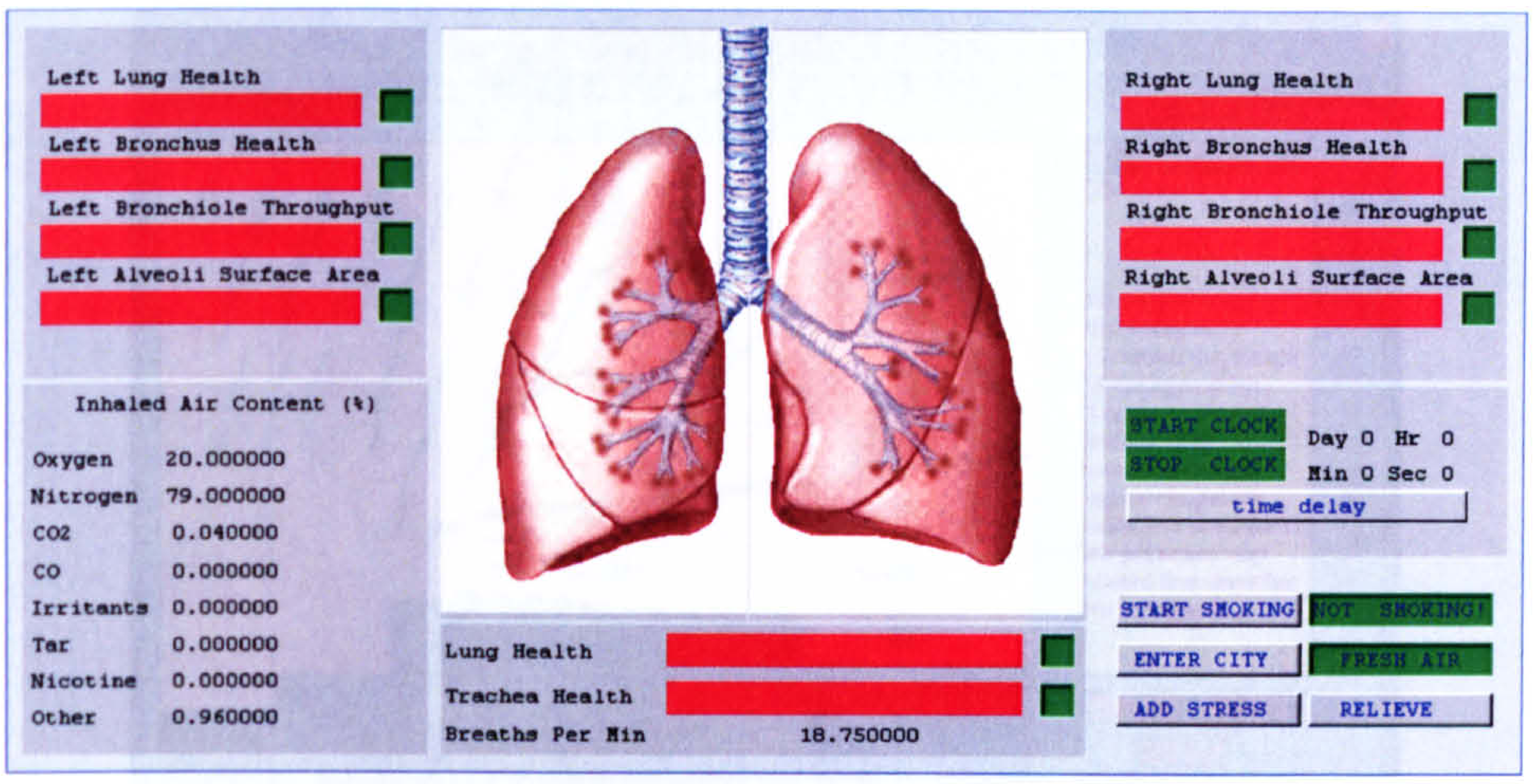


Figure 5.2: An educational artefact for understanding the effects of smoking on the body [WEBEM1].

effects of acceleration and braking. As shown in Figure 5.1, the model makes use of a steering wheel and pedals and is distributed on up to 4 computers to simulate different viewpoints on the car (windscreen, mirrors, and rear window). The flexibility of the EM model allows for challenging situations to be practised, such as an obstruction in the rear window or a misaligned door mirror. Another example from engineering can be found in D'Ornellas's project on learning about electronic circuits [DOr98].

Various EM projects have related to areas of science and in many cases the models have educational value. In the area of biology the ant navigation model stands out for its contribution to understanding ant behaviour, as explored in Chapter 6. Another educational artefact relating to biology or medical science was developed by Koorosh Heshmati (submitted to [WEBEM1]) for teaching and learning about the effects of smoking on the body (see Figure 5.2). There have been a number of models that have involved physics in many different ways. Games like billiards [EMP:billiardsCarter1999] and football [EMP:footballTurner2000] have required a physics model of ball motion for example. One model that has a strong physics element is the agent-based bridges model [WEBEM1]. The author of this model developed an environment for constructing simple bridges with partially elastic components that enabled experimentation with different configurations of components and weights.

Humanities has been one of the most promising areas of application for EM, as evident from research in EM together with Willard McCarty into humanities computing [BRM06]. The softer, subjective nature of humanities (compared to science) has much

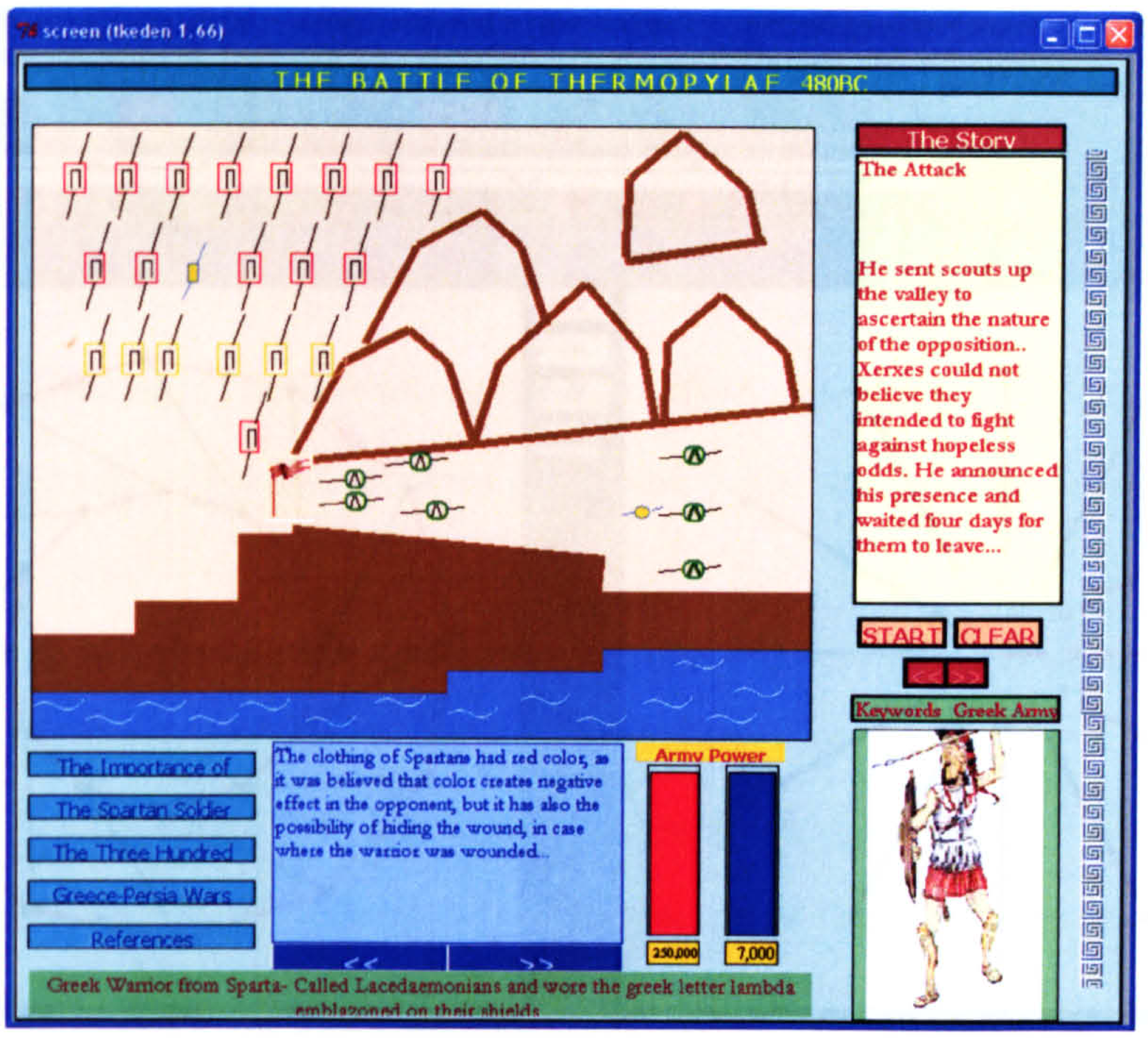


Figure 5.3: A model of the Battle of Thermopylae.

in common with EM, and is not supported particularly well by the systematic structure and formality of programming [BRM06]. A distributed model of the historical Clayton Tunnel railway accident is one of the best known EM models, and is discussed in more detail later in this chapter in connection with collaborative modelling. Care's final year undergraduate project [Car04b] explored the history of planimeters—mechanical devices for calculating area—and developed a number of accompanying models for gaining insight into their workings. An account of the learning involved in Care's model-building activity is given in the next chapter. Another historical model is from a famous Greek battle (Battle of Thermopylae), shown in Figure 5.3, built by Theodorou as part of a final year undergraduate project [The05].

There are a number of models relating to musical composition and analysis. The most significant work is Beynon's exploration of Erlkönig (a ballad by Goethe set to music by Schubert) shown in Figure 5.4 [EMP:kaleidoscopeBeynon2005]. The common ingredient in these examples from humanities is that they place the model-builder in an exploratory situation (e.g. in the Battle of Thermopylae). These models offer different qualities to more conventional uses of technology in the humanities. There are many resources available on the Internet about the Battle of Thermopylae including animations, maps and detailed narratives. The Battle of Thermopylae model, on the

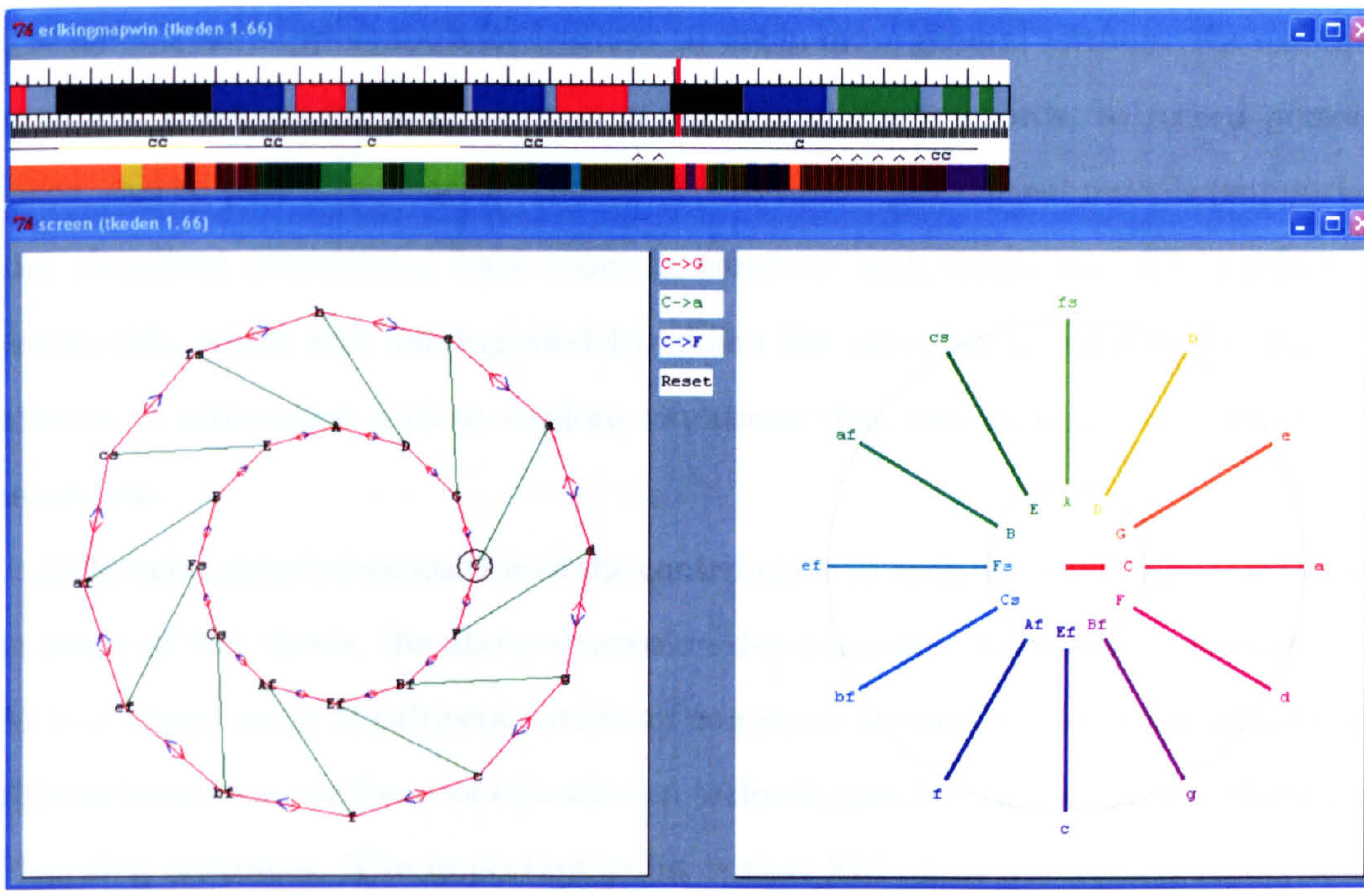


Figure 5.4: A musical analysis of Erlkönig [EMP:kaleidoscopeBeynon2005].

other hand, is distinctive in the manner in which it offers a unique exploration of a state. The significance of all of these humanities models is that the model-builder is placed in a state corresponding to a situation, and exploration of the state is unconstrained. A model-builder exploring the Erlkönig model is tracing Beynon's unique personal understanding of the music that has arisen through years of practical experience. The visualisation is an attempt to convey metaphorically the relationships between events in the poem and the harmonic effect of the music. This leads Beynon to distort the traditional cycle of keys in a fashion outside the scope of conventional interpretations—the twisting effect on left-hand side of Figure 5.4 reflects this distortion. Interacting with the model of the Greek battle and the model of the Clayton Tunnel accident, there is a sense that not only does the artefact represent a historical event, but that it is a personal interpretation of an event that can be interacted with as if for the first time. In this way, EM captures the aspiration for modelling identified by Dening: "[that we may] return to the past the pasts own present, a present with all the possibilities still in it, with all the consequences of actions still unknown" [Den98:p48].

Other models relating to music are composition focussed and include a keyboard model [Kin07], a musical score creator [EMP:musicWai2000] and a guitar tutor [Kon07]. While Erlkönig is about learning on a personal level by a specific model-builder, these three models are support tools for learners to use in more general settings. For example, the guitar tutor can be used to practice identifying given chords, to record personal chords, and to play a sequence of chords. These more 'generalised' models (supporting more ritualised interaction) have more in common with traditional ET artefacts or microworlds, whilst still offering model-builders the potential to subvert the standard patterns of interaction and to explore situations that are particular or personal to themselves.

Although a detailed exposition of the contribution in each of these subjects is beyond the scope of this thesis, the above discussion demonstrates the possible application of EM to a broad range of subjects outside of computer science. Some, if not all, of these subjects have been the focus of educational technologies that can support a wide range of learning outcomes. The important point is that EM offers a unique perspective on technology enhanced learning as is evident from some of the models introduced above. The next section looks particularly at language learning as a subject in which EM has a unique perspective as far as technology support is concerned and in which EM has the potential to positively change current educational practices.

5.1.2 Language learning

Foreign language acquisition is an area where learners have been quick to utilise new media and technology. Although language books are still seen by many as essential to learning a second language, the arrival of audio tapes, and later compact discs, has transformed the learning experience because of a new emphasis on the spoken form as opposed to the written form. Some language schools have developed techniques, such as the Michel Thomas method [Tho03], that do not make use of any written materials, but in most classrooms it is more common to see books and tapes being used to complement each other. The introduction of television and video for language learning followed after audio technologies, and now, the development of computers and the Internet is leading to new possibilities for language learning as Delcloque suggests [Del00].

Uses of educational technology for language learning are often referred to as Computer-Assisted Language Learning (CALL) or Technology-Enhanced Language Learning (TELL). Warschauer, an eminent scholar in the area of technology for language learning, suggests that there have been roughly three phases in the development of CALL: the *behaviouristic* phase, the *communicative* phase, and the *integrative* phase [War96]. The three phases are described in Figure 5.5. In the *behaviouristic* phase (roughly appearing in the 1970s & 1980s), computers were used to support behaviourist theories and programs consisted of practice, drills and tests. The computer is a particularly effective medium for such activity due to the consistency of materials and the ease of repetition. In the *communicative* phase (during the 1980s & 1990s), in response to a perceived lack of authenticity in the first phase, an emphasis was placed on the computer to take a more communicative approach to teaching. This led to programs where the computer became the language tutor, offering the learner a choice of material and routes through the material. These led to the development of games and other programs that stimulated language learning. Both the first and second phases of CALL highlight the paradigmatic conflict in educational technology, as described in §1.1.4, in this case between the way languages are learnt in an everyday setting and the methods that are employed for computer-based language learning. This is partly because the methods seek to use the computer as a teacher instead of a support for learning. The *integrative* phase (starting around the turn of the century), the one that Warschauer would say we are currently in, has come about because of the integration of multimedia and the Internet. The ease with which multimedia can be combined through hypermedia and made available dynamically through the world wide web has led to new forms of CALL. For example, students learning not-so-commonly-taught languages such as Thai are not only able to access Thai language learning resources[†] but they can also gather up-to-date sources from Thai websites and communicate with Thai people using email and text-based chat for reading and writing skills and voice chat for speaking and listening skills. The Internet itself has become a tool for CALL because language learners around the world can communicate cheaply and quickly using email, instant messaging, Internet voice calls and video conferencing. In this way there are some positive outcomes for using the Internet as a support (rather than a teacher), but there are still large research agendas concerned with emphasising the computer as teacher aspect of language learning. For example, researchers in adaptive hypermedia systems advocate that text, images and videos that adapts to individual learner needs are ben-

[†]See the extensive language site developed by a school in Thailand at `www.learningthai.com`.

Stage	70s-80s: Behaviouristic CALL	80s-90s: Communicative CALL	90s-00s: Integrative CALL
Technology	Mainframe	PCs	Multimedia and Internet
English-Teaching Paradigm	Grammar-Translation & Audio-Lingual	Communicate Language Teaching	Content-Based, ESP/EAP
View of Language	Structural (a formal structural system)	Cognitive (a mentally-constructed system)	Socio-cognitive (developed in social interaction)
Principal Use of Computers	Drill and Practice	Communicative Exercises	Authentic Discourse
Principal Objective	Accuracy	And Fluency	And Agency

Figure 5.5: Warschauer's three phases of CALL [War00].

eficial because it teaches the most relevant aspect to the learner‡. Starting from such a viewpoint is equivalent to the behaviouristic and communicative phases of CALL in the 1970s and 1980s, and while adaptive hypermedia may offer benefits to learners in some situations, it is not comparable to the flexible uses of the Internet for language learning that Warschauer sees emerging in the current *integrative* phase [War96] or to the experimental approaches to language learning suggested by EM later in this section.

While it is true that multimedia, hypermedia, and the Internet are being used for *integrative* CALL, many applications of CALL (e.g. latest language-related technologies promoted at Warwick's e-learning Showcase Day 2007 [UoW07]) would be classified by Warschauer as *behaviouristic* CALL and most exemplify traditional audio/video techniques. For example, the current iPod (or mp3 player/phone) craze is seen as an ideal opportunity for language learning on the move. This is pedagogically similar to 20 years ago when Walkmans (portable cassette players) were being used for language learning. There have been technical advances in that mp3s are quicker to copy from device to device than cassette tapes, but the style and content of the learning remains the same. In this way current practice demonstrates that there has been little departure from conventional language learning techniques. The main improvements that have arisen so far from educational technology are to communication and the transmission of information. In many respects educational technology remains unsuitable for providing support for learning in an everyday sense (as depicted in Figure 1.1 on page 13) because it lacks experimental, flexible and meaningful characteristics.

‡See [DeB02].

5.1.2.1 Approaches to language learning

Before evaluating what EM may contribute to language learning, it is essential to understand the approaches to the subject. First of all, as acknowledged by Milton's review of language learning and technology [Mil02], language learning is difficult because it involves a combination of learning explicit formal vocabulary and grammar rules, and developing fluency which may include informal and cultural aspects. The basic approaches to understanding language learning fall within two camps. In the one camp there is the behaviourist approach which views language learning as essentially a repetitive engagement with vocabulary and phrases. In the other camp there is the cognitive approach which views language learning as a more gradual subtle process of gaining ability and understanding through experience of a language in ways that are not necessarily formalised. The behaviourist approach is aligned to the formal aspect of language learning where explicit vocabulary and grammar rules are emphasised, whereas the cognitive approach is relevant to the informal aspect of gaining fluency in a language. So although these two approaches are generally seen as coming from separate camps, successful language acquisition is likely to include both [CHM04].

The use of educational technology, such as digital audio and video, generally follows the behaviourist approach. Most CALL, especially from the behaviouristic and communicative phases, focuses on supporting the formal aspect of langugage learning. This may be due to the emphasis that has been placed on formal interpretations in software development. Brian Cantwell Smith identifies two aspects to the interpretation of software [Smi96]. First, there is the relationship between the *program* and the *abstract process* that is treated explicitly as having a single unambiguous interpretation[Smi96]. Second, there is the relationship between the *abstract process* and the *world* or *subject matter* which can have many interpretations and cannot be easily formalised [Smi96]. Smith criticises software development for focussing too heavily on the first interpretation and not giving enough attention to the informal aspects of software [Smi96]. This imbalance suggests that software leans towards supporting domains that can be formulated explicitly and that have one unambiguous interpretation. It is therefore well-suited to repetitive tasks in language learning, as is evident from the common use of software for spelling correction, vocabulary reinforcement, and grammar suggestion. The support for the informal aspects of language learning using computers

Language learning...

1. is an incremental process of gaining understanding of a language and its usage

2. requires repetition of familiar situations and interactions

3. is deeply embedded in context and culture

4. does not require a complete understanding of a language to be able to make use of it for meaningful communication

Figure 5.6: The characteristics of language learning that are well-aligned to EM.

is not obvious. Interacting with a computer (using CALL from the behaviouristic or communicative phases) may teach the definition of "tea", but it is unlikely to be able to explore the implied meanings of an ambiguous phrase out of context such as "would you like to come for tea?". Such situations that are natural for native speakers can sometimes lead to misunderstandings between native and non-native speakers†. Where misunderstandings occur, there is a mismatch in expectation, and this realisation is what leads to 'conceptual change' in the learner as described by Strike & Posner [SP85] (see §1.1.4). Language learning in the world, in an everyday setting, involves informality that educational technology with its background in logic does not satisfactorily support. The argument in Chapter 3 is that EM offers support for learning (and more generally computing) that can involve both formal and informal aspects. Therefore EM should be well-suited to supporting language learning.

Figure 5.6 highlights some aspects of language learning that can be closely related to characteristics of EM as explored in Chapter 2. This is to show that EM has much in common with language learning and that the activity of learning a language has many of the same ingredients as using EM to create an artefact. In the following case study involving learning Thai language, it is shown that through these similarities EM is much better aligned to supporting not just the formal aspects of language learning but also the informality that is required for deeper learning leading to fluency and cultural transfer.

†One of my friends was pleasantly surprised when expecting a cup of tea to be served what he would have called dinner!

5.1.2.2 A case study from learning Thai language

To illustrate the potential of EM for language learning and the close connection between language learning and EM, an exercise in creating a language learning artefact is undertaken by the author. The topic of this case study is "buying items at a market in Thailand". The specific situation could be: an individual has a shopping list of grocery items and is going to visit a market to find and buy the items. This situation was chosen because it is an activity that happens in most countries, it would be familiar to most people, and it invokes a variety of language aspects:

- understanding items on a shopping list (items and quantities)
- finding items within a market (navigation)
- asking about items (vocabulary of items)
- paying for items (numbers and currency)

These are some of the aspects that can be formalised or that can be made explicit and expected from a visit to a market. There are other informal aspects that are just as likely to occur but which might not be specifiable so clearly, or which cannot be formalised in a computer-based artefact. For example, how to deal with unexpected occurrences like a market seller trying to engage in everyday conversation. The informal aspects may be tied up with culture and context like a situation where it is not known whether the price is fixed or whether you are expected to bargain for the price.

There are two sides to this case study: the construction of an artefact that represents interactions at a market, and the exploration of the artefact by the learner (in this case, the author). These need not be viewed as two separate stages (as demonstrated with the TLJ model in §4.1); both activities can be interdependent in an EM environment and the learner is likely to be switching from one activity to the other as their learning progresses. Similarly, the learner will often be involved in two related activities: understanding the language, and understanding (or making sense of) the situation. For example, in a market the learner might be grappling with questions about the prices, whether there are any hidden charges or discounts, or whether bargaining is expected. In an unfamiliar cultural context understanding the use of language is often intimately linked with understanding the situation.

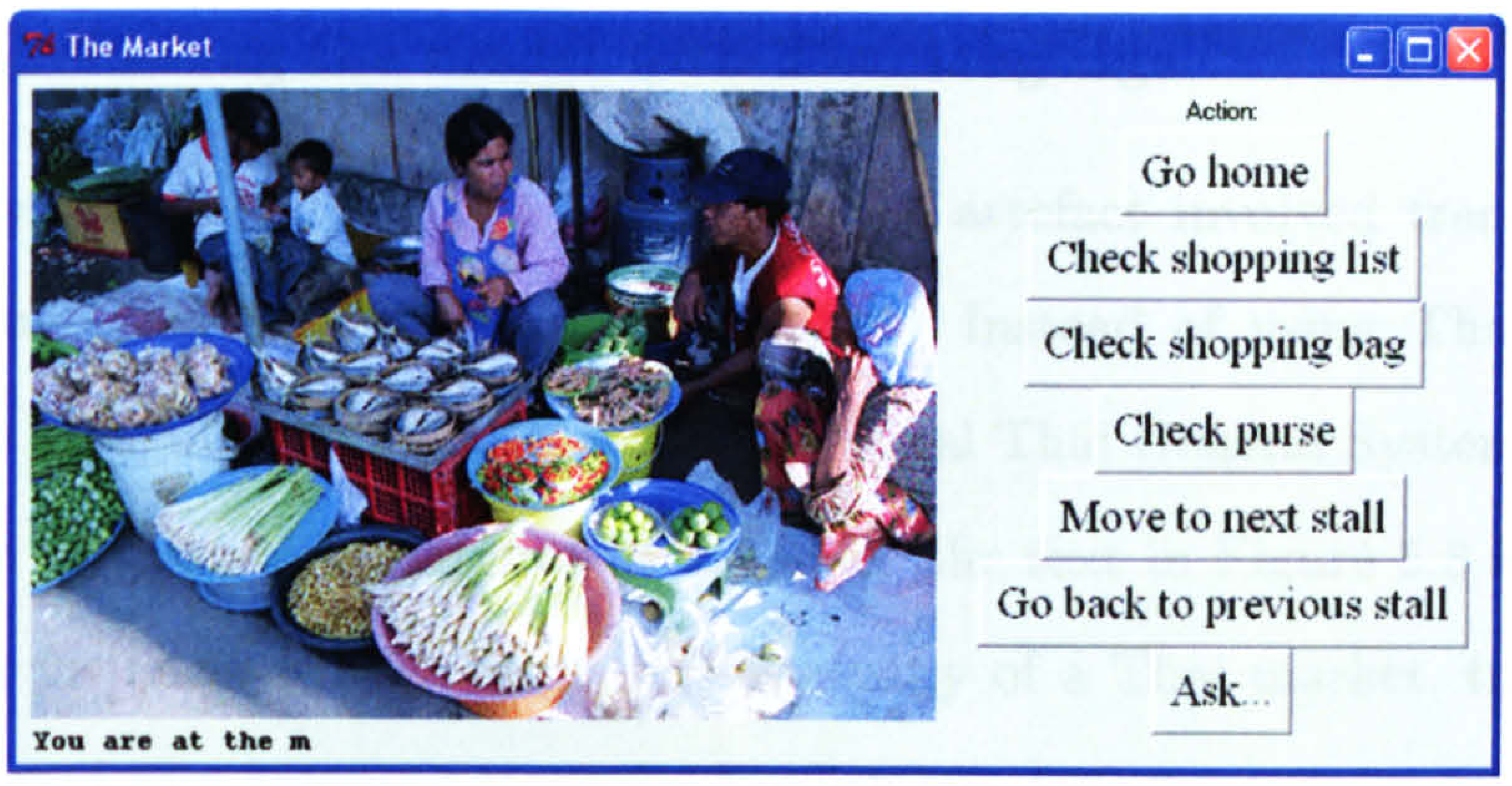


Figure 5.7: The initial market model before adding Thai.

5.1.2.3 Establishing an initial model based on existing understanding

The initial construction of the artefact involved using existing knowledge to create an environment that was similar to a market. This activity, like any other EM activity, is guided by the modeller or learner in such a way that the artefact will represent their idea of a market, and their personal experience of being at a market.

An initial artefact was constructed with typical photos from a market as shown on the left of Figure 5.7. This included images of market stalls containing different items (fruit, vegetables, meat). At this point, a learner could browse the 'virtual' market as a passive observer, unable to perform any actions in the environment. The next stage was to extend the artefact with the potential to enquire about and ask for items on the market stalls. A further development to the model enabled the keeping of a shopping list and a shopping bag, with the idea of buying items and placing them in the shopping bag in order to fulfil the shopping list. Figure 5.7 is a model of a market situation, constructed using the GEL notation [EMP:gelHarfield2007] (see Figure 4.5 on page 95) that was developed by the author as introduced in Figure 4.5 on page 95.

This was considered a basic starting point for the market artefact. At this point, the artefact was at a basic conceptual level, using only English language, as a means of formalising my personal understanding of interactions at a market (see Figure 5.7). This may be a typical conception of a market for an English person. In this way EM supports the qualities of language learning as necessarily incorporating repetition of existing understanding and practice as highlighted in Figure 5.6.

5.1.2.4 Embellishing the model with Thai language

The first steps towards creating a Thai language artefact involved translating some of the text contained in the artefact into Thai. Instead of using Thai characters, which may be unfamiliar to most learners, the Royal Thai General System [Kan06] for transliteration to roman characters was used (see the text in Figure 5.8 for example). Although this distances the learner from the reality of a Thai market, transliteration is commonly used as a stepping stone for English speakers towards understanding the Thai alphabet.

Even when translating some of the basic elements of the artefact from English to Thai as shown in Figure 5.8, difficulties arise and changes must be made to the artefact. On a sentence level, there are components that must be reordered or reconstructed. For example, if you translate "how much are the bananas?" into Thai then you can literally say "banana how much?". In Figure 5.8 the option "...tao rai?" meaning "how much?" is selected and "gluay" meaning "banana" has been entered in the text box. In modifying the artefact in response to translating the "how much?" question, the subject of the sentence has moved to the beginning and, in the artefact, the position of the subject text box has changed. It may seem obvious that sentences cannot be translated word for word, but building an artefact magnifies this fact to the learner (cf. Jonassen's maxim that if you can build a model of something then you understand it [Jon06]). At least in my interactions with the market model, I was forced to think carefully about how to express sentences in Thai and consider the differences between Thai and English. The benefit of EM over traditional programming environments is that such refinements to the artefact can occur as part of the interaction with the artefact and as part of the learning process (as opposed to the 'correcting' of programs which requires a developer or at the very least returning to the development environment). As the artefact is developed, the refining of the details goes on endlessly. If the artefact was modified to use Spanish then there would no doubt be other details to be considered—on a word level the artefact would need to cater for the gender of the words. A specification of the artefact could not capture all the possible details that might go into the artefact, and so it is natural to need to develop and refine these details as the artefact is developed. In this way EM more naturally supports the quality of language learning as incremental and not necessarily complete from the beginning as highlighted in Figure 5.6.

Figure 5.8: The Thai market model.

5.1.2.5 Exploring cultural aspects of Thai language

The "how much?" example is concerned with the more formal aspects of Thai language because it is related to the ordering in sentences which can be explicitly defined to some extent. There are other issues addressed that relate to more informal aspects of Thai language. For example, if you asked the question "can I have 5 apples?" in Thai then you might simply get the answer "yes" because "can i have" would likely be interpreted by the seller as a question instead of a request for apples. The second option in Figure 5.8 ("kor... dai mai?") is a fairly close translation of "can I have?", according to a native Thai speaker, but such is Thai language that you can actually say it more bluntly (from an English point of view) as "I want..." and add an extra word on the end to make it polite (such as "krub" for a man or "ka" for a woman). There are many similar subtle issues that could be taken into account in the Thai market model, and through interaction the model can be refined to reflect new understanding by the learner. The use of dependency for flexible redefinitions is one example of how EM can help a learner take account of nuances in a language. The polite sentence endings "krub" and "ka" which change depending on the sex of the speaker are one such example. By redefining a small set of definitions as in Figure 5.9, every sentence in Figure 5.8 is appended with an appropriate ending. The 'ending' observable in Figure 5.9 is dependent on the sex of the speaker and therefore any change to the 'sex' observable will select the appropriate ending. More complicated endings, such as those appropriate for speaking to a close friend and to make the sentence cuter, can be easily incorporated by redefining the ending observable as given in the last definition

```
/* Adding polite endings to sentences */
howmuchare_text is "... tao rai" // ending // "?";
mayihave_text is "kor ... dai mai" // ending // "?";
doyouhave_text is "mee ... mai" // ending // "?";
idontwant_text is "mai aow khorb khun" // ending // ".";

/* Defining the ending */
sex = "female";
ending is sex=="male" ? "krub" : "ka";

/* More complicated ending */
cute = 0;          ## set to 1 to sound cute
closefriend = 0;   ## set to 1 if speaking to close friend
ending is (cute ? "na" : "") //
    (sex=="male" ? "krub" : (closefriend ? "ja" : "ka"))
```

Figure 5.9: Redefinitions to the market model to add Thai polite endings to sentences.

in Figure 5.9.

Another cultural aspect of the language incorporated in the market model is that bargaining is accepted and expected in a Thai setting. Refining the model to reflect the structure and vocabulary of different contexts or cultures as is necessary to learn about bargaining is much the same as making any other interactions with a model (e.g. the redefinition of the 3D room in Figure 4.8 on page 102 or the modification of the HTML environment in Figure 4.14 on page 113). The use of dependency in EM, as in Figure 5.9, provides a support for exploring issues where language learning is deeply embedded in context and culture as highlighted in Figure 5.6.

5.1.2.6 The advantages of never finalising a model

As described in §2.2.5, there is not necessarily such a thing as a 'complete' artefact when engaging in EM. Although the Thai market example may reflect all of the learner's current understanding, it still has the potential as an instrument for developing the learner's understanding further. Similar sentiments are expressed by Kynigos in a paper on 'half-baked' microworlds where he suggests that partially developed microworlds offer potential for constructionist learning because they encourage learners "to build on them, change them or de-compose parts of them in order to construct an artifact for themselves or one designed for instrumentation by others" [Kyn07]. In the market model, the learner might want to take a step towards learning the Thai alphabet, or may be interested in developing the market artefact towards conducting business. As with the potential for using language, there are many possibilities for the market model and it can serve as a 'half-baked' starting point for learners as Kynigos advocates. In this way EM supports the qualities of language learning as not requiring a complete understanding of the language in order to make use of it for meaningful communication as highlighted in Figure 5.6.

The artefact itself might not appear to be particularly useful to others once the learner has finished with it. As characterised in Chapter 1, learning is an active process that requires interaction to benefit from any artefacts. Model-building with EM involves essentially two elements: *artefact* and *interaction*. The potential for developing understanding is in the process of constructing and exploring the artefact. Therefore, unlike most conventional educational technology where the program stands out, the EM artefact may not appear on its own as impressive or useful, but through interactions the usefulness of the artefact may emerge.

There is potential for other learners to make use of the artefact for learning Thai in that the artefact could be used as scaffolding for someone less experienced in Thai. Although this is not explored in this case study, the artefact could be given to a beginner and through interaction the person may be able grasp some elementary Thai. A more experienced learner would be able to extend and refine the model in light of their knowledge of Thai language and related cultural issues. As Kynigos states, the benefit of 'half-baked' artefacts is that they can be unpicked, built upon and combined for the benefit of the learner [Kyn07], and EM provides a suitable environment (i.e. using dependency) in which this activity can occur.

5.1.2.7 Evaluation of case study

One criticism of the case study is the extent to which attention may be placed on model-building that is not connected with the target language. This concern is inevitable with any subject, but it could be argued that learning about the mechanics of the situation and the context is also valuable in that it is emphasising the characteristic that all learning is situated and relating to authentic experiences. In cases where there are suitable models to be reused (or 'half-baked' models), much less time could be taken up with building a model of the situation. In the case study Rungratanaubol's Thai number model [Run02] was reused for talking about quantities and prices at the market. It is possible that existing models may be used for language learning in a

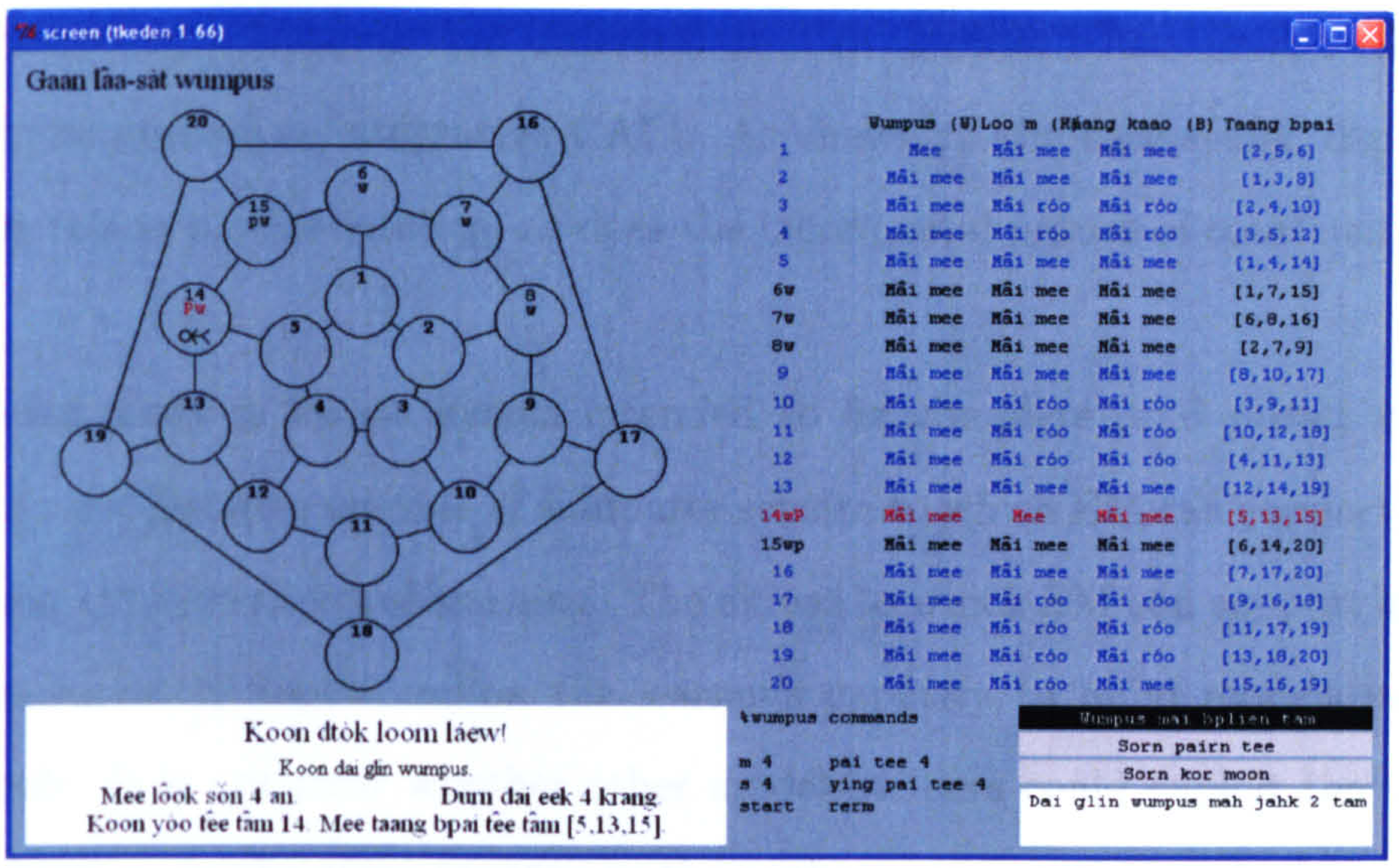


Figure 5.10: The wumpus model in Thai.

specific domain. As an example, I took the wumpus model by a student as a submission for WEB-EM-1 [WEBEM1] (discussed further in Chapter 6) and, as a model-building exercise, I translated the game into Thai (see Figure 5.10). Such activity requires only a minimal amount of model-building, and hence the challenges were all related to understanding the language in a specific situation. The types of learning that were encountered include: increased vocabulary, improved grammar knowledge, skills for reading and writing, specific domain language understanding for the wumpus game. It is not necessary to undertake the building of a large model in order to use EM for language learning. Furthermore, artefacts already containing suitable language, like the wumpus model in Thai, could be used initially by other learners with little or no model-building experience.

The Thai market case study demonstrates that the practice of EM is well-aligned to the nature of language learning in an everyday sense as described in Figure 5.6. EM as a support for language learning can also be seen as departing from conventional approaches to CALL (e.g. the early uses of CALL described by Delcloque [Del00] or the current usage of mp3 players and other technology used at university level [UoW07]). In particular, EM supports an approach to language learning that surpasses the behaviouristic and communicative phases of CALL defined by Warschauer [War96] because it is concerned with more than transmission and communication. It fits into the current integrative phase of CALL because model-building using EM can be seen as a cognitive support for learning instead of a teacher or communicative tool. However, EM has many distinctive features that differentiate it from other educational technologies that might be classed as integrative CALL. As shown in the case study, dependency plays a key role in model-building, as does the incremental nature of construction with EM.

This case study is by no means intended to be complete, and purely offers an example of a subject area outside of computer science to which EM can support more of the everyday characteristics of learning. The extent to which EM can support language learning is currently dependent on the learner's capacity to build models using the current tools. It is not clear whether other model-builders could exploit the tools for language learning in the same manner as the author—with better tools this might be possible. A wider investigation is needed to evaluate the application of EM to language learning. The discussion of the Thai market model only provides an insight into what might be possible (using better tools) on a much larger scale with different languages and levels of learning.

5.2 Learning through lectures and presentations

Chapter 4 explored a specific example of constructing a presentation environment with the 3D room viewer to teach and learn about 3D to 2D transformations in computer graphics. It demonstrated that EM tools are flexible because we can reuse and combine models (see Figure 4.10 on page 105) to be taken in a new context relevant to the model-builder. A concrete example of development has discussed how a model-builder plays the role of student, teacher and developer in a single environment (see Figure 4.16 on page 115). This section examines the potential of the EM Presentation Environment (EMPE) [EMP:empeHarfield2007]—see Figure 5.11—as a teaching and learning tool, contrasting its use with more common tools for preparing and giving lectures or presentations.

The EMPE was first used for the presentation of a paper at ICALT 2006 [BH06] as shown in Figure 5.11. During the presentation I demonstrated that: the clock model could be exercised in ways that need not be preconceived (e.g. linking the clock model to the current time); the presentation could be changed on-the-fly (simultaneous construction and use); and the EMPE itself could be changed in response to audience

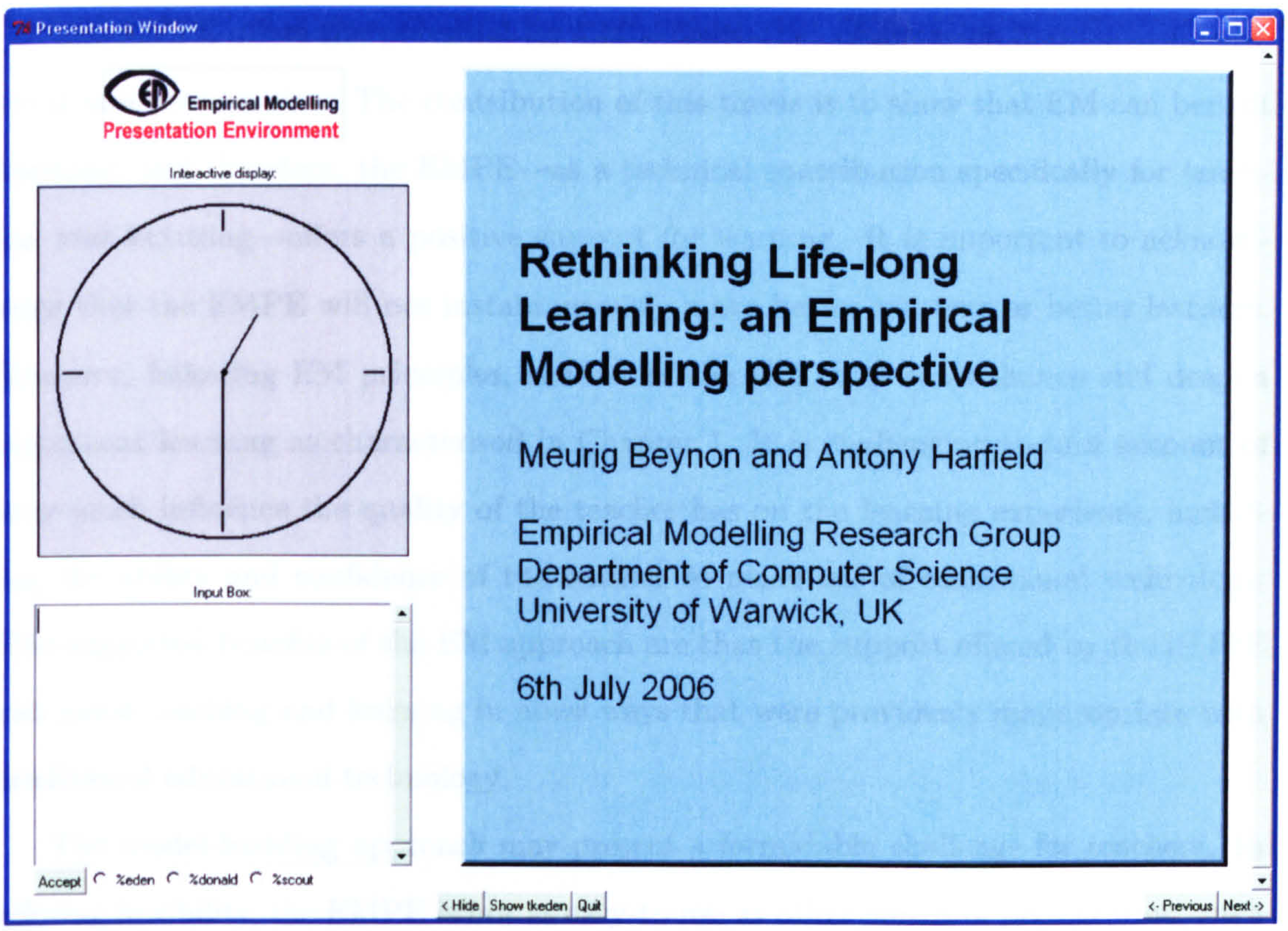


Figure 5.11: An early version of the EMPE as used for a presentation at ICALT 2006.

feedback or other factors in the environment (e.g. varying the slide colour based on the time shown in the clock model). During the *preparation* of the presentation, I was not just creating the presentation slides and model, but also extending the EMPE, the HTML Environment and the GEL notation at the same time, as discussed in §4.2.2.

The development of the EMPE, as discussed in §4.2.2, has shown that, as a tool for presenting, it has a number of features that differentiate it from other presentation tools. Firstly, the EMPE allows the model-builder to create and reuse models (such as the 3D room viewer in Figure 4.8 on page 102) within and connected to the presentation. Parts or all of the model can be displayed within the presentation window (although the assumption here is that there is enough screen space)—for example the clock model used in the presentation for ICALT 2006 [BH06] shown in Figure 5.11. The presentation can be connected to the model by creating html slides that are dependent on observables in the model (using the `<eden>` tag) and creating html slides that trigger agent actions within the model (using the `<script>` tag). A model-builder has the potential to manipulate and change slides on-the-fly during the preparation of the presentation and during the presentation itself if required.

The EMPE offers the potential for EM activity (i.e. observation and experimentation) in a presentation. The contribution of this thesis is to show that EM can benefit learning, and therefore, the EMPE—as a technical contribution specifically for teaching and lecturing—offers a positive support for learning. It is important to acknowledge that the EMPE will not instantaneously make better teachers or better learners. However, following EM principles, the use of the EM tools can enhance and deepen significant learning as characterised in Chapter 1. It is challenging to take account of how much influence the quality of the teacher has on the learning experience, including the ability and confidence of the teacher to make use of educational technology. The suggested benefits of the EM approach are that the support offered by the EMPE can assist teaching and learning in novel ways that were previously inappropriate with traditional educational technology.

The model-building approach may present a formidable challenge for teachers. In offering flexibility, the EMPE is not as easy-to-use as other means of presentation, such as using a blackboard or projecting presentation slides. The simplicity of 'writing on the blackboard' is lost, and so is the simplicity of creating presentation slides which require very little effort on the part of the presenter—instead the EMPE requires the model-builder to write slides in HTML. The EMPE in its current state offers a very primitive presentation tool, but as an EM model there is potential for it to evolve, just as it evolved from an environment for documenting models (as discussed in §4.2.2)—future features might include slide effects and timed actions as are found in other presentation tools, or specialist input devices such as Wii remotes† that are not standard features of presentation tools.

5.2.1 Traditional presentation tools

Presentation tools, such as Microsoft PowerPoint‡, are commonly used as teaching aids in academia and industry. Lecturers use presentation tools to teach undergraduates, and researchers use presentation tools to demonstrate their work to other academics at workshops and conferences. In the business world, the presentation tool has become an essential office application, not just for marketing purposes, but also for training

†I have developed an extension to tkeden that enables Nintendo Wii remote controls to be used as input devices, along with some examples of how they could be used to augment existing models [EMP:wiimoteHarfield2007].

‡Microsoft and PowerPoint are registered trademarks of Microsoft Corporation.

and education. Microsoft PowerPoint has become the most popular and widely used presentation tool [Wik07a]. Using a presentation tool like PowerPoint is considered the standard, and in many contexts is considered a requirement for a 'good' presentation. Despite this tool being regarded as one of the most successful applications for the PC, the role it plays has been criticised by some, such as Norvig [Nor03], for 'dumbing down' presentations. Research into undergraduate courses shows that although students generally prefer PowerPoint lectures [BC03] [FB02], the use of PowerPoint often does not have a positive effect on student grades [SH00].

In a long exposition on the failures of PowerPoint, Tufte [Tuf03] states that the basic problem with PowerPoint is that it "is entirely *presenter-oriented*, and *not content-oriented, not audience-oriented*". In terms of learning, the presentation tool is not a *teaching* aid but a *teacher* aid—it makes presentations easy for the presenter. Tufte says that the focus on the presenter is damaging to both content and audience and results in:

> "foreshortening of evidence and thought, low spatial resolution, a deeply hierarchical single-path structure as the model for organizing every type of content, breaking up narrative and data into slides and minimal fragments, rapid temporal sequencing of thin information rather than focused spatial analysis, conspicuous decoration and Phluff, a preoccupation with format not content, an attitude of commercialism that turns everything into a sales pitch." [Tuf03]

He calls this the 'cognitive style of PowerPoint', which I am going to compare to the use of the EMPE [EMP:empeHarfield2007] with specific reference to the 3D room presentation [EMP:graphicspresHarfield2007].

Before the arrival of PowerPoint presentation tools, lecturers mostly used blackboards as teaching aids. A teacher using a blackboard has to construct material on-the-fly, working through arguments in real-time instead of serving up bullet points on ready-to-digest slides. The blackboard approach is more content-oriented, and makes for a narrative that the audience can follow (or if the audience gets involved then they can guide the narrative). Tufte agrees that presentation tools should show more support for this style of *good teaching* instead of supporting the 'cognitive style of PowerPoint': "Teachers seek to explain something with credibility, which is what many presentations are trying to do. The core ideas of teaching—*explanation, reasoning, finding things out, questioning, content, evidence, credible authority not patronizing authoritarianism*—are

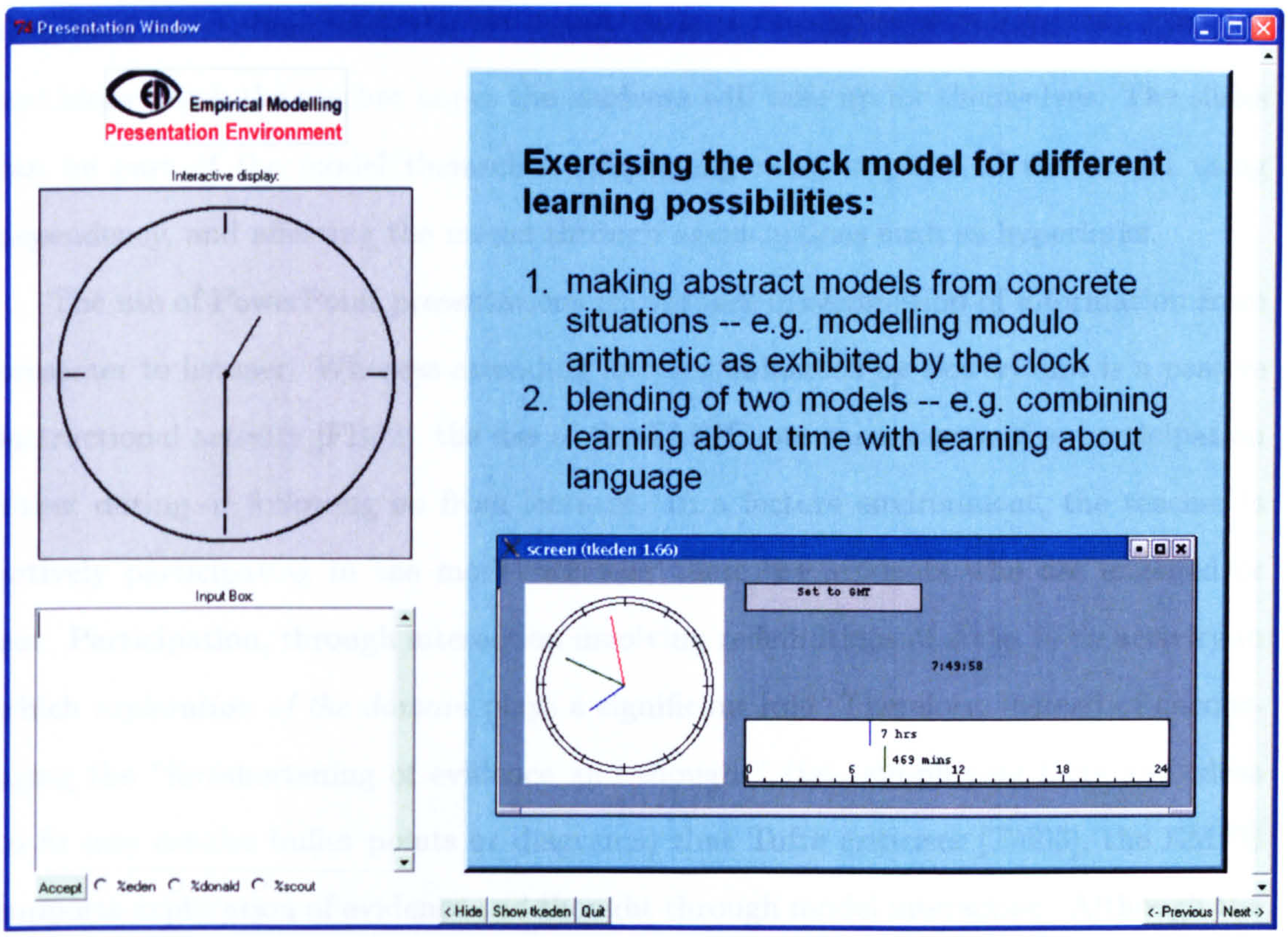


Figure 5.12: Introducing the clock model in the ICALT 2006 presentation.

contrary to the hierarchical market-pitch approach [of PowerPoint]" [Tuf03]. In examining the use of the EMPE, as in the computer graphics presentation in §4.2.2.3, it is shown that these core ideas of teaching highlighted by Tufte are much better supported by the EMPE than by PowerPoint.

5.2.2 Contrasting PowerPoint with the EM Presentation Environment

One possible reason for the cognitive style of PowerPoint is that it is limited to static information (slides with text, graphics, audio, and video). Tufte points to "a deeply hierarchical single-path structure as the model for organizing every type of content" as the problem [Tuf03]. PowerPoint provides support for communicating information, that Tufte calls the "sales pitch", but little support for the process of teaching (or learning about) the ideas behind the information. There is very little scope for demonstrating practical elements relating to the content within a presentation. Contrast this with the EMPE where there are circumstances when the model can take centre stage, as shown in Figure 5.12, and the slides are more like pointers to parts of the model to be demonstrated and exercised (e.g. the scripts in Figure 4.9 on page 103 offer some suggested interactions with the 3D room). It is the model that contains the concepts and ideas which the teacher hopes the students will take up for themselves. The slides can be part of the model themselves, displaying relevant parts of the model using dependency, and affecting the model through agent actions such as hyperlinks.

The use of PowerPoint presentations encourages dissemination of information from presenter to listener. Whereas attending lectures enhanced by PowerPoint is a passive instructional activity [FB02], the use of the EMPE can encourage active participation either during or following on from lectures. In a lecture environment, the teacher is actively participating in the model whether there are students who are engaged or not. Participation, through interaction involving redefinitions of state, is an activity in which *exploration of the domain* plays a significant role. Therefore, instead of encouraging the "foreshortening of evidence and thought" (i.e. simplifying facts and ideas to fit into concise bullet points or diagrams) that Tufte criticises [Tuf03], the EMPE supports exploration of evidence and thought through model interaction. Although the slides in the EMPE may suffer from the same "low spatial resolution" as PowerPoint [Tuf03], the model can offer a depth to the exploration space by allowing interaction and experimentation. Rather than foreshortening thought, model interaction and experimentation gives rise to the potential for extending evidence and thoughts. For example, when discussing orthogonal projection in the 3D room presentation, by redefining the eye distance a practical appreciation of the relationship between making the eye distance large and orthogonal projection could be obtained (see Figure 4.8 on page 102). A few fragments of script on a slide can be exercised to understand the model, as depicted in Figure 5.13, and then modified to find new ways of interacting with the model (e.g. in Figure 5.13 the minute hand is being redefined to be half an hour fast), potentially leading to practical evidence for such understanding.

Tufte also criticises PowerPoint for "breaking up narrative and data into slides and minimal fragments" [Tuf03]. Sometimes the use of bullet points is justified as a means of reminding the presenter (like notes on a piece of paper) of the story they are telling, and similarly as a reminder of the story for the audience should they return to the presentation slides after the presentation. However, there is a tendency for the presenter to make the bullet points the *whole* story and rely on them more than their own knowledge of the story. This might be where Tufte's disapproval lies, and he

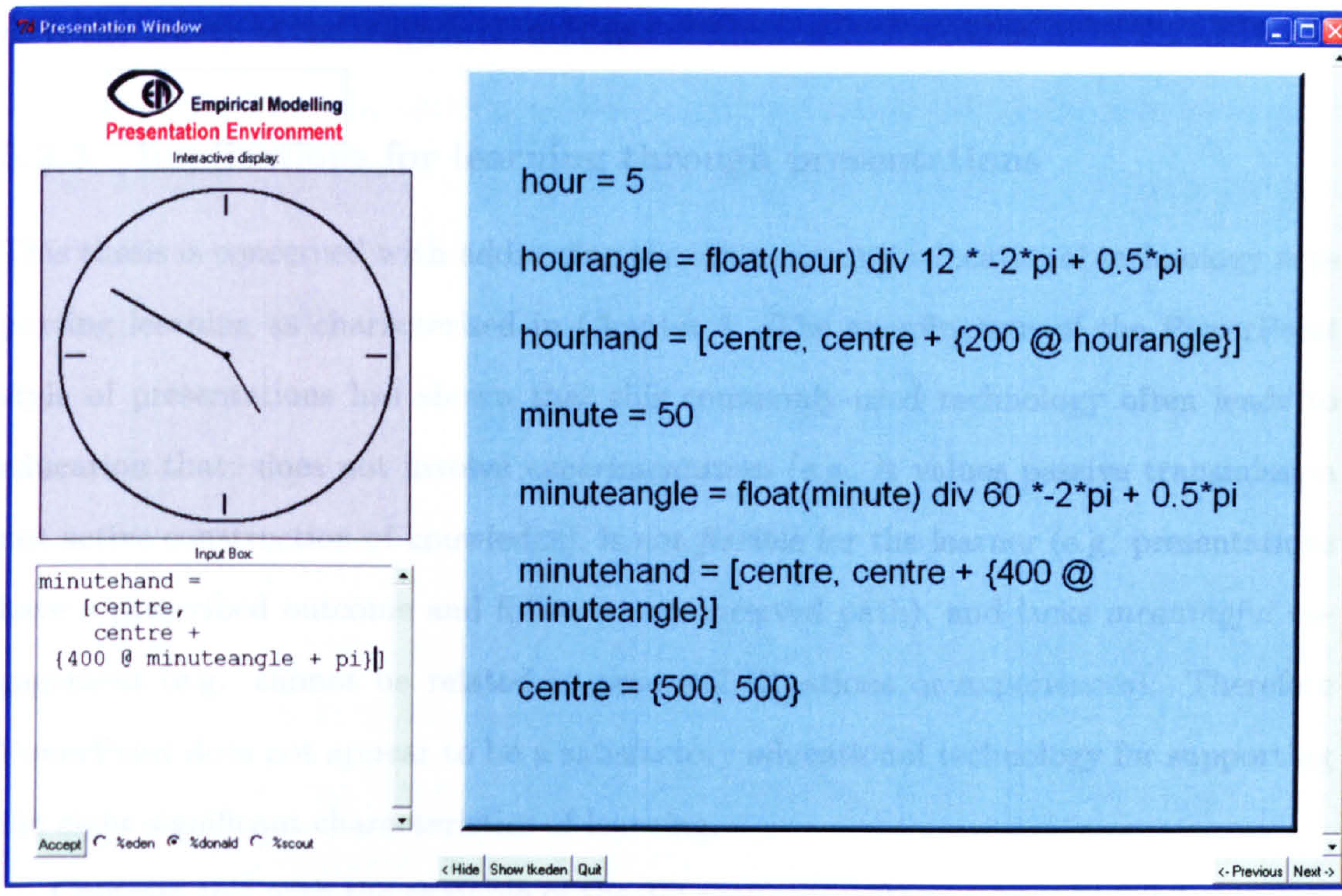


Figure 5.13: Making redefinitions to the clock model during a presentation.

blames PowerPoint for encouraging this tendency. Although there is nothing to stop a presenter using the EMPE for similar lazy presentations, the EMPE changes the focus of the presenter to the model. The EMPE encourages the presenter to explore the model and to tell a story with the model. Furthermore, a student exploring the presentation in the environment can create further narratives that differ from the original presenter's narrative. For example, Beynon's extensive walk-through [Bey07b] is a story of his interactions with the 3D room model, which happen to uncover certain aspects that are particularly relevant to 3D to 2D transformations in computer graphics. As a subsequent explorer of the 3D room model, I have developed my own story of the 3D room model which relates to how the EMPE can assist technology-enhanced learning in a unique way.

The final criticism of the cognitive style of PowerPoint is that it is too concerned with "conspicuous decoration and Phluff, a preoccupation with format not content" [Tuf03]. The EMPE is not concerned with decorative slides, design templates and fancy graphics (as is evident from Figure 5.13). Slides are created from basic html, usually as a commentary on the model. Clean, smart, or fanciful slides are not important, it is clear, intricate, imaginative and interesting models that have the focus of the presentation. Individual style is still important, especially when exercising the model, as is the ability of the teacher or student.

5.2.3 Implications for learning through presentations

This thesis is concerned with addressing the issues around educational technology supporting learning as characterised in Chapter 1. The examination of the PowerPoint style of presentations has shown that this commonly-used technology often leads to education that: does not involve *experimentation* (e.g. it values passive transmission not active construction of knowledge), is not *flexible* for the learner (e.g. presentations have a prescribed outcome and follow a preconceived path), and lacks *meaningful* engagement (e.g. cannot be related to practical situations or experiences). Therefore PowerPoint does not appear to be a satisfactory educational technology for supporting the eight significant characteristics of learning.

Contrast this with the example of the 3D room presentation from §4.2.2.3, which is focussed on observing and exploring a model. The EMPE can be used for experimentation, questioning aspects of a model (e.g. 'how can the camera angle be changed?') and trying out theories or ideas (e.g. 'if the camera x-position is incremented then panning can be simulated'). The use of the EMPE can be flexible in nature (as described in §1.2.4 & §1.2.5) because learning need not have a prescribed outcome—it can offer the flexibility of being able to explore a wide range of potential learning opportunities—and learning does not necessarily follow a planned route—it can be guided by the presenter or the listener in response to their questions and experiments. The learning material can incorporate a variety of sources by including new and existing models into the presentation environment, either to add depth an idea or to make it relevant to a particular learner. The use of the EMPE addresses Tufte's suggestions that presentation tools should pay attention to meaningful content [Tuf03] because the EMPE, like good teachers, is focussed on explanation, questioning, and finding things out. As seen in the example, the 3D room model enables meaningful explanations to be found (e.g. of why different projection functions cause certain effects) because they are derived from practical interactions (e.g. 'what happens if the eye distance is negative?').

The EMPE may not be the easy-to-use tool that is expected in PowerPoint style applications because it requires some effort to become a model-builder and to experiment with a model. However, as a support for learning, it offers potential to assist learning that involves experimental, flexible and meaningful characteristics as discussed in Chapter 1. The contribution of this section to the overall thesis is to show that model-building with EM is not only of interest as a personal, individual activity that is peripheral to traditional educational practices—it is possible that practices such as lecturing using presentation tools can benefit from an EM approach.

5.3 Lifelong learning

In the previous section the focus has been on the benefits of EM principles and tools for teaching and lecturing activities. The aim of this section is to illustrate the benefits of an EM approach to lifelong learning†.

The term 'life-long learning' has become prominent within the educational community and in government proposals. In the UK, the Secretary of State for Scotland has declared that "Lifelong learning is a feature of modern life and will continue to be so" [TSO98]. The use of the word 'lifelong' is somewhat difficult to interpret as it can refer to all kinds of learning, encompassing pre-school, school, higher and further education, as well as both formal and informal learning. For the purposes of this thesis, 'lifelong learning' will be taken to mean learning activity that takes place as a part and expression of living. This accords with the popular archetype for life-long learning: adult education outside the schooling years through work (e.g. in training courses) and also for pleasure (e.g. night classes, etc). It also embraces the kind of unsupervised, self-motivated learning that is associated with over-a-lifetime learning of specialist disciplines, hobbies and skills outside the classroom.

5.3.1 Characteristics of lifelong learning

As Piaget discovered in his detailed childhood studies, a child's conception of the world can be very different from an adult's [Pia29]. It follows from this that there may be differences in the way children and adults learn. The word 'andragogy' was originally formulated by Alexander Kapp in 1833 to mean "the art and science of helping adults learn" in order to differentiate it from 'pedagogy' which originates from the study of learning in children [Smi99]. Knowles, one of the most prominent researchers in

†The material in this section is based on a joint paper with Meurig Beynon presented at the International Conference of Advanced Learning Technologies 2006 [BH06], which was subsequently published in an extended form in the Journal of Computing [BH07].

andragogy, defines the following five differentiating characteristics of adult learning [Kno84:p12]:

1. As a person matures, they become more self-directed.
2. Adults have accumulated experiences, which can be a rich resource for learning.
3. Adults become ready to learn when they experience a need to know something.
4. Adults tend to be less subject-centred then children; adults become increasingly problem-centred.
5. For adults, the most potent motivators are internal.

The role that educational technology can play in supporting these characteristics in relation to lifelong learning has yet to be clarified. There are reasons to suppose that current technology is well-suited to supporting independent learning activities on the periphery of established educational frameworks [BEC04], but optimism is tempered by the knowledge that educational technology has yet to live up to its expectations within these frameworks [McP05]. This ambivalence about the potential of technology is reflected by Gulati: "The impact of formal education continues to limit the flexibility and learner choice in online learning, through increased focus on surveillance and compulsory participation ... if we are to realise a lifelong learning society, and to enable self-reflexivity and innovation, then we need to challenge the limiting influences of the dominant educational discourses." [Gul03]. The limitations of current educational technologies are confirmed by recent research findings revealing that greater online interaction does not significantly improve student grades [DG05]. Though there has been a trend towards many more online students and classes, there have also been exceptionally high drop-out rates—up to 80%—for online courses [Car00].

This section argues that adapting the conventional e-learning environment to support lifelong learning is exceptionally challenging because of the mismatch between the characteristics of lifelong learning and the traditional conception of development and use in computing (as described in §1.1.4). In some respects, a more appropriate orientation is found in the computing technologies behind recreational activities such as game-playing, digital photography or electronic music. The principles that are emergent in these technologies have yet to be properly recognised, and are in tension with the established framework of computing that is based on the view of a program as an input-output relation. Stein, from her experiences teaching novice programmers (in CS1), argues that programming should be more concerned with everyday concurrent environments ("computation-as-interaction") and less concerned about input-output relations [Ste99]. Furthermore, the thinking that underlies classical computer science and the input-output relation leads to a duality between development and use, as discussed in elsewhere in this thesis (§3.1). As Stein would argue, an alternative conception of computing is needed in order to take account of interaction in applications such as online chat [Ste99]. In terms of education, a view of computation as interaction is needed *to liberate the constructionist ingredients essential for life-long learning* [BH06]. Building on critiques of conventional programming in support of constructionism [BH05b], model-building based on Empirical Modelling (EM) principles is suggested as an alternative approach that is particularly well-suited to the demands of lifelong learning.

5.3.2 Educational technology for lifelong learning

Technology as a medium for communication is the current driving force behind lifelong learning. There are two aspects to this communication. Computers have become popular for the distribution of information since the birth of the World Wide Web, and are now commonly used as resources of downloadable course material. Developing web resources is perceived as enabling learning outside the classroom, allowing learners access to information in an ubiquitous manner. Computers have also been used for two-way communication in environments where students and teachers can interact. Such communication in support of e-learning can be synchronous, asynchronous or a combination of both. For example, a teacher can communicate with a student by email or organise an online session to instruct many students at the same time. This potentially provides universal access for learners to teachers and virtual classrooms.

Organised learning activity that exploits technology as a communication medium in these ways is not well-matched to the needs of the lifelong learner. Typical e-learning environments are best-suited to supplying the framework for the systematic exposition of a discipline. Such environments perform best where the learner 'begins at the beginning' and follows the prescribed learning paths sufficiently conscientiously to enable the system to build up a useful learner profile at every stage. Ideally, it should be possible for the learner to enter the framework at any point without having to incur a large overhead in supplying the contextual information about their learning status that is required by the system. In practice, any customisation of resources to the learner has to rely heavily upon the previous history of interaction with the learning environment. This is one of the problematic issues for e-learning environments, accounting for the frustration felt by learners who wish to engage with advanced topics, but are first obliged to perform routine exercises in order to inform the system of their status. Friesen, in a paper stating objections to learning objects and e-learning standards [Fri04], criticises the military approach to education that views any learner as just another component to be specified in a 'learning management system'.

In the context of lifelong learning, the casual use of the internet both to acquire information and to use or download interactive 'learning objects' has greater promise as a model for e-learning. Though the web does not necessarily provide the electronic analogue of an accredited teacher or secure classroom, nor the structured framework of a school curriculum, it meets the needs of the independent learner in some respects. The choice of resources offers the opportunity for self-directed learning; material is generally more self-contained and can be accessed and adapted as required; the range of perspectives represented can be rich and wide. These potentially dangerous characteristics are virtues for learners with the appropriate level of discrimination and experience. The limitations of the web as a medium for life-long learning relate primarily to the predominantly passive and unstructured nature of the learner's interaction. In a keynote address at ICALT 2006, Oleg Liber proposes that "personal learning environments" be developed to enable learners to combine structured education with informal resources and communities through the Internet so that learning can be realised as not just "life long" but also "life wide" (i.e. engaging with a wide variety of content, people and activities, particularly outside of the educational system) [Lib06].

Both e-learning environments and the web typically offer relatively limited and closed forms of interaction for the learner. Because so much lifelong learning is self-motivated, a greater degree of autonomy in interaction is desirable. The environment that best suits the lifelong learner is then one that contains elements that are constructionist in spirit [PH91], and gives opportunities for learning by building in an experimental, flexible and meaningful manner. Since lifelong learning also typically takes place in close association with concrete external activities, it is natural to consider using microworlds to provide a virtual environment within which exploratory learning can take place in context.

5.3.3 Model-building for lifelong learning

The concept of lifelong learning clearly invokes an evolution over time, both in respect of the learner's experience and of the context for learning. Such evolution is of course conceived in traditional environments for e-learning, but is typically constrained to follow prescribed paths and preconceived outcomes—as criticised in Chapter 3. In such environments, the learner is exposed to new concepts, experiences and contexts in a systematic fashion, and the exposition is managed in such a way as to keep track of the learner's performance. But whereas the classroom learner's experience is shaped in an artificial closed environment, that of the lifelong learner is not. As Knowles describes, when a person matures they become more self-directed, they draw on accumulated experiences, they become more problem-centred and they develop their own motivations [Kno84:p12]. The lifelong learner frequently combines a sketchy explicit understanding of fundamental principles with a depth of experience and a familiarity with practical contexts of application that seems incongruous and inappropriately advanced.

In these circumstances, the e-learning environment that is designed to suit the learning purpose best under stereotypical conditions is no longer necessarily effective. It may be appropriate to address topics in any order, to make opportunistic, serendipitous links, or to change the strategy mid-process in the light of developments in the open world outside the classroom. Such issues can only be addressed to a limited degree by the preconceived design of an e-learning system. It is hard enough to develop adaptive systems that are selective and discriminative when the learning trajectory has been comprehensively monitored; it is impossible when the learner's engagement is casual and incidental to much broader interaction in the outside world. In the typically informal and unstructured setting of life-long learning, the onus of bridging the gap between standard textbook knowledge and procedures and their often disguised or distorted real-life counterparts then has to fall upon the learner.

Such 'soft' learning needs can be addressed by developing technology to support

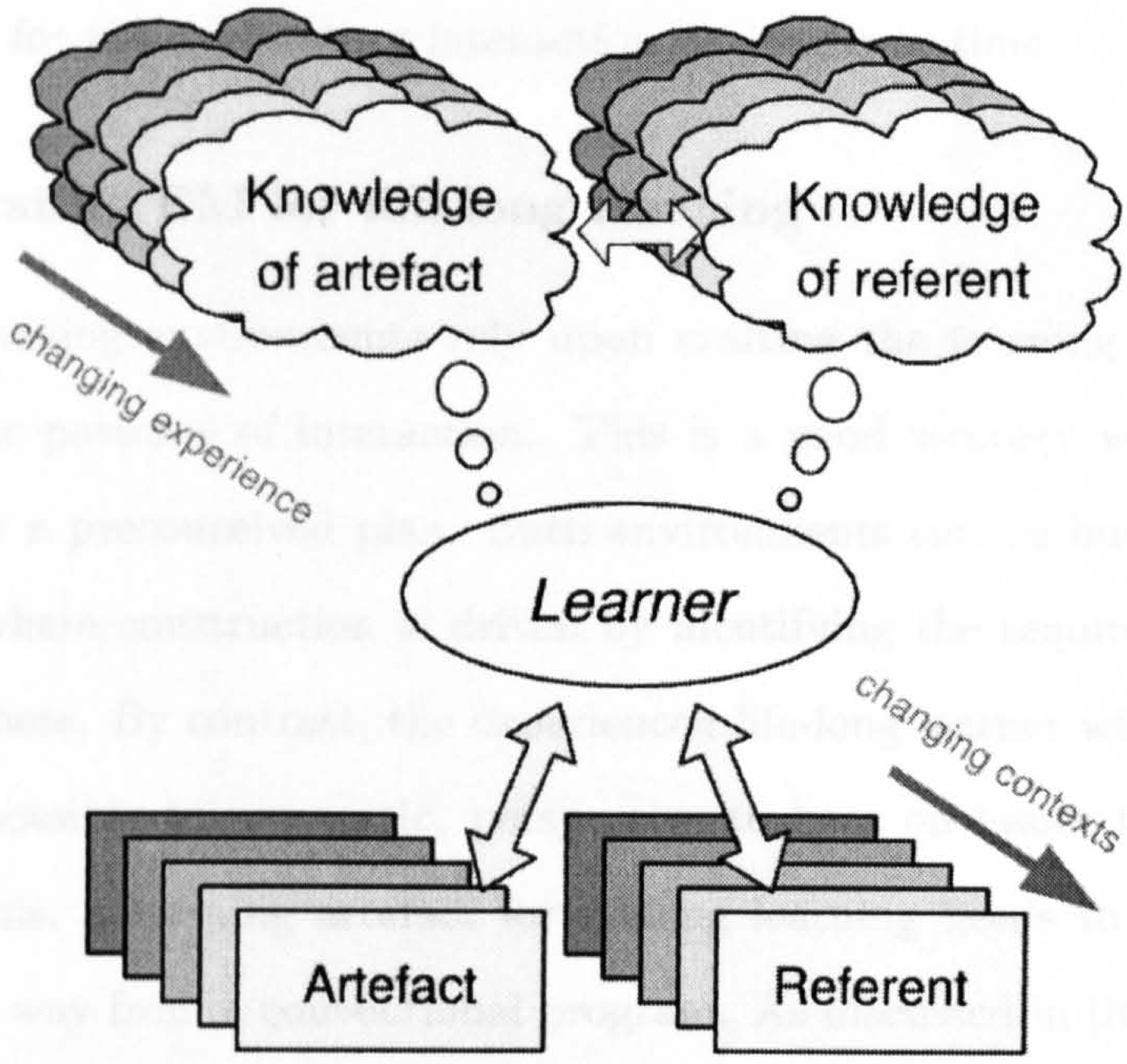


Figure 5.14: Learning through model-building with changing contexts and evolving experience.

the learner in sense-making activities. In a lifelong learning setting, this sense-making can take many forms. It may involve making a model of a situation drawn from the learner's working environment that can be used to gain a deeper understanding of what relationships and mechanisms are at work. Alternatively, it may involve a process of concretisation: constructing a physical artefact to embody an abstract process whose practical relevance and application is obscure. As a prominent component of much lifelong learning is the exposure and rationalisation of activities and concepts of which the learner already has implicit informal knowledge, the construction of models and artefacts cannot in general be based on a pre-existing theory. As in constructionism, the process of building can itself be a process of active learning, through which connections are made and relationships between different experiences come to be better understood.

The nature of the model-building activity that can meet the life-long learner's requirements is depicted in Figure 5.14, and is an elaboration for lifelong learning of the artefact/referent in Figure 2.5 on page 39. The added layering in Figure 5.14 is used to convey the idea that the relationship between the artefact and its referent evolves over time. The context in which the artefact and referent are being experienced is constantly changing, and invokes a change in the implicit knowledge of the artefact. As is to be expected in the lifelong learning setting, both the experience of the learner and the context for the exploratory interaction develop over time.

5.3.4 Illustrating EM for life-long learning

Traditional e-learning environments rely upon crafting the learning context through imposing specific patterns of interaction. This is a good strategy when learning activity can follow a preconceived plan. Such environments can be built by traditional programming, where construction is driven by identifying the required use-cases and optimising for these. By contrast, the experienced life-long learner will typically bring an individual, possibly idiosyncratic, perspective to bear on issues to be learnt. To accommodate this, a learning artefact for lifelong learning needs to be conceived in quite a different way from a conventional program. As discussed in [Run02], EM is an alternative approach to computer-based model-building that suits the vision of learning depicted in Figure 5.14. An extended example will serve to illustrate the principles of EM in relation to lifelong learning. The theme of this example—that of learning about time and clocks—is too simple to be fully representative of the applications of EM to lifelong learning, but highlights many of the essential characteristics, as discussed in §5.3.1.

As remarked above, the sense-making activity depicted in Figure 5.14 can reflect many different kinds of learning. Relevant topics might relate for instance to: being familiar with clock mechanisms; understanding the relationship between digital and analogue representations of time; appreciating how the analogue clock concretises abstract relationships in modulo arithmetic; or knowing how to tell the time in different languages and in different time zones. In a life-long learning context, each learner will bring a different orientation and experience to these diverse perspectives on clocks and time. The process of construal that EM supports reflects this rich and potentially confusing combination of concerns. The various perspectives and their interrelationships are reflected in variants of what can be regarded as one model, as developed in different directions according to the learner's particular needs. An important feature of this model is that in principle it concurrently offers the same potential for redefinition and adaptation to all participants in the learning—whether model-builder, teacher or learner. In this way, it can serve in many different educational roles as a learning artefact—some aspects being developed autonomously by the learner, some supplied

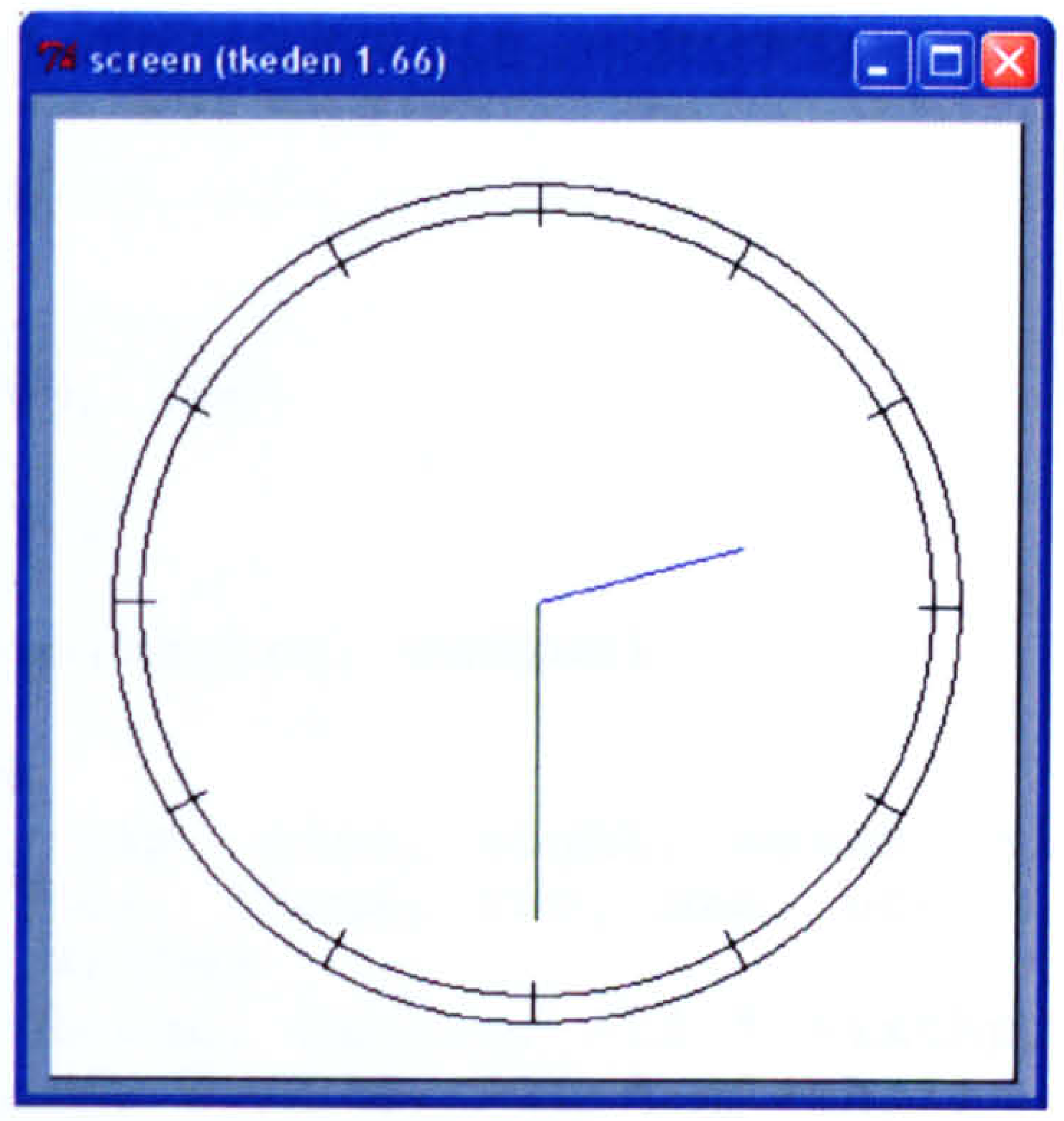


Figure 5.15: A screenshot of the basic clock model [EMP:clockHarfield2006].

by an expert modeller, and some adapted and customised by a teacher.

The simplest form of sense-making model takes an analogue clock as its referent as shown in Figure 5.15. The relevant observables in this context include the current local time, the time as shown on the clock, the radius of the clock face and locations of the marks around its rim, and the lengths, colours and positions of the hands, as shown in Figure 5.16. Relevant dependencies link the position of the hands to the time as shown on the clock (e.g. in Figure 5.16, `hourAngle` determines the angle of the hour hand and `hourHand` determines the line representing the hour hand), which in turn may depend on the current time in a variety of ways according to the status of the clock. For instance, the clock may be fast or slow, refer to a distant time zone, or take account of daylight saving, as shown in Figure 5.17. To reflect the physical integrity of the clock, the positions of the marks on the rim and the lengths of the hands depend upon the radius of the clock face (e.g. the definition for `noon` in Figure 5.16). In developing the EM construal, the geometric elements of a line-drawing to depict the clock can be specified as points, lines and circles whose attributes are linked by definitions to scalars and textual data that represent times, dimensions and other geometric attributes such as colours and line styles. The `clock/t` observable in the DoNaLD script (Figure 5.16) is referred to in EDEN as `_clock_t` as shown in Figure 5.17 redefined for different time zones. Full details on the scripts that I have developed for this model can be found in the EM archive [EMP:clockHarfield2006], as well as notes on how this clock model differs from the original clock model by Beynon [EMP:clockBeynon2001].

The merits of the EM construal as a learning artefact relate to the open-ended

```
%donald
openshape clock
within clock {
   point centre
   centre = {200, 200}
   real radius
   radius = 150.0
   circle edge
   edge = circle(centre, radius)

   real sixthpi
   line eleven, ten, nine, eight, seven, six
   line five, four, three, two, one, noon
   sixthpi = 0.523599
   eleven = rot(noon, centre, -11 * sixthpi)
   ten = rot(noon, centre, -10 * sixthpi)
   nine = rot(noon, centre, -9 * sixthpi)
   eight = rot(noon, centre, -8 * sixthpi)
   seven = rot(noon, centre, -7 * sixthpi)
   six = rot(noon, centre, -6 * sixthpi)
   five = rot(noon, centre, -5 * sixthpi)
   four = rot(noon, centre, -4 * sixthpi)
   three = rot(noon, centre, -3 * sixthpi)
   two = rot(noon, centre, -2 * sixthpi)
   one = rot(noon, centre, -sixthpi)
   noon = [centre+{0,0.9*radius}, centre+{0,radius}]

   int t  # representing time elapsed in minutes
          # from midnight (i.e. 1440 per day)
   int minute, hour
   minute, hour = t mod 60, t div 60

   real minAngle, hourAngle
   minAngle = (pi div 2.0) - (minute * pi div 30.0)
   hourAngle = (pi div 2.0) - (hour * pi div 6.0)

   line minHand, hourHand
   minHand = [centre + {0.75*radius @ minAngle}, centre]
   hourHand = [centre + {0.5*radius @ hourAngle}, centre]
}
```

Figure 5.16: A DoNaLD script describing the clock face.

```
%eden

/* Making the clock 5 minutes slow */
_clock_t is greenwichmeantime - 5;

/* Introducing a new time zone (9 hours ahead) */
japanesestandardtime is greenwichmeantime + (60*9);
_clock_t is japanesestandardtime;

/* Adjusting a time zone for daylight saving */
daylightsaving is 1;
mexicotime is greenwichmeantime - (60*6) + (daylightsaving?60:0);
_clock_t is mexicotime;
```

Figure 5.17: An EDEN script containing modifications to the clock model for displaying the time in different time zones and with daylight saving time.

interactions that it enables. Though the clock exhibits standard modes of interaction and behaviour, its observables, dependencies and the agency to which it is subject are all open to revision at the discretion of the learner—whether or not they respect the boundaries of common-sense. This is in keeping with the principle that the engineer learns most not just by observing the clock in normal operation, but by dismantling and rebuilding it, and the user learns most by interacting with the clock in exceptional contexts and exploratory ways.

By way of simple illustration, in the clock model as described above, the positions of the hour and minute hands are independently determined by the current time (e.g. in Figure 5.15, the minute hand is showing half past the hour but the hour hand is not half way between two and three). In practice, the hands of a mechanical clock are linked so that you can move the minute hand and the hour hand moves at a slower (but proportional) rate. The clock artefact can be adapted to exhibit this behaviour by introducing the dependency that links the position of the hour hand to that of the minute hand as depicted in Figure 5.18. Underlying the design of the analogue clock as an engineering product are simple principles to connect elapsed time in hours and minutes modulo the number of minutes in an hour and hours in a day. The abstract relationships between 'time as recorded by hours and minutes elapsed in a day' and 'time as displayed on digital and analogue clocks' are given concrete expression in the variant of the original clock model shown in Figure 5.18. This variant is derived simply by giving visual expression to scalar relationships that are already explicit in the original clock model.

By way of further illustration, the time as shown on the clock can be redefined in such a way as to be totally independent of the current local time, or so as to reflect the time in another time zone (as shown in Figure 5.17). The significance of specifying the time difference between Japan and UK time as plus 9 hours, rather than minus 15 hours, or even plus 5 hours, exposes the physically constrained and socially constructed nature of world time. The focus on clocks and time in physical and cultural context is well-oriented to life-long learning, where contextual factors potentially both enrich and obstruct understanding. For instance, it is indicative of the imperfect and potentially confusing nature of learning to read clocks in a real-world setting that (e.g.) the hour hand may be misaligned so that it is not quite vertical at midday. It is easy to tweak

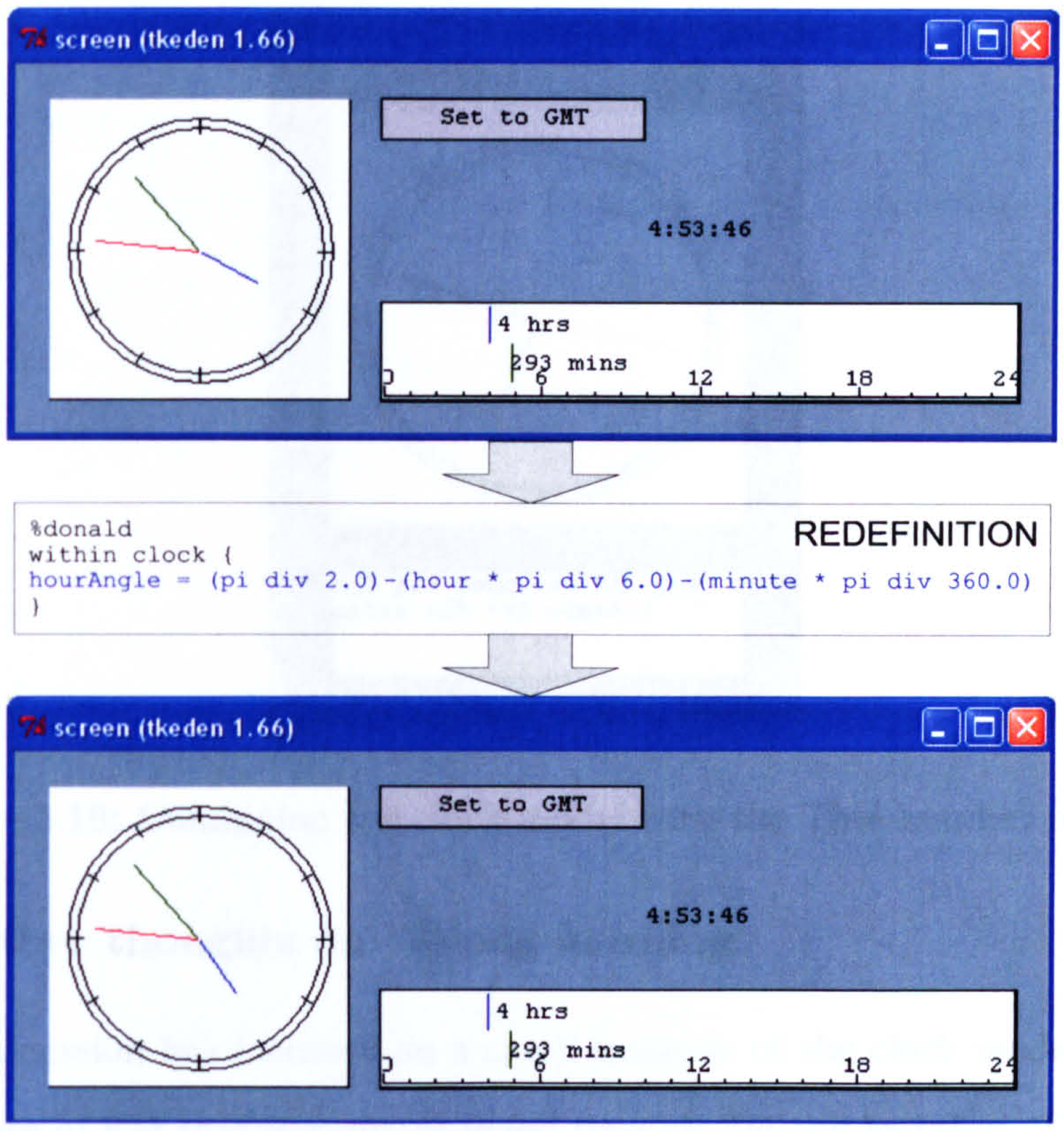


Figure 5.18: A redefinition to the clock model to make the hour hand dependent on the minute hand as would be found in a mechanical analogue clock.

definitions to imitate this condition, or to express more dire forms of mechanical failure, as when the minute hand comes loose and hangs vertically downwards.

The EM construals described above illustrate how Figure 5.14 applies both to the modelling of a concrete referent, and to the concretisation of abstract relationships. Because of the dynamic and provisional nature of the relation between artefact and referent in Figure 5.14, it is also possible to regard it as a framework within which two or more artefacts can be combined and can evolve into a new learning artefact. Previously, an EM artefact for learning about counting in different languages was built by Rungrattanubol [Run02] as introduced in §5.1.2. By placing this artefact in conjunction with the clock artefact, and adding new observables and dependencies it is relatively easy to derive an artefact for telling the time in different languages (as well as different time zones). Figure 5.19 depicts the artefact displaying the time in Japan whilst expressing the time in Thai.

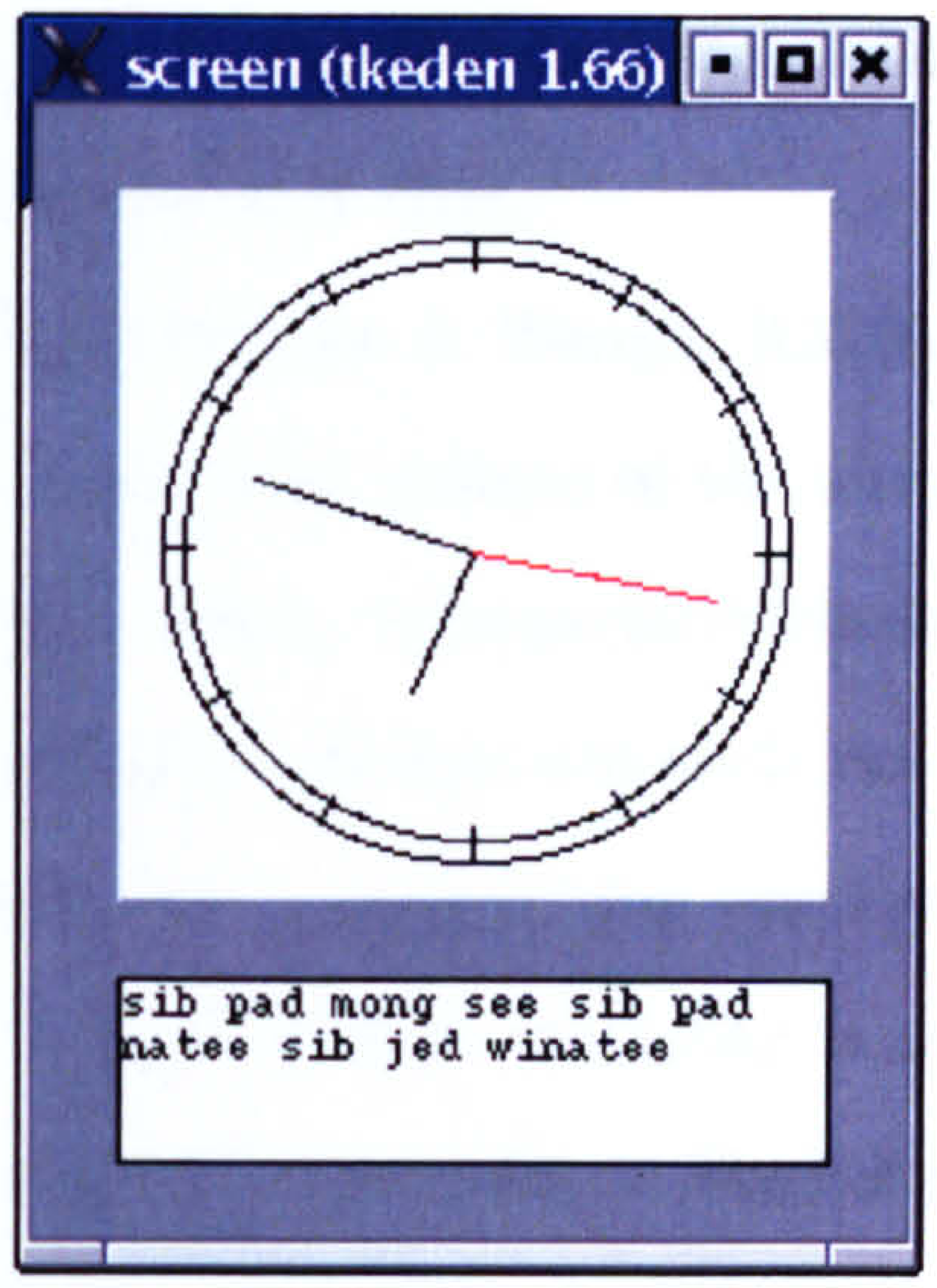


Figure 5.19: Combining the clock model with the Thai number model.

5.3.5 Further thoughts on lifelong learning

The above discussion has focussed on a small example of the clock model to illustrate several features of EM that are well-oriented towards lifelong learning. The purpose of this exercise is to demonstrate the wider applications of EM principles to learning beyond school and higher education. Lifelong learning is a particular area where EM has significant advantages over conventional e-learning environments because EM allows learners to take account of something of the 'superabundant' nature of our experience (to paraphrase William James [Jam12]). Where current educational technologies are best oriented for well-planned and organised learning situations, learning in a real-world setting typically begins in some degree of chaos and confusion. EM principles and tools, although still at an early stage of development, show promise in supporting learning activities that integrate educational roles (as discussed in §3.3), promote flexible opportunistic learning, and blend the concrete and the abstract across disciplines. These are qualities that can be most helpful in engaging the lifelong learner.

5.4 Collaborative learning

The focus of this thesis has mainly been on individual model-building and personal learning, as is evident from the previous section on lifelong learning. Although student, teacher and developer perspectives are examined in terms of model-building, the majority of the discussion has been concerned with one model and one learner at any one time. The aim of this section is to explore the potential for EM to be used collaboratively with more than one learner at a time.

As discussed in section §1.2.7, Lave & Wenger [LW91] suggest that all learning is embedded in the activity, context and culture of the situation. Social interaction is a critical component of situated learning because individuals become involved in communities of practice where shared ideas, beliefs and skills are acquired [Wen99]. Similarly, according to Gerhard Fischer, the power of the unaided individual human mind is highly overrated [Fis05]. Much of human creativity is social, either arising in social situations or as the product of social interaction. Fischer points out that innovation (a leap in learning) usually emerges from "joint thinking, passionate conversations, and shared struggles among different people" [Fis05]. It therefore seems crucial that the social aspect of model-building for learning is examined from an EM perspective.

A large area of research into educational technology is concerned with collaborative learning. This mostly falls within research associated with the term 'computer-supported collaborative learning' (CSCL). The EM approach departs in some respects from the common application of CSCL as will be highlighted in the following sections.

5.4.1 Previous research on EM and collaboration

There are a few projects in EM that have touched upon the topic of collaborative learning, but there are no in-depth accounts of collaborative learning using EM. However, there are many more projects which have involved the construction of distributed models which may be relevant to collaboration (e.g. Clayton Tunnel rail accident [EMP:claytontunnelChanHarfield2005], 5-a-side football [EMP:footballTurner2000], planimeters [EMP:planimeterCare2005], car parking simulator [EMP:carparkingsimMcHale2003]).

The most significant contribution to collaborative learning with EM tools to date is by D'Ornellas [DOr98] and Sheth [She98]. In their project on group learning, they construct a model for exploring electronic circuits and examine its potential for facilitating interaction in the classroom. The electronic circuit simulator is a distributed model, with the teacher controlling the server and a student interacting at each connected client. Four means of interaction are discussed: students interacting independently on their own client; students sharing parts of their model with the teacher or rest of the group; students interacting solely on global model; and the teacher leading the interaction or moderating the students' interaction.

The wide range of activities that D'Ornellas and Sheth were able explore with their electronic circuit model using EM is due to the flexibility of the EM tools [DOr98]. Taking one student's work and linking it to another student's work could well be a major operation if the implementation was undertaken with procedural programming tools and would almost definitely require recompilation and a restart of the environment. The advantage of EM is that collaborations with other model-builders or connections to other models can be treated just as any other interaction. Model-builders have essentially the same mechanism for state-change (in the same way that there is no difference in the way students, teachers and developers interact with a model—see §3.3). Furthermore, the use of dependency is a powerful tool for connecting models in that aspects of one model (on one computer) can be dependent on certain features of another model (on another computer). The benefits of dependency can be compared to a spreadsheet where a cell is dependent on a cell in another spreadsheet on another computer, or on some piece of information on the Internet†. As D'Ornellas and Sheth suggest [DOr98], EM's potential for learning is enhanced by dependency that enables models to be connected together to create new models in a distributed and collaborative manner.

5.4.2 Three types of collaborative learning with EM

Before exploring any particular examples of collaborative EM, it is worth reflecting on what collaboration may entail. A wide variety of activities can be considered collaborative even in terms of model-building. The following three types of collaboration (inspired by Resnick's categorisation of distributed constructionist activities [Res96]) explain the nature of collaborative activities that are relevant to EM:

1. discussing models;

2. sharing models;

3. building models.

Discussing models is the simplest level of collaboration, involving communication between two or more people relating to a specific model. Examples might include a

†Some online spreadsheets, such as EditGrid (`www.editgrid.com`), feature limited forms of dependency that pull data from remote sources on the Internet [Gib06].

teacher discussing the TLJ model with a student, students discussing the computer graphics 3D room model amongst themselves, or an online discussion of the jugs model in a student forum. The extra computing technology that may be required for this type of collaboration ranges from nothing to instant messaging, voice calls, and video-conferencing. In terms of EM, models could be discussed without any special tools, just as students undertaking an EM project discuss their models with each other. This type of collaboration is already occurring frequently.

On the next level of collaboration, instead of communicating thoughts about models, the models themselves are communicated by sharing parts or all of a model. This can occur informally, such as between teacher and student or between students themselves, like when giving the students a small script to achieve a particular function. Otherwise there may be a formal way of sharing models, such as the EM projects archive [EMP], where students can download other models, to explore or reuse, and also upload their own models for others to use. Model reuse was discussed in relation to the computer graphics presentation in Chapter 4. At the most advanced stage of sharing models, it is possible that two or more model-builders can collaboratively develop a model by sharing it backwards and forwards, constructing the model asynchronously. The computing technology required for this type of collaboration is email, network file-sharing, or file version control (such as Subversion [CFP04]). Once again, collaboration in this manner has been occurring in EM through the projects archive and through model-builders sharing models (or often snippets of models) with each other. This type of collaboration is encouraged among model-builders but so far the only formal place to share models is the EM project archive.

The third level of collaborative learning involves multiple learners constructing a distributed model at the same time. This is the area which is the most complex, and relatively unexplored. Computer technology for supporting this type of interaction is generally quite specialised—a collaborative whiteboard as in Microsoft NetMeeting [Mic04] used by many participants to construct a drawing is a simple but specialised example. Other tools such as the 3C platform [CKW05] support 'desktop sharing' for collaborative construction in any application. This type of collaboration using EM has been explored by D'Ornellas [DOr98] and Sheth [She98] as a potential technology for learning. Even though the EM tools have been used to create distributed models, there

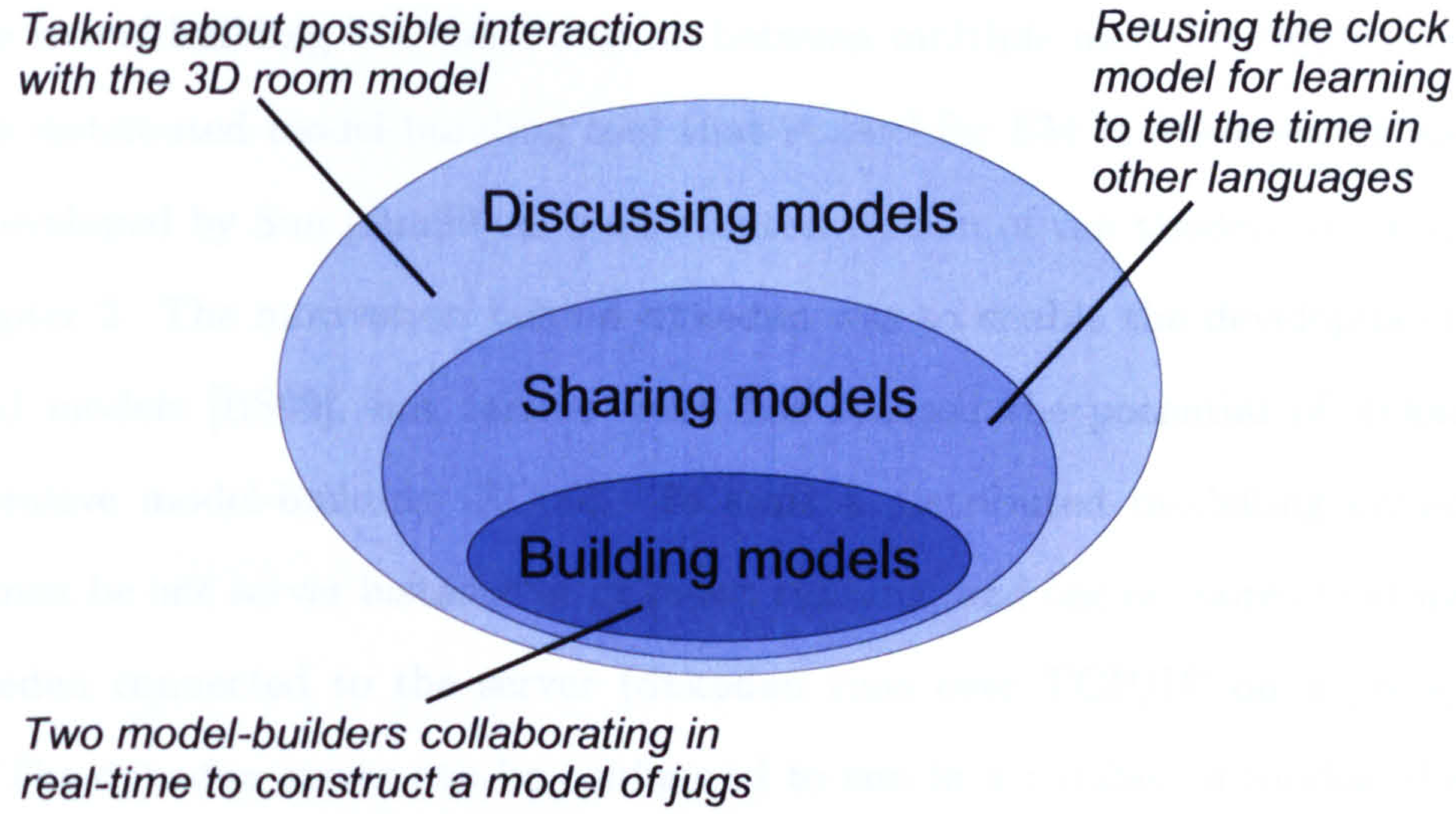


Figure 5.20: Three types of collaborative learning activity.

is little evidence that any collaborative model-building has taken place. The rest of this section will be devoted to explaining and comparing EM technologies that can support such collaborative learning.

Each type of collaboration is a specialisation of the previous levels as depicted in Figure 5.20, in that the basic type *discussing models* is also used in all levels of collaboration. Therefore *building models* can involve *sharing models* and *discussing models*. Tools such as NetMeeting or MSN Messenger support specialised third level activities (e.g. a collaborative whiteboard), backed up with first and second level discussion and sharing tools (e.g. chat and file sharing). In the following sections I discuss how EM tools can be adapted for *building models* collaboratively, and in the spirit of EM, it should be possible to build further models for sharing and discussing models, leading to an integrated environment that could be used for collaborative learning (as well as collaborative software construction as envisaged by Chan [BC06]).

5.4.3 A simple example of collaborative model-building

The jugs model is introduced in Chapter 3 as an artefact for learning about the greatest common divisor in mathematics. A simple example of building a distributed jugs model serves the purpose of illustrating the principles of collaborative model-building with EM in order to demonstrate the potential for collaborative learning. This may be contrasted with the lone model-builder version of the jugs model. There are two new aspects to the jugs model discussed in this section: the model is distributed across multiple machines;

and the model-building is a collaboration between multiple model-builders.

The distributed model-building tool that is used for EM is dtkeden. It was originally developed by Sun [Sun99] as a distributed version of the tkeden tool introduced in Chapter 2. The motivation behind dtkeden was to enable the development of distributed models [BS99], but further work has realised the potential of dtkeden for collaborative model-building [BC06]. To start a distributed modelling environment there must be one server instance of dtkeden running, and one or more client instances of dtkeden connected to the server (dtkeden runs over TCP/IP on a pre-specified port). The dtkeden server can be configured to run in a number of modes (described in the dtkeden manual [SC98]), depending on the intended collaborative activity. The default mode (also known as 'Normal' mode) gives each client their own personal model-building space and allows the model-builder to define observables within the model which can be shared (using the %lsd notation [SC98]). A special mode, developed as part of this thesis[†] together with Chan whose interests lie in collaborative software construction [BC06], enables all model-building activity to take place on a single shared model (cf. Newell's blackboard metaphor [New69]). This 'Blackboard' mode enables model-builders to interact in the same way that multiple people gathered around a blackboard can all contribute to a single space by adding, changing and erasing various items.

In order to illustrate the differences between the 'Normal' and 'Blackboard' modes of model-building, two mini experiments are described and evaluated in terms of their potential for collaborative learning.

5.4.3.1 Normal mode for working in local and global contexts

The first experiment in collaborative model-building was performed during the 'Introduction to EM' module (discussed further in Chapter 6) when one laboratory session was dedicated to distributed and collaborative modelling using dtkeden. Previous lab sessions had introduced the jugs model [EMP:jugsBeynon1988] as an educational artefact designed to introduce pupils to the idea of the 'highest common factor of m and n' by allowing them to fill, empty and pour between jugs of integral capacities m and n (as discussed in §3.2). While the original jugs was designed for one pupil at a time,

[†] Available in a special version of dtkeden from www.dcs.warwick.ac.uk/~ant/dtkeden/.

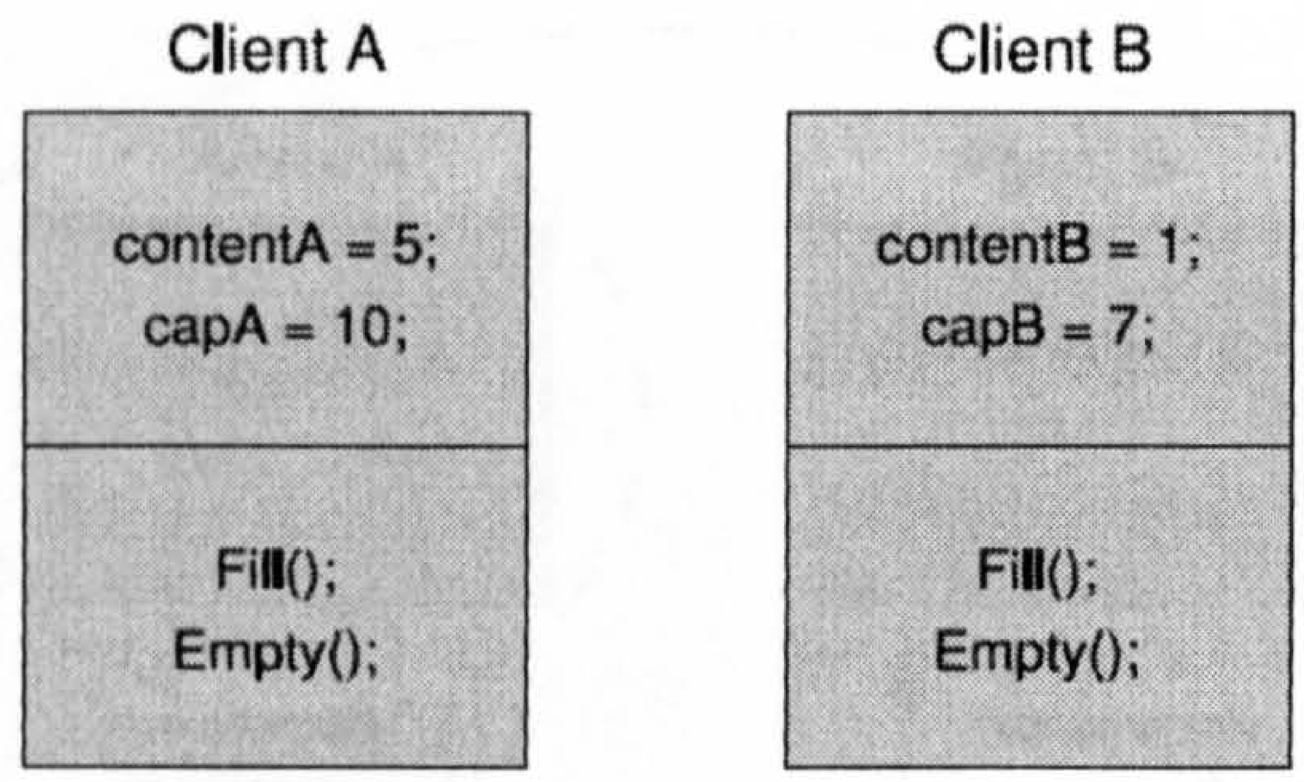


Figure 5.21: Executing definitions in a local context only.

this lab considered a distributed variant of the jugs model in which each pupil owns one jug, and pouring involves transferring liquid from one pupil's jug to another's.

Working in pairs or threes on neighbouring workstations, each group started a `dtkeden` server and then each person in the group connected a `dtkeden` client on their workstation to the server. The groups were then instructed to begin their model-building by constructing a single jug for themselves within their own local context (i.e. not sharing the observables). After completing this task, each client would have a set of observables and procedures that were only available to themselves, as shown if Figure 5.21. In reaching this state there was a little discussion (level 1 collaboration) about the jug and appropriate dimensions for the jug, and some scripts were shared (level 2 collaboration) to enable others to construct a jug quickly.

In the second phase of the lab, the students were introduced to the `%lsd` notation (as described in the `dtkeden` manual [SC98]) for defining *oracles* and *handles* to specify what observables a client is privileged to observe and change in the global context. The global context is the collaborative modelling environment where clients communicate and share observables as shown in Figure 5.22. A *handle* is an observable in the global context which a client can change. An *oracle* is an observable in the global context which a client can respond to (or is aware of). The students were first instructed to add oracles and handles so that each person was aware of their collaborator's jug capacity and content. Then they were asked to add oracles and handles so that they could pour from their jug into another jug. Some students attempted to implement the full repertoire of interactions required for achieving a specified target collaboratively.

As an outcome of this experiment, it was demonstrated that it is possible to use `dtkeden` to build models collaboratively. Also important is that EM activities can be performed in a collaborative environment. Furthermore, there is evidence that learning

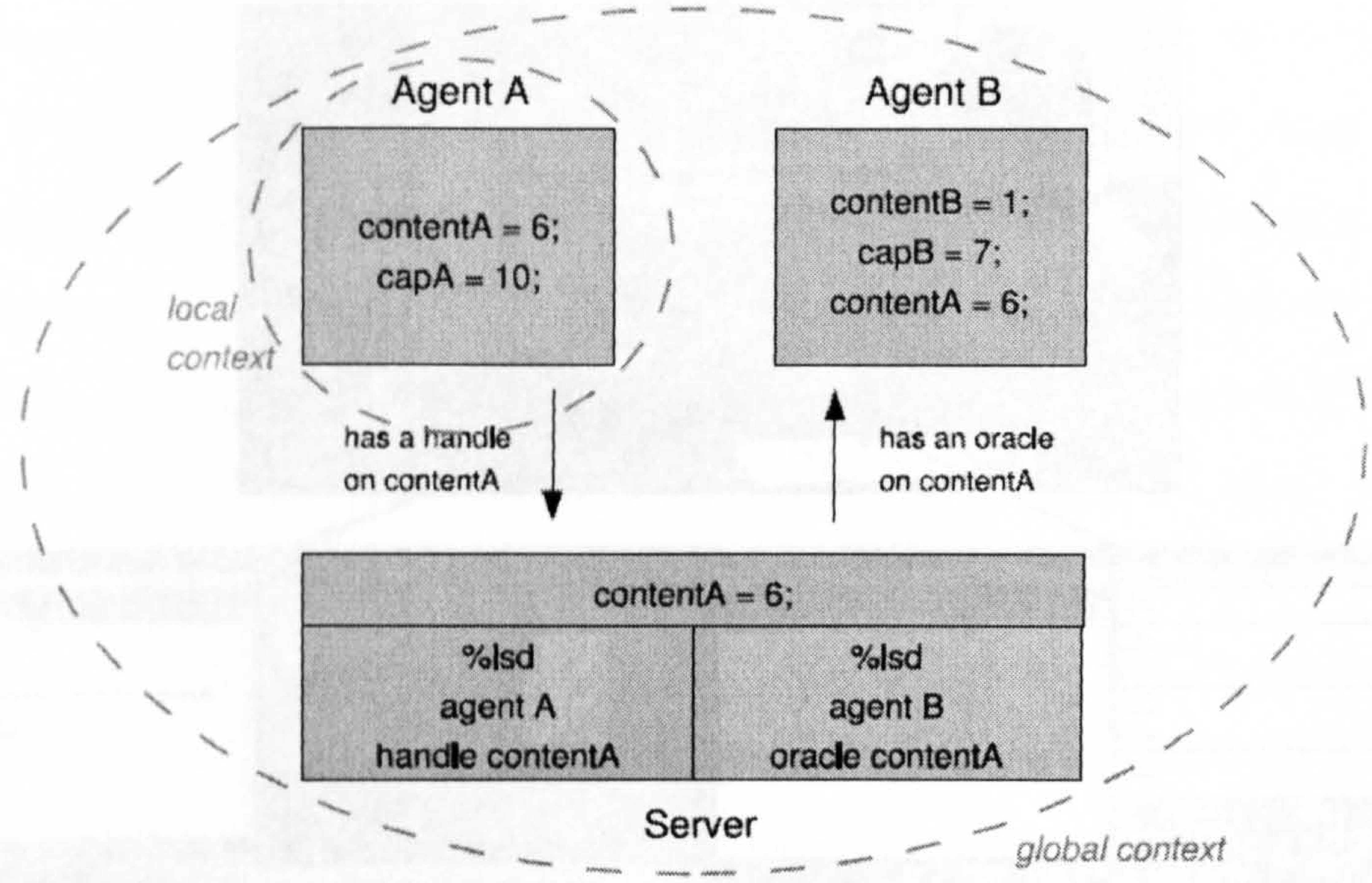


Figure 5.22: Executing definitions in a global context from client A.

by model-building with EM is possible in a collaborative environment. This shows some potential for learning activities with experimental, flexible and meaningful characteristics to be undertaken not just by solo model-builders but also through collaborative model-building.

The problems with the methods for collaborative model-building in this first example is that they require a significant knowledge of how to use the tools collaboratively. This may prove difficult for learners if they need to be too concerned with the technical issues. Even if model-builders grasp the concepts of global and local contexts, they must also spend a lot of time organising which observables should be global and which should be personal (or local) to each model-builder. These technical problems stem from the fact that `dtkeden` was designed for distributed models, and not collaborative model-building.

5.4.3.2 Blackboard mode for working in a simplified global context only

The use of local and global contexts (using the `%lsd` notation) can involve a significant effort by the model-builders, and they must coordinate their efforts with each other in order that effective communication of observables takes place. It is possible that better tools may decrease the cognitive load that is put on the model-builders having to consider local and global contexts, but these have yet to be developed. In response to the difficulties of the above type of collaboration, a simplified mode was designed by

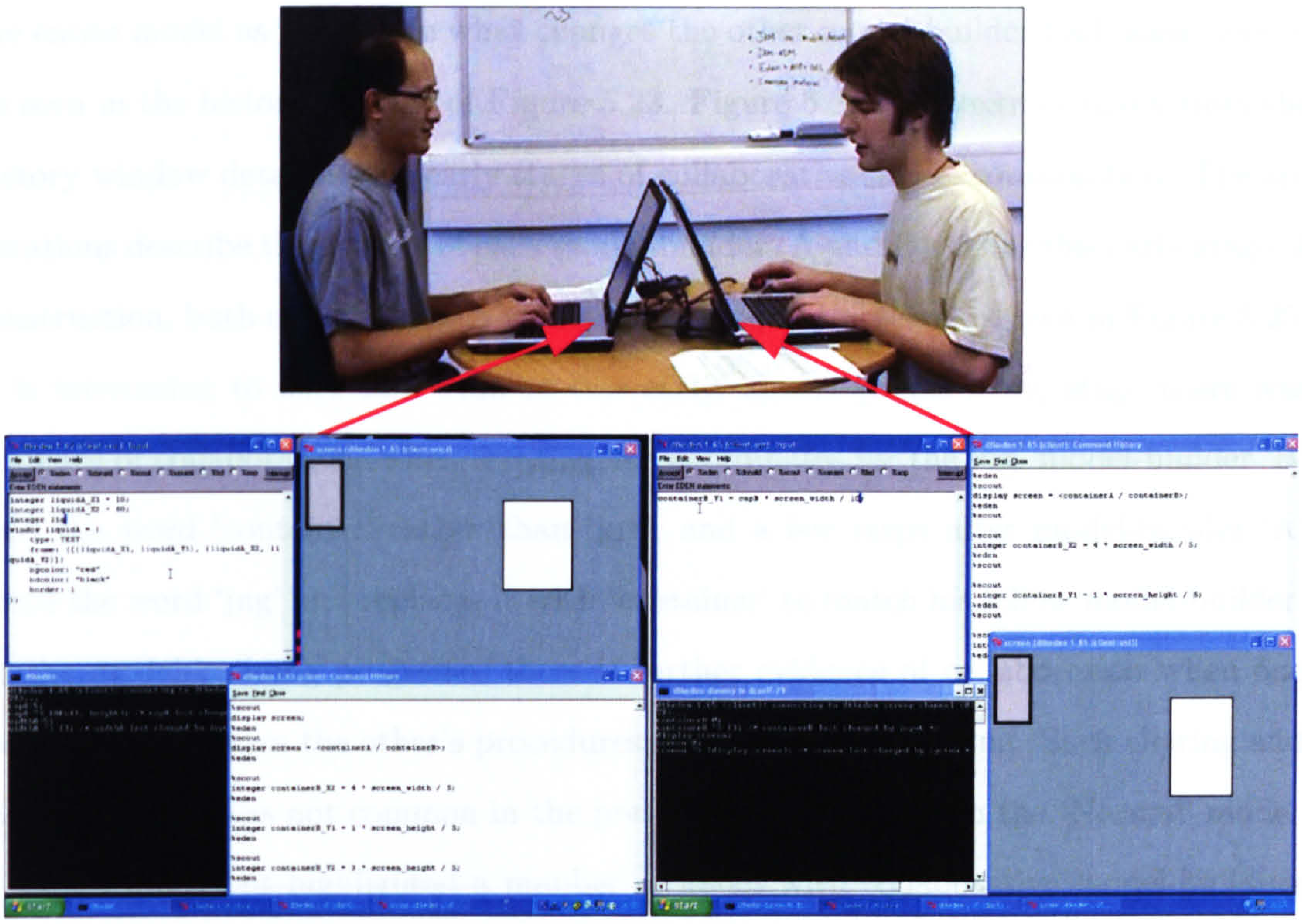


Figure 5.23: Chan and Harfield collaboratively building a jugs model.

Chan and myself to allow for the situation where the model-builders wish to share all of their observables and therefore are not concerned with local and global contexts. Such a situation is metaphorically similar to the collaborative use of a blackboard where each person has their own chalk (and board rubber) and can freely draw on (or erase) the global workspace. The motivation for developing the simplified mode was because of the inspiration that collaborative activities in real life are performed in a global space that everyone has access to, and such activities can either be constructive (i.e. like the chalk) or destructive (i.e. like the rubber) towards the overall creation. The disadvantage of the blackboard metaphor for model-building is the possibility of interference between model-builders in the global space. To test the potential of this simplified mechanism for collaboration, a second mini experiment was performed whereby Chan and myself undertook a collaborative exercise involving the construction of a jugs model in a similar manner to the laboratory session discussed above.

Figure 5.23 shows the two model-builders constructing a jugs model (taken from the video footage of the entire experiment). Starting completely from scratch both model-builders created a standard jug or container on the screen with basic procedures for filling and emptying. Throughout the activity both model-builders could observe the entire model as well as see what changes the other model-builder had made, as can be seen in the history window of Figure 5.23. Figure 5.24 is an extract taken from the history window detailing the early stages of collaborative model construction. The annotations describe the actions of each model-builder (A and B). After this early stage of construction, both model-builders would be looking at the screen shown in Figure 5.25. It is interesting to note that even at this early, almost preparatory, stage there was evidence of collaboration: when creating the coordinates for the jug, model-builder 'B' used the word 'container' rather than 'jug', and a few steps after model-builder 'A' drops the word 'jug' and replaces it with 'container' to match his fellow model-builder. As the model-building progresses there is further evidence of collaboration when one model-builder reuses the other's procedures for filling and emptying. Such sharing and communication was not common in the previous experiment with the 'Normal' mode.

The experiment highlighted a number of issues with collaborative model-building using `dtkeden` in the 'Blackboard' mode. Firstly, working in a global context creates an added concern about conflicting observable names and other types of interference between model-builders. Secondly, as there is no local context in which to work, it is not possible to test a part of a model in the individual's personal space before making it available to other model-builders. Despite these two concerns, building the model is much more of a fluid activity—as described in Chapter 2—because there is little need to stop to plan the building activity. Unlike the first experiment when a model-builder has to work blind to another's model and therefore much discussion was needed, the blackboard metaphor enables all model-builders to see each others work. As all aspects of the entire model are open to negotiation through redefinition, connections in the form of dependencies can be quickly created between each model-builder's work. Although the 'Normal' mode has the advantage of a private local space that is useful for testing, in the spirit of experimental model-building as encouraged in EM, it can be useful to try things out in the global space for all to see. In fact, it opens up the possibility for other model-builders to offer suggestions as was found during the experiment. The benefit of not having to deal with observables in two different contexts is that the collaborative model-building flows more naturally and encourages experimentation.

One technical contribution of this work is that Chan and I have invented a new mode for `dtkeden` that is especially designed for collaborative model-building. As model-

```
/* Begin */
integer jugA_X1 = 10;
>>
integer containerB_X1 = 300;
>>
integer containerB_X2 = 400;
>>
integer containerB_Y1 = 100;
>>
integer containerB_Y2 = 400;
>>
integer jugA_X2 = 60;
integer jugA_Y1 = 10;
integer jugA_Y2 = 110;
>>
window containerB = {
   type: TEXT
   frame: ([{containerB_X1,containerB_Y1},{containerB_X2,containerB_Y2}])
   border: 2
   bgcolour: "white"
   bdcolour: "black"
};
>>
screen = <containerB>;
>>
window containerA = {
    type: TEXT
    frame: ([{jugA_X1, jugA_Y1}, {jugA_X2, jugA_Y2}])
    bgcolor: "grey"
    bdcolor: "black"
    border: 5
};
>>
containerB_X1 = 3 * screen_width / 5;
>>
display screen = < containerA / containerB >;
>>
containerB_X2 = 4 * screen_width / 5;
>>
containerB_Y1 = 1 * screen_height / 5;
>>
containerB_Y2 = 3 * screen_height / 5;
```

>> marks end of input

A's first input.

B's first few inputs set up the coordinates of a jug.

And then A follows suit.

B creates the SCOUT window.

B adds his window to the screen.

A catches up with a window for his jug.

B redefines his jug coordinates.

A redefines the screen to take account of both jugs.

B makes further changes to his jug so that it is dependent on the size of the screen.

Figure 5.24: The script resulting from initial stages of collaboration by two model-builders.

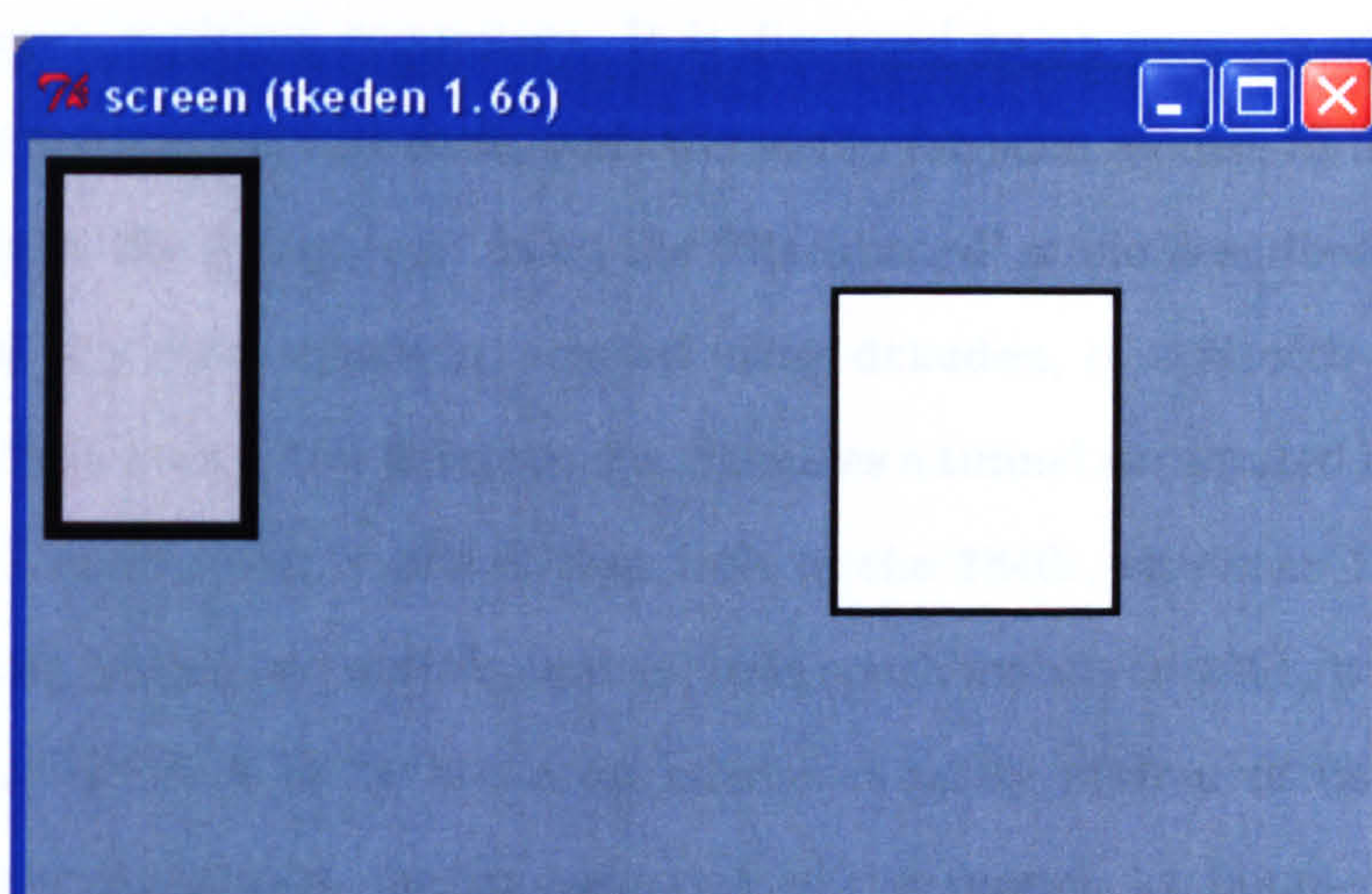


Figure 5.25: A screenshot of the jugs model at an early stage of collaborative model-building.

building with EM can be beneficial for a conception of learning that can be characterised as experimental, flexible, and meaningful, it is suggested that such learning can also be achieved with a collaborative element. This section has demonstrated that collaborative model-building for learning is possible. The next section explores a specific example of a historical rail accident that highlights how learning in a collaborative EM environment offers benefits over other forms of CSCL.

5.4.4 Learning about the Clayton Tunnel accident

The jugs model serves as a simple illustration of the ideas behind collaborative model-building in EM using dtkeden. In order to show potential for collaborative learning, it must be shown that model-building collaboratively can be extended to more interesting situations where there is greater scope for the development of understanding. The following example is based on a larger model, that has been developed by a number of model-builders, over a period of a number of years. The Clayton Tunnel model was an exercise in building a distributed model to reconstruct a famous accident in 1861 [Sun99]. Subsequent work on the model by Chan and myself has demonstrated the potential for the model in learning about the history of the Clayton Tunnel accident†.

5.4.4.1 Background to the model

The Clayton Tunnel model was not constructed in a collaborative manner (i.e. built by multiple model-builders collaborating together at the same time), but the model is distributed across multiple computers. It is discussed as an example of what can be achieved with the current EM tools, with the vision for such models to be constructed collaboratively in the future (e.g. using the 'Blackboard' mode described in §5.4.3.2).

The model is a reconstruction, created using dtkeden, of a stretch of railroad in England between London and Brighton that includes a tunnel as pictured in Figure 5.26. The Clayton Tunnel, as it is called, was built in the 1840s, stretches 1.25 miles and contains two rail tracks, one northbound and one southbound. In 1861, it was equipped with what was believed to be the most advanced safety system of its day. It was operated by two signalmen, one at each end of the tunnel, 24 hours per day. The

†This section draws on material from a poster presented at Kaleidoscope 2005 and used subsequently for EM demonstration purposes [HCW05]. The Clayton Tunnel model discussed in this section is the Chan and Harfield version [EMP:claytontunnelChanHarfield2005].

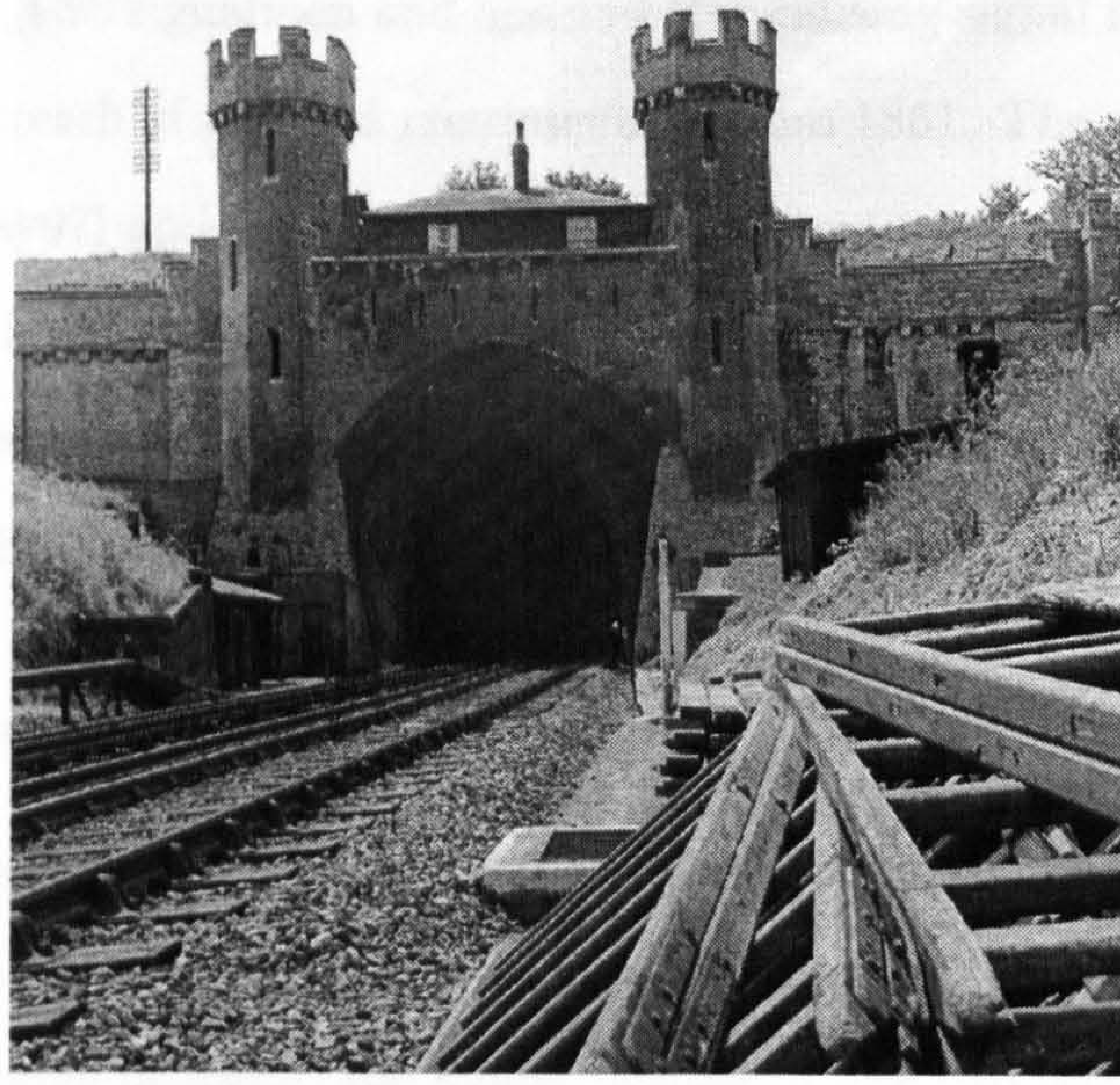


Figure 5.26: A photograph of the Clayton Tunnel (reproduced by permission of English Heritage.NMR).

tunnel used a space-interval system whereby only one train was allowed to occupy the tunnel (in any one direction) at a time. The signalmen guarded the tunnel using a combination of telegraphs, warning signals and traditional semaphores.

In order to appreciate the causes of the accident, it is necessary to explain the protocol for safe operation of the tunnel. Between the signalmen at each end of tunnel, a needle telegraph was used to communicate the status of the tunnel. When a train entered the tunnel, an 'occupied' message was sent, and when a train exited at the other end then a 'clear' message was returned. Communication between the signalman and the train driver relied on an automatic warning signal and traditional semaphore flags. At a safe stopping distance before the tunnel, the signal indicated to the driver whether the tunnel was currently 'clear'. After the train passed this signal, a treadle in the track caused the signal to automatically set to the 'caution' state. When the signalman received a message that it was clear, then the automatic signal was set to 'clear' again by the signalman. The protocol for the driver was that if the signal was in the 'caution' state, then he must slow the train and wait for the white flag from the signalman before proceeding into the tunnel. If automatic signal failed then an alarm would signal in the signalman's box and he could revert to semaphore flags (e.g. a red flag for stopping the train).

The Clayton Tunnel model is distributed on up to six workstations to enable up to three train drivers, two signalmen and one god (for causing signal failures) to take part in operating the stretch of railroad reconstructed from 1861. The original model, initiated by Taylor [Tay97] and subsequently developed by Sun [Sun99], runs on six workstations each running dtkeden. The 3D VRML version, initially created by Woodforth [Woo00] and revised by Chan and myself in 2005 [EMP:claytontunnelChanHarfield2005], offers more flexibility in terms of which people take part in the model.

5.4.4.2 Significance of the EM approach

In evaluating a traditional CSCL application, it would be appropriate to consider the significance of the EM approach firstly for *constructing* the model and secondly for *exercising* (i.e. playing or using) the model. Usually these two issues would be treated separately, one being concerned with software development and the other with software use. The first significant advantage of the EM approach, as introduced in §3.1.5, is that there is little difference between modelling activity that leads to construction and modelling activity that does not involve construction (i.e. exercising a model). From the point of view of a software developer it may be difficult to see the benefits of a blurred distinction between development and use, but for an active learner the benefits are clearer: learners can engage in open-ended *experimentation* through construction; learners are able to *flexibly* choose their own learning path not just exercise a preconceived program; and therefore, learners can explore models in ways that are *meaningful* to their own life and interests.

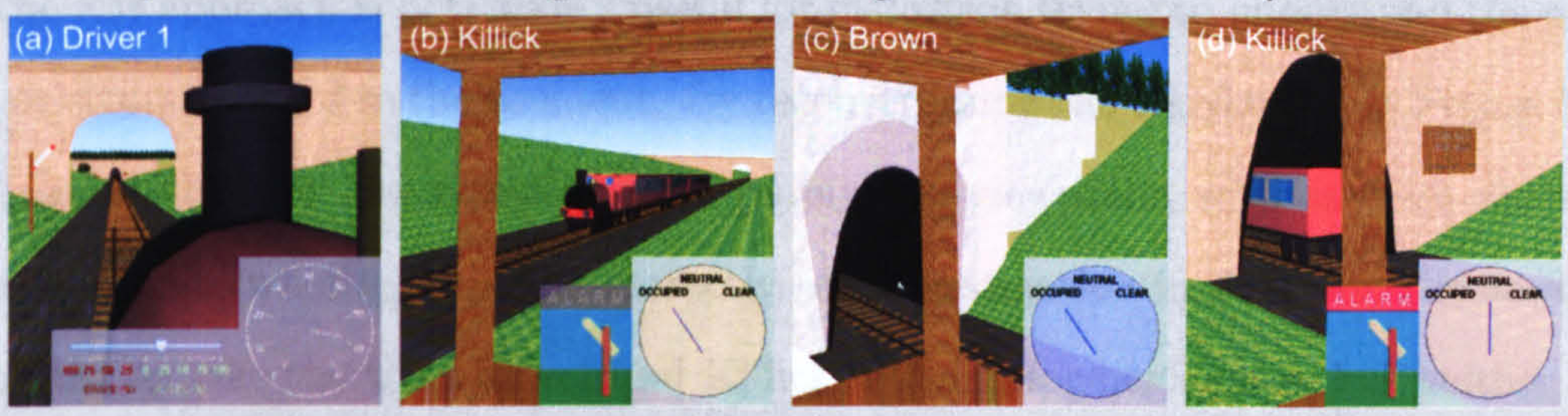
As an example, in exercising the model to understand the causes of the Clayton Tunnel accident, it was advantageous to semi-automate some of the agent's actions due to a lack reliable train drivers in the department. One action to be automated was stopping the train when a red flag is shown at the entrance to the tunnel. This was achieved by adding a dependency to reflect the observation of a red flag by the train driver and a triggered action to apply the brake to the train. Such a change to the model could be considered building a new feature into the model, but it was performed in-the-flow of exercising the model, even whilst the trains themselves were running and being controlled by one of the collaborators. This small automation enables the signalman (whose observation is restricted to the entrance of the tunnel) to experiment with putting the flag out at different times to see whether firstly the train can stop before the entrance to the tunnel, and secondly, whether the signalman is able to determine if the train driver has seen the flag and is stopping—both of which were important details in the events that led to the accident.

The experimental and flexible characteristics of EM enable collaborative model-builders to explore 'what if' scenarios in a completely unrestricted way. Whereas microworlds generally offer a pre-selected set of parameters for experimentation [RB07], in an EM model the whole environment is open to change. The model-builder can experiment with familiar or expected occurrences (e.g. signal failure or brake failure) as easily as unfamiliar or unexpected occurrences (e.g. a train driver acting out of character by stopping at a 'clear' signal, a train disappearing from the tunnel, or a signalman with a sixth sense for detecting imminent train arrivals). EM, by providing the mechanisms with which to explore the domain, offers the potential for a richer experience of the signalman and driver roles that can lead to a deeper understanding of how and why the accident occurred.

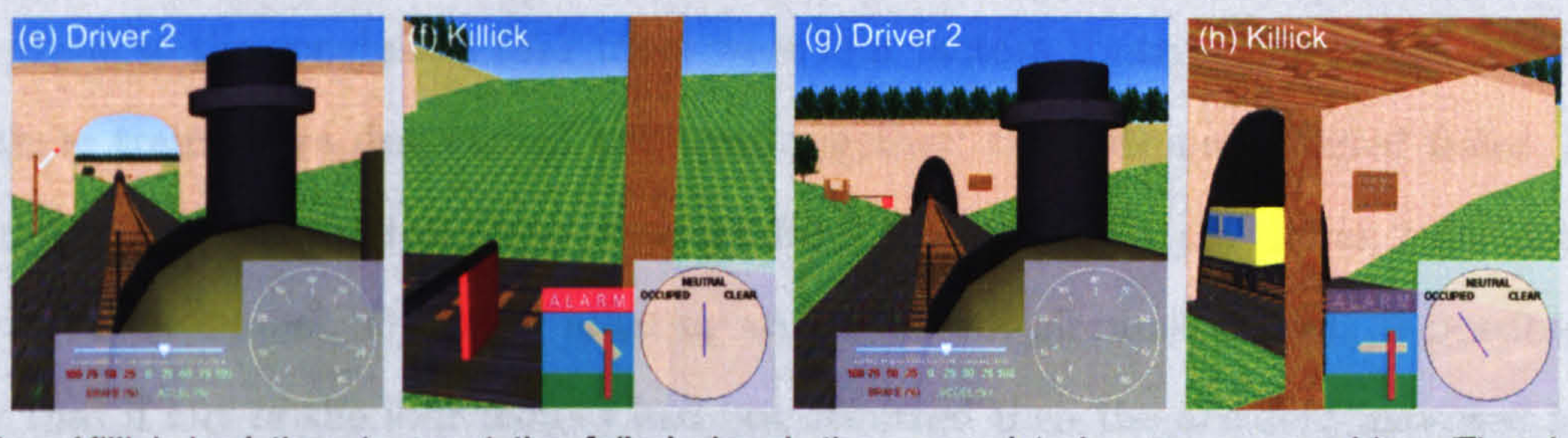
5.4.4.3 Potential for learning

There are a number of ways in which the EM approach to the Clayton Tunnel model could be useful for learning. From a personal point of view, I have been involved with exercising and demonstrating the Clayton Tunnel model over a long period of time, as well as making changes and improvements to the model. In this time I have found that my understanding of the accident has become stronger and can be backed up by experiments in the model. The first few times I observed the model I could not see its significance, and when I exercised one of the agents myself I had very little idea why the accident occurred. Through repeated interaction with the model and through attempts to change it, I started to appreciate some of the reasons why the crash occurred—one being that Killick (the signalman at the entrance to the tunnel) misinterpreted a telegraph message for Brown (the signalman at the exit). In seeking accuracy for the model I found out that one historical account gives the times that the trains left the station giving an indication as to how much time there was between each train. Then by playing Killick's role, I was able to appreciate how little time he had to react when the automatic signal failed, and thus I learnt that the station-master at Brighton played a part in the accident by despatching the trains in quick succession. My

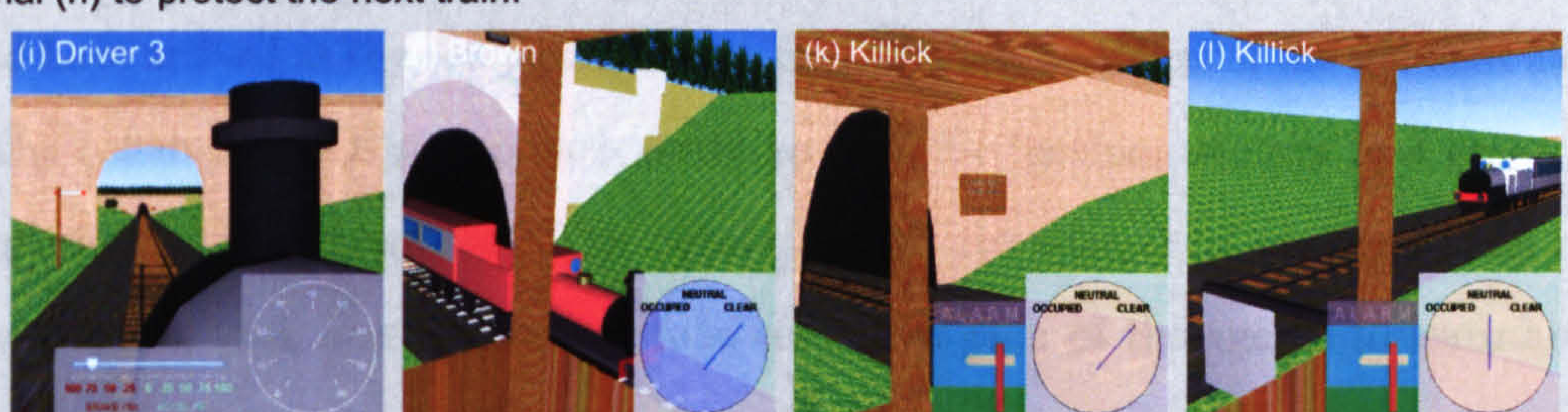
On Sunday 25th August 1861, there were trains due to depart from Brighton station at 8:05, 8:15 and 8:30. The trains were running late, but eventually they were dispatched by the stationmaster in quick succession at 8:28, 8:31 and 8:35 respectively, thus disregarding the 5 minute interval that was required between departing trains. And so began an unfortunate sequence of events.



As the first train driver approached Clayton Tunnel, he observed the 'clear' signal (a) and proceeded into the tunnel (b). The signalman at the entrance to the tunnel, Killick, sent a message to Brown at the other end to indicate the tunnel was occupied (c). Killick realised that the automatic signal warning the next train that the tunnel was occupied had failed to set to caution (d).



Before Killick had time to reset the failed signal, the second train was approaching. The driver observed the `clear' signal (e), not realising it had failed, and proceeded towards the tunnel at full speed. At the last minute, Killick waved a red flag (f) but he was unable to determine if the driver saw it before he entered the tunnel (g). Killick sent an occupied message and reset the failed signal (h) to protect the next train.



As the third train approached, the driver observed the caution signal (i) and prepared to stop before the tunnel. Killick was unsure whether the second train had exited the tunnel and so he sent a message to Brown, asking if it was clear (j). Brown, unaware of the second train, responded with a `clear' (k) as he had just observed the first train leaving the tunnel. Now the Killick had been given the all clear, he waved a white flag at the third train to proceed into the tunnel (l).

However, the driver of the second train had seen the red flag and was able to stop his train inside the tunnel (m). He decided to reverse the train out of the tunnel to find out the problem. The third train had already begun to accelerate (n) when the driver realised there was a train in the tunnel (o). It was too late and the trains collided inside the entrance to the tunnel (p). The crash, one of the biggest in British railway history, killed 22 people and injured 177 people.

Figure 5.27: An account of the Clayton Tunnel accident reconstructed from historical sources and experiments with the model.

current understanding is shown in the account of the accident in Figure 5.27, together with pictures of the reconstructed events from the viewpoint of the relevant agents.

The EM approach is potentially useful for historians trying to understand the problems or errors that led to the accident. By reconstructing the accident as in Figure 5.27, it is possible to view the situation from purely one viewpoint, such as train driver 2, and reason as to whether he acted correctly. An investigation in 1861 into the accident attempted to take the train driver and Killick to court for negligence, but neither was convicted. This seems to me a correct decision, because after playing the accident from each viewpoint there is no single person to blame. In my account (see Figure 5.27), I have come to the conclusion that there were four factors contributing to the accident: the three trains were despatched too close together; the automatic signal failed; Killick misinterpreted the telegraph message from Brown; and train driver 2 decided to reverse in the tunnel. By experimenting with different circumstances it is possible to show that if any of the four factors had not occurred then the accident may well have been avoided. These conclusions arose out of collaborative use of the model, recreating the accident and experimenting with alternative situations. It questionable whether I would have arrived at the same conclusions from written accounts of the accident alone.

The collaborative EM approach could be used for teaching history in a group situation such as the classroom. The agents can be automated to use the model as a demonstration tool, or students can take control of each of the agents to learn about the scenario collaboratively. From experience of demonstrating this model to university students, it is quite difficult to orchestrate the accident as it happened following the scenario in Figure 5.27—it is more likely that students will crash the trains accidentally in trying to understand their role in the scenario and learning to master the controls a train driver should be familiar with. However, learning about operating trains and tunnels is a necessary part of being able to understand the accident. Once mastered, it can lead to more advanced interactions, such as 'what if' scenarios to be explored as an aid to learning more about (e.g.) the reasons behind railway accidents.

There is further potential for EM to be used to support training signalmen or drivers using a collaborative environment like the Clayton Tunnel model. The emphasis on open-ended model-building enables many different situations to be played out that would be too difficult and time consuming to arrange in the world—e.g. reconfiguring the track, changing the position of the signals, or driving different types of trains.

5.4.5 Review of EM's approach to CSCL

One of the benefits of educational technology that has utilised the Internet is that it can enable collaborative learning by a group of people not necessarily in the same physical location. Discussing and sharing are the two types of collaborative activity on the Internet that have been most popular in CSCL to date, but collaborative building is the activity that is emerging as offering the most potential for learning from a constructionist viewpoint [Res96]. The ambition for collaborative building is evident in the Kids' Club project at the University of Joensuu in Finland [ESV02], where school children, collaborating with university students and researchers, build novel technologies using programming environments, robotics and control systems to learn problem solving, creative thinking and ICT skills. Physical construction using intelligent bricks (I-Blocks) developed at the University of Southern Denmark have proved worthwhile in collaborative problem solving at schools and universities in Africa [LV04]. On a more general level, Web-based applications, such as EditGrid [Gib06] developed by Team and Concepts, enable collaborative construction of spreadsheets in an open-ended manner on the Internet. With the advances to EM tools for collaboration highlighted in this thesis (e.g. the 'Blackboard' mode), EM can be seen as the next step in widening the possibilities for collaborative learning by providing a methodology for open-ended model-building that could be used in a broad range of areas. The EM approach to CSCL is unique because it is fundamentally different from programming and it is concerned with learning through collaborative model-building that emphasises the experimental, flexible and meaningful characteristics of learning, as described in Chapter 1 and as demonstrated in Chapter 4.

The jugs experiments have demonstrated that collaborative model-building is possible within an EM framework using the `dtkeden` tool. The first experiment led to the development of a new mode, based on the blackboard as a metaphor, for collaborators to build models that were not possible in previous versions of `dtkeden`. The second jugs experiment highlighted the benefits of the 'Blackboard' mode as offering more potential for the collaborative exploration of models that is in the fluid spirit of EM. The Clayton Tunnel model is introduced as an example of what could be achieved through collaboration and as an example of learning in a specific domain. The author's experience of building and interacting with the Clayton Tunnel model has led to a deeper understanding of the operation of the tunnel, and of the sequence of events that resulted in the famous accident in 1861, than could be expected through written accounts alone.

The jugs experiments and the Clayton Tunnel model have introduced an EM approach to CSCL. Provided that they had an appropriate level of proficiency in dtkeden, there is no reason why other model-builders could not collaboratively build, interact and learn in other domains, whether it be historical events like the Clayton Tunnel accident, or aspects of computer science as discussed in Chapter 4, or languages as discussed earlier in this chapter. In fact, there is potential for any of the models demonstrated in this thesis to be used collaboratively for learning; learning that still emphasises *experimental*, *flexible* and *meaningful* characteristics.

Chapter 6

Empirical evidence of learning through Empirical Modelling

The previous chapters have indicated respects in which Empirical Modelling is intimately linked with learning activity of many different varieties. In this chapter, I will examine further informal evidence in support of this claim that is drawn from various student projects in Empirical Modelling. An assessment exercise attached to the 'Introduction to Empirical Modelling' module is one source of evidence for learning in student projects. This assessment takes the form of an open-ended modelling and paper-writing exercise. Such an exercise is shown to be effective for learning about Empirical Modelling. It also promotes self-motivated exploration in unknown domains that is one of the key skills for experiential and lifelong learning. The extent to which students not only learnt about Empirical Modelling, but also about the domain which they chose to model was unexpected. This leads to the suggestion that Empirical Modelling could be effective in facilitating learning in other domains.

Further evidence of learning through EM can be found in third year projects. These projects are significantly larger, taking up most of a student's academic year and contributing to a quarter of their mark for the year. I shall examine two specific projects in detail in an attempt to understand on a deeper level the ingredients that give EM potential for learning in other domains.

6.1 A module for learning about EM†

'Introduction to Empirical Modelling' is a module that has been run in the last 4 years for final year undergraduates on the 4-year MEng Computer Science course at the University of Warwick. The module aims to teach the basics of EM as briefly described in Chapter 2 of this thesis, and covers the application of EM in a wide range of areas such as artificial intelligence, computer graphics, concurrent systems, human computing, and educational technology. In 2004-5, a new form of assessment for the module was introduced. This involved the informal publication of an online journal 'the first Warwick Electronic Bulletin on EM (WEB-EM-1)' [WEBEM1] to which the students were required to submit papers and associated EM artefacts.

6.1.1 The module assessment

The module ran for 10 weeks, with 2 hours of lectures and 1 hour of computer laboratory sessions per week (with extra laboratory time available). Students were introduced to the concepts of observation, dependency, and agency that they were expected to use when analysing their problem domain. The laboratory gave students experience of the principal EM tool, the `tkeden` interpreter, with its associated family of built-in notations for framing dependencies between scalars, strings, lists, geometric entities and screen display elements. The module also introduced the LSD notation for accounting for inter-agent communication (as discussed in §5.4), and other features of `tkeden` such as agent-oriented parsing [Har03] and models to depict networks of dependencies [EMP:dmtHarfield2006].

For the module assessment, we issued a Call for Papers requiring two submissions. Students first submitted a paper title and abstract. These submissions were reviewed and feedback was given. Students subsequently submitted their full paper and accompanying model.

The coursework had two objectives. The first was to assess the students' understanding of Empirical Modelling through written and modelling exercises based on a common theme of the students' own choice. The second was to equip the students with basic research skills that would be useful in further education. In the Call for Papers,

†This section takes the joint paper presented at ICALT 2006 [BHB06] and builds on the findings with more recent data.

we requested that students submit original and high quality papers relating to Empirical Modelling and its applications supported by a relevant documented modelling study.

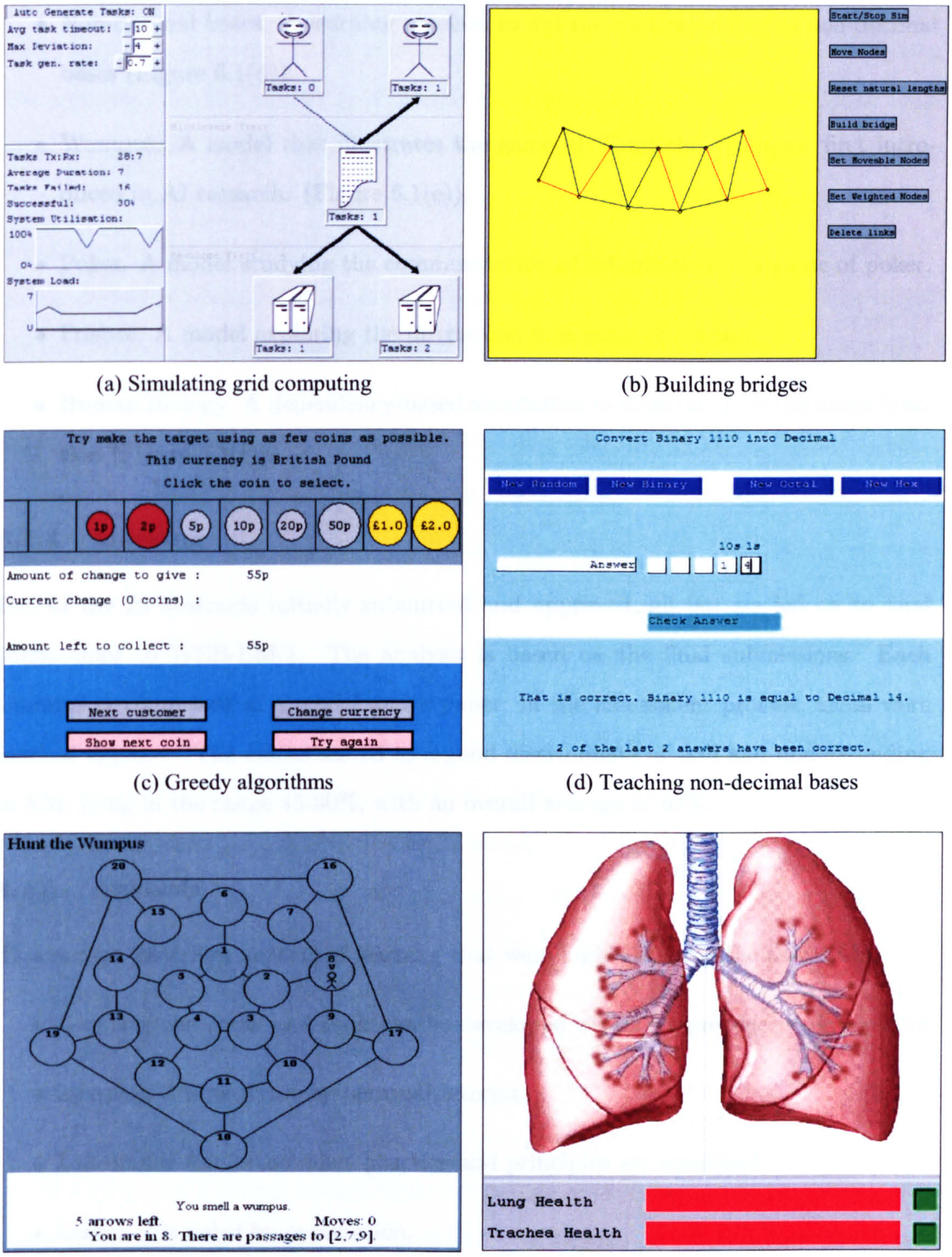
6.1.2 Previous assessments

In the academic years 2002-03 and 2003-04, the coursework assessment required students to build a model using the EM tools. The students were all given the same task in 2002-03 to implement a board game, and in 2003-04 to make a model of a heating system. Although many good submissions reflected the hard work of the students it was felt that the scope was not wide enough for capable students. On the evidence of their submissions, many students were keen to put effort into developing their submissions beyond the original specification even though their knowledge of board games and the workings of heating systems was limited. On that basis, it seemed natural to give students the opportunity to apply the EM tools and principles covered in the module to a subject of their choosing. Since many fourth year students are likely to proceed to research the open-ended assessment was also seen as a good way to promote research skills that would assist their future studies.

6.1.3 Submissions

The selection of the submissions to which I shall refer in the following section is listed below. A complete list of submissions is available from the first Warwick Electronic Bulletin on Empirical Modelling [WEBEM1]. Screen-shots from the models of some of the following submissions are shown in Figure 6.1.

- Tournament. A notation and model for the organisation of knock-out tournaments.
- IceCube. An exploration of IceCube, a technology developed by Microsoft that deals with reconciling divergent replicas of some shared system state.
- Grid computing. A simulation to allow exploration of the efficiency of a computing grid (Figure 6.1(a)).
- Bridges. A model exploring basic engineering principles behind bridge building (Figure 6.1(b)).



(a) Simulating grid computing

(b) Building bridges

(c) Greedy algorithms

(d) Teaching non-decimal bases

(e) Hunt the wumpus AI game

(f) Learning about human biology

Figure 6.1: A selection of models from WEB-EM-1.

- Greedy algorithms. A learning artefact to demonstrate a greedy algorithm for the 'making change' problem with different currencies (Figure 6.1(c)).
- Non-decimal bases. A learning artefact to aid the understanding of non-decimal bases (Figure 6.1(d)).
- Wumpus. A model that illustrates the game of 'Hunt the Wumpus' first introduced in AI research. (Figure 6.1(e)).
- Poker. A model studying the communication of information in a game of poker.
- Frisbee. A model exploring the interaction in a game of frisbee.
- Human Biology. A dependency-based simulation to illustrate how the lungs function (Figure 6.1(f)).

6.1.4 Marking

Out of the 25 abstracts initially submitted and approved, all but six led on to final submissions to WEB-EM-1. The analysis is based on the final submissions. Each submission comprised of a model and a paper; in the assessment process, these were marked together. The marks served as a good discriminate of skill and understanding in EM, lying in the range 45-80%, with an overall average of 63%.

6.1.5 Analysis

This section describes aspects of learning that were highlighted by the assessment:

- Learning can occur and skills can be developed without a preconceived objective
- Learning is stimulated by personal interest.
- Learning is reinforced when practice and principles are combined.
- Learning is aided by exploration.

In the following section I shall consider respects in which Empirical Modelling is well-suited to supporting learning that exhibits these characteristics.

6.1.5.1 Learning can occur and skills can be developed without a preconceived objective

As in previous years, the coursework helped to develop practical skills with the EM tools. However, the potential for emphasising different aspects of Empirical Modelling was apparent with this new style of assessment. Some students stuck to the basic tools whilst others made use of other, often more technical, tools and notations. The Frisbee model made use of only of the basic notations for data manipulation and line drawing, introduced at the beginning of the course, and the student was clearly proficient in building models with these notations. Other models, such as Tournament, involved the development of special-purpose notations which exercised a different skill-set associated with agent-oriented parsing. Another student modelled the game of poker from different viewpoints using the distributed EM tool. Others emphasised the incremental aspect of model-building in their model. For example, the Making Change model used incremental development to show how a learning artefact might be adapted to situations that arise and evolve as the understanding of the learner develops. Each student developed the same basic skills but some also demonstrated extra skills. The fact that students could choose what skills to develop within certain broad constraints contributed to the diversity and richness of the submissions.

The journal-style of assessment demands a different skill set from the typical computer science coursework. Coursework is usually a specific task which the student should tackle in a preconceived way and hence often results in similar submissions. In our assessment, the students were given a set of tools and asked to develop their own theme within a general framework of possible applications of EM. This required the student to be self-motivated and to think for themselves about how they should approach the coursework. As can be seen from Figure 6.1, a wide range of topics and interests was represented in the submissions.

We found that students were able to direct their own learning without being given a specific coursework objective. The student who submitted the Wumpus model initially set out to reconstruct the original Wumpus game using the EM tools provided. This proved successful but furthermore the interactive, open-ended nature of the environment allowed the student to model different scenarios they had not originally considered, e.g. by changing the rules of the game and/or manipulating the information presented to the player. In his submitted paper, he explained how through this interaction with his model he had begun to appreciate how his ability to win the game depended upon the rules of the artificial Wumpus world and how, outside such a constrained environment, pure reasoning was not always sufficient.

6.1.5.2 Learning is stimulated by personal interest

The account of learning in Chapter 1 cites James [Jam25] and Dewey [Dwo59] stressing the importance of personal interest as a motivating factor for active and experiential learning. In this spirit, the students were encouraged to think about issues of which they had particularly rich experience or were particularly interested in learning about. One student explored applications of greedy algorithms by carrying out empirical research into how her younger siblings learnt about giving the correct change. Coursework often forces students to study situations with which they are unfamiliar or topics that do not interest them. By choosing their topic, students were able to draw and build upon a wide range of prior knowledge, interests and experiences. EM actively encourages this type of learning [Roe03]. Because students worked on topics of their own choice, the focus of their effort could be on EM principles and tools and not on an arbitrary topic prescribed for them.

All of the submissions showed evidence of an interest in domains other than Empirical Modelling. These domains ranged from personal hobby interests to aspects of the computer science curriculum. The Poker and the Frisbee models were inspired by recreational interests. One student made use of his Grid Computing model to complement his coursework for another computer science module. Another developed some research by Microsoft into the IceCube framework. From the depth and quality of his submission, it is apparent that this student spent as much time learning about IceCube as they did about Empirical Modelling. This contributed significantly to the quality and ingenuity of his final model; it also demonstrated how the model-building could stimulate learning in other domains. Yet other students chose to model phenomena in other academic subject areas. One submission relating to human biology was a model of the lungs that incorporated a primitive simulation to expose the effects of damage to organs or of cigarette smoking. The simple but effective use of dependency in this model highlighted the extent to which naive medical knowledge of bodily functions is knowledge of basic inter-relationships between physical conditions and parameters. This underpinned the educational purpose behind the model—just one of many references to education in the written submissions.

6.1.5.3 Learning is reinforced when practice and principles are combined

The practical element of a subject can often become divorced from reflection on principles. Although model-building is a useful tool for developing basic EM skills, it should be guided by higher-level motivations and interpretative activities. In previous years, the written component of the coursework had been primarily oriented towards the technical documentation of models. Introducing the paper-writing exercise into the written component of the coursework helped to promote a broader awareness of the thinking behind EM and its implications.

Several models successfully illustrated deeper concepts of direct relevance to EM. In the Wumpus model, for instance, the environment can be configured so as to expose the limitations of logic outside a context of stable expectations and reliable knowledge. In his extension to the traditional AI game, the student was able to expose problematic aspects of a purely logicist outlook on intelligence in a manner that had not been preconceived. The Grid Computing model was a good example of a student using a model to convey concepts. In this case the model served to illustrate the basic principles of grid computing by generating animations of the kind of diagram that would typically be found in an introductory textbook.

6.1.5.4 Learning is aided by exploration

The quality of the submissions was such that most of the students were able to grasp the use of the EM tools and, in some cases, exploit their more advanced aspects. This is one reason why exploration into other domains occurred so naturally in the coursework. Once the tools had become familiar, the student no longer had to focus on the technicalities of modelling, but could make use of the tools to communicate or develop their domain understanding. As is to be expected, the better the student's EM skills, the more they were able to explore their problem domain.

To demonstrate this, we have categorized the submissions by the extent to which they explored their problem domain: little/no exploration, controlled exploration, and free exploration. Submissions that showed little/no exploration were generally based on the style of implementation that is quite familiar in computer science. A typical example is a model of a game that concentrates more on the implementation than exploring the observations, rules and interfaces that shape the interactions within the game. In the 'controlled exploration' category, the submissions often related to a problem domain with which the student was familiar, possibly drawn from the academic field. An example is the Human Biology model that enabled the user to explore the effect of smoking on the oxygen intake via the lungs. Models in the 'free exploration' category typically resulted from a student's engagement in unfamiliar problem domains. For example, the Bridges model originated in a basic study of the strength of bridges and ended up modelling complex issues in suspension bridges. Applying a statistical T-test at a 99% confidence level shows that students who engaged in exploration achieved higher marks than the students who did not show signs of exploration in their coursework.

These observations suggest that students who had a good grasp of EM tools were able to engage fully with the problem domain and produce coursework of a higher standard. This is what you would expect as we were evaluating Empirical Modelling ability rather than expertise in the problem domain.

6.1.6 The Second and Third Warwick Electronic Bulletins on EM

The submissions from the First Warwick Electronic Bulletin on EM (WEB-EM) demonstrate that there is some potential for EM to support learning that exhibits the above characteristics. These findings were published in [BHB06]. Since then, the module has continued to run in the same format and, so far, a 2nd WEB-EM and a 3rd WEB-EM have been published online [WEBEM2] [WEBEM3].

One additional aspect to the data is that I requested the students to fill in a short questionnaire at the end of the course (before the final examination). The questionnaire was in two parts. The first part contained 17 statements which the students were asked if they agreed with, rating their level of agreement on a scale of 1 to 5. For example, one statement on the questionnaire was "The labs were easy", and students were asked to tick box 1 if it they felt it was always true (i.e. 'I completely agree'), 2 if it was mostly true, 3 sometimes, 4 rarely, and 5 if it was never true (i.e. 'I completely disagree'). The second part of the questionnaire asked the students to write a short (up to a page) biography of their experience of EM. They were asked to reflect on their first contact with the subject, their personal experiences during the labs and coursework, the difficulties they encountered and the problems they overcame. The students were only given 10-15 minutes to fill in the questionnaire. The questionnaire was not an official part of the course, it was not compulsory, but most of the students were happy to fill in the questionnaire, and from the results it seems that many of them put in considerable effort to relate their experiences of EM in the second part of the questionnaire.

In the first three years of WEB-EM there were 43 papers (along with 43 models) published. (A complete list of WEB-EM submissions are detailed in the appendices.) The extra 2nd and 3rd year data has strengthened the original findings. The 4 characteristics highlighted in WEB-EM-1 are also relevant to the 2nd and 3rd year submissions. Reflecting on the overall experiences of the three years of submissions, as well as the feedback in the last two years, there are a number of other points that are worth highlighting.

6.1.6.1 The value of the coursework part of the module

In the first part of the questionnaire, students were asked to rate how much they agreed with 17 statements. Only 3 of these statements were related to the coursework. In analysing the results, I discovered that the three coursework statements were in the top five 'most agreed' statements (see Figure 6.2). (The other statements that were in the top five in terms of agreement were: "the lab tutors were helpful" and "I enjoyed the lab sessions".) Figure 6.2 shows the results from the first part of the questionnaire. Of the 24 students that took the module in the years 2005 and 2006, I was able to collect 17 completed questionnaires.

The statements in Figure 6.2 are sorted according to the level of agreement with each statement. The 'Agreement' column represents the mean as a percentage, in such a way that if the average answer to a statement was 1 (i.e. every person agreed with the statement completely) then the truth percentage would be 100% (and if mean was 5 implying every person disagreed with the statement completely, then the truth percentage would be 0%). The top statement is therefore the one that was felt to be most agreeable on average.

The three coursework related statements are highly ranked in second, fourth and

Statement	Agreement	Mean	S.D.
The lab tutors were helpful	88%	1.47	1.07
The coursework was relevant to the course	87%	1.53	0.80
I enjoyed the lab sessions	82%	1.71	0.99
The coursework helped me understand EM	78%	1.88	0.78
The coursework helped me understand more about my chosen topic through the modelling exercise	76%	1.94	0.83
The labs helped me understand the principles of EM	75%	2.00	1.00
The lab sheets were clear and instructive	72%	2.12	1.22
There are advantages to using EM tools over traditional programming tools	72%	2.12	0.78
I understand the principles of EM	68%	2.29	0.69
The labs were more useful than the lectures	63%	2.47	0.80
The EM tools (tkeden) are easy for beginners	62%	2.53	0.87
The tools are unpredictable	62%	2.53	1.07
The tools are easier to learn than Java	54%	2.82	1.01
EM has helped me understand other aspects of CS	50%	3.00	0.87
The labs were easy	49%	3.06	0.75
EM has no relevance to the rest of the CS course	41%	3.35	1.06
The tools are easier to learn than spreadsheets	37%	3.53	1.37

Figure 6.2: Part 1 results from the 17 questionnaires.

fifth positions in the table in Figure 6.2. The mean for the statement "the coursework was relevant to the course" was 1.53, representing an 87% level of agreement. This was the considered the second most agreeable statement among the students out of the 17 statements. The two other coursework related questions offer more insight in terms of analysing the effectiveness of EM for teaching and learning. The statement "The coursework helped me understand EM" achieved on average 78% agreement. The high rank of this statement helps to confirm the constructionist idea that practical building activities are a benefit to learning. The statement "The coursework helped me understand more about my chosen topic" provided the most interest as it achieved almost as much agreement as the previous statement on understanding EM. This is surprising because there was no intention for the students to learn about another domain—they were simply asked to engage in a model-building exercise using EM. However, there is a strong agreement that by engaging in an EM model-building exercise the students have learning more about their chosen topic. Such results echo Jonassen's view that if you want to understand a topic then it is useful to build a model of it [Jon06].

The relatively low standard deviation (S.D.) on the coursework statements implies that there was more consistency between students' opinions of the coursework than other aspects of the module (the students agreed with the statement and also with one another!). This shows that there is some indication that these three statements have an element of truth for the majority of students.

The weakness of this analysis is that it is based on 17 results. Due to this, the first part of the questionnaire cannot offer any conclusive evidence. I was not really expecting to find any significant results in the first part when I designed the questionnaire, although I was surprised after analysing the data that the only strongly held opinions among the students were regarding the coursework. My interest was really in the second part of the questionnaire where I asked to students to write a small biography of their EM experience and to give comments. Given that quantitative data would suffer from the low numbers of students taking 4th year modules, I felt that the qualitative evidence might prove more relevant to this thesis. The idea of using biography as a method for analysing EM activity in the 'Introduction to EM' module was inspired by Knobelsdorf & Schulte, whose talk I attended at Koli 2005 [KS05]. Their research is concerned with understanding how students' learn computer science, not what factors affect the learning: "Research has already revealed many influence factors, like e.g. gender, math grade, role model, prior programming experience, self-confidence, and so on. But addressing one (or some) of these factors might not be sufficient to change students general understanding of computer science or to improve the effectiveness of teaching. Instead of revealing more (singular) influence factors we aim to understand students' preconceptions, their conceptual framework of the subject matter and how it evolved." It is in this spirit that I wish to evaluate learning about and learning through EM.

Biography as a method (as used by Knobelsdorf & Schulte) is discussed in detail by Denzin in his book entitled *Interpretive Biography* [Den89]. Denzin recognises various forms of biography, all of which are concerned with the use and collection of "personal-life documents, stories, accounts and narratives which describe turning-point moments in individuals' lives" [Den89:p13]. To account for the wide range of possible biographies that Denzin describes, I left it to the students to decide what they would like to write about in the second part of the questionnaire. As a guide I suggested that they write a short autobiography of their experience of EM, possibly reflecting on their first contact with the subject, their personal experiences during the labs and coursework, difficulties that were encountered and problems that were overcome. Many of the students completed their questionnaire including their autobiography during a revision class, and so only spent a small amount of time on the exercise. Also, only a page was given for the students to write their autobiographies including any comments. These factors meant that the biographies were not as comprehensive as those gathered by Knobelsdorf & Schulte [KS05].

Many of the biographies mentioned the coursework in a positive manner. In a significant number of biographies there were indications that the coursework reflected the most important part of the module: "I think the labs were the 'frustrating' period of learning [...] whereas the coursework allowed you to get deep into a model and actually understand the power of EM." This complements the results from part 1. There are a number of comments that other aspects of the course, such as the labs, were not as useful as the coursework. One student explained that the coursework clarified the ideas of the module: "The first lectures were obscure to me and I didn't get the idea. Later, facing the first draft of my coursework, I realised aspects of the lecture and began to understand." This last sentence builds on the sentiments of section §1.2.6 that learning occurs best when principles and practice are combined. In referring to the EFL (Figure 2.7) discussed in Chapter 2, learning inevitably must involve a practical/tacit knowing and a more explicit knowledge [Roe03].

6.1.6.2 EM offers something different to the standard CS curriculum

In the biographies given by students there is a common attitude that EM offers a complementary perspective on computing that is not acknowledged or covered in the standard components of a computer science course. One student commented: "The main thing that EM has encouraged me to think about is the relationship between computing and thinking, which as a more abstract concept had not really been touched upon in the rest of the CS course." Similar feelings were echoed by another student: "The first lectures seemed quite philosophical and were a welcome new take of CS. Nothing else on the [computer science] course has really considered things the way EM does."

Furthermore, evidence from one student suggests that EM can be beneficial in other areas of computer science: "I found EM to be a completely different module to any I have studied." ... "It took me a while to gain an appreciation for the module, but I have learned to look at all aspects of my course from another perspective." Although the student is not specific about what aspects of the course he was able look at differently, the statement shows that there is some potential for using EM for widening a computer scientists view of their subject in a positive manner.

Another student was more specific in describing the benefits that EM brings to a computer science course: "Its great and refreshing to see things up on a screen and how we can alter it. I enjoyed programming in it, something I thought had lost through finding other languages less 'exciting'." The immediacy of experimentation is highlighted here as a refreshing aspect of EM. The comparison with other programming languages is hinting that EM offers something special that other languages are unable to offer.

6.1.6.3 Difficulties in understanding the theory and using the tools

Despite the positive comments in biographies substantiating the potential for EM as a unique approach to computing, there were also a number of comments describing the difficulties of EM. The first issue is that students develop misconceptions initially and it is not until later in the module, usually during the coursework, that these misconceptions begin to be rectified to some degree. The collection of biographies contained many comments similar to this: "My first contact with EM in the first few lectures led me to mistakenly believe that it is another programming course like the java or functional programming module I took in year 1." There was one student who still held some misconceptions at the end of the module: "Still don't get how EM differs from simulation other than being more flexible."

Another issue was that students were seriously concerned by the 'what is EM?' question. Computer science students it seems are keen to have definitions, and when a simple definition is not easy to find or representative of the activity then it presents a difficulty for many students. One student put it very simply in their biography: "Understanding exactly what EM is was quite difficult." Another student observed a general feeling that there was not a simple link between the philosophy and the practice: "I got the feeling from some people that they couldn't understand the link between their application & the EM principles." King's work is a recent attempt to define EM as a philosophy clearly connected to practice [Kin07], resulting from his own reflections upon the question 'what is EM?'.

The final issue relates to the quality of the EM tools and has been discussed in detail elsewhere [Kin07]. The tkeden environment is an experimental piece of software developed by students, over 18 years, and suffers from many idiosyncrasies. A number of student biographies highlighted problems with the tool, for example: "The tools (tkeden) are functional but still could benefit from small usability improvements—some times you're wrestling against the tools instead of doing the actual modelling." It is clear from this that the quality of the tool is lagging somewhat behind the quality of the principles. A number of improvements have been made to the tools for educational applications, as used in this thesis (i.e. AOP, GEL, & EMPE, as discussed in Chapter 4), but further work is needed.

6.1.7 The significance of EM for learning

The characteristics of the learning exhibited in the WEB-EM submissions accord well with the characteristics of EM. Where conventional programmers are encouraged to assemble a secure base of knowledge prior to writing the first line of code, EM practitioners are encouraged to initiate their exploration of the application domain and their construction of a computer-based model simultaneously. The fundamental reason for this distinction in outlook has to do with the perception of knowledge that underlies thinking about conventional programming and EM. The conventional programmer targets sophisticated knowledge that is sufficient to provide a robust logical framework ('knowing that certain relationships hold') and complementary precise recipes for action ('knowing how to achieve specified goals'). By contrast, EM is primarily concerned with a much more primitive conception of knowing (cf. [Bey05a])—with conjunctions between experiences as personally encountered by the modeller. The qualities of EM artefacts stem from this grounding in experienceable connections that pervades the context within which all 'knowing that' and 'knowing how' is subsequently rooted.

The principal technical contribution of EM to moulding one experience so that it 'knows' another is to be found in the notions of observable, dependency and agency. The diversity of the domains represented in WEB-EM-1 as depicted in Figure 6.1 is further evidence of the pervasive relevance of these primitive notions, which are viewed as conceptually prior to the identification of formal objects and structures. The integrity, functionality and interpretation of an EM artefact is framed by the meaningful interactions that the modeller can carry out with it, as guided and constrained by past experience of interaction. In this respect, EM artefacts are ontologically quite unlike computational objects and structures, even though in practical settings they may resemble them closely. The character of EM artefacts is consistent with the features of the learning activity: not necessarily being associated with a specific learning objective; connecting closely with personal experience; synthesising empirical and theoretical elements; drawing upon extensive exploratory activity. These characteristics are also relevant to whether students chose to pursue the 'Introduction to EM' module to its completion, since it epitomises an orientation towards knowledge that is congenial to some but alien to others. Unsurprisingly, there is only a loose correlation between good overall performance in computer science and aptitude for EM.

As the module was an introduction to Empirical Modelling, it was intended and expected that the students would learn about the basic concepts of EM. Students demonstrated their proficiency with EM tools and techniques through the models they were able to construct. What is more surprising is the extent to which some students learnt about the topic area chosen for their modelling exercise in carrying out the assessment. Learning evidently occurred in both domains—in EM and in their area of interest. Given the strong agreement among the students, it is likely that this will be the case in other domains beyond those covered by student submissions to date.

Furthermore, the assessment exercise promotes those meta-skills relating to self-motivated, self-directed exploration of a problem domain that are most needed by life-long learners (see §5.3). It is characteristic of life-long learners that they are not necessarily following a formal path of education. They are much more likely to have personal goals and individual learning objectives. The above review indicates that EM tools have the potential to address the needs of life-long learners, as discussed in more detail in Chapter 5.

EM and the assessment strategy seem to work well together, and to offer prospects for learning in other disciplines. Whether the style of assessment would be so effective with traditional programming tools is unclear, since it is hard to interpret embodying observation, dependency and agency in models in that context.

When all students built models from the same specification, assessment was more straightforward because it was easier to identify the relative merits of each piece of work. The new style of assessment has meant that there is a greater focus on the use of EM principles and how they have been applied to the specific topic area chosen by the student. This presents challenges for the marking procedures as it requires additional time to familiarize with each individual model and its topic. However, we have found that the students who focused on unusual subject areas for their model building tended to build models that in themselves promoted our own understanding of that area.

6.2 Third year projects

There is further empirical evidence of EM's potential for supporting experiential learning in a number of undergraduate third year projects. Every computer science undergraduate must undertake a project in their third year which counts for a quarter of their marks for the year (the project is worth 30 CATS and the normal load for a year is 120 CATS). Students may undertake a project of their choice (within the bounds of the course and assuming they find a supervisor who deems the project suitable). For a number of years there have been students undertaking projects relating to Empirical Modelling, supervised by Meurig Beynon and Steve Russ. Each year there have been up to 15 students, and so over a number of years the number of projects has been substantial. Some of these projects are available and freely downloadable from the EM Archive [EMP], which as of May 2007 contains over 161 models. These projects have contributed significantly to the EM tools and to students' modelling activities, as well as the thinking and philosophy behind EM. Many (if not all) of these projects have a learning element that is significant to this thesis. A complete account of all the third year projects is beyond the scope of this thesis. However, these projects perhaps stand as the best source of empirical evidence to date of deep significant learning through Empirical Modelling. This section only touches upon a few selected aspects of two selected projects in which I feel there is a strong sense in which EM has led to significant learning. The aim is to show on a micro-level the contribution that EM can make to the learning of an individual exploring a particular domain.

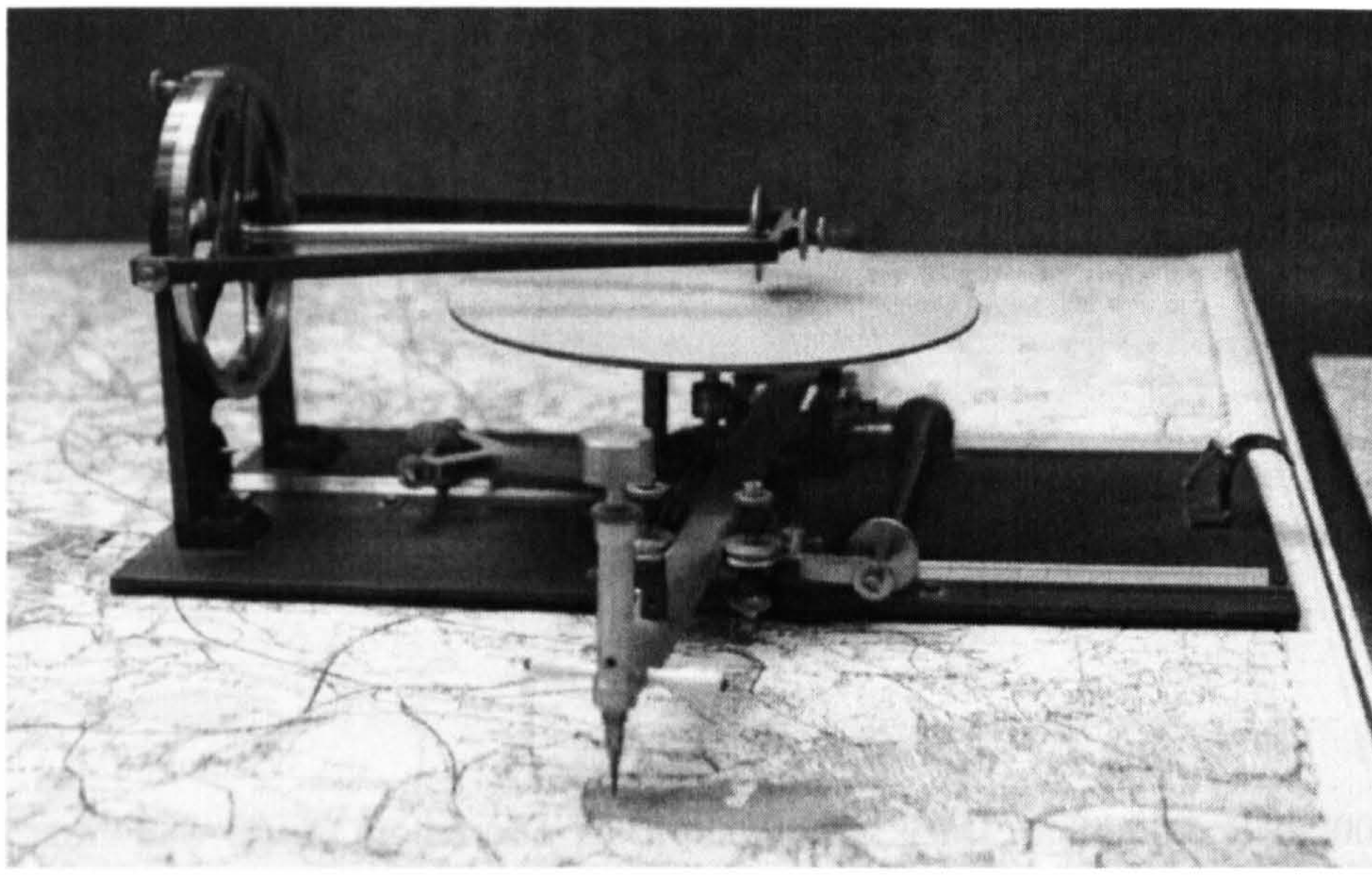


Figure 6.3: A Wetli planimeter as displayed in the London Science Museum (replicated from [War04:p13], photo by Mark Lloyd).

6.2.1 Learning about planimeters

The first third year project I would like to highlight is that of Charles Care who developed several models of planimeters for his final year undergraduate project [Car04b]. A planimeter is a mechanical device, invented in 1814 for land surveying, that can be used to measure area [Asp90]. It is considered an important device in computing as its integration capability signifies the beginning of analogue computing [Car04b]. An early example of a planimeter is the Wetli planimeter that uses a disc and wheel mechanism as shown in Figure 6.3. Care developed a number of models using EM techniques including a 2D model, a distributed model, and 3D models of different planimeters—Figure 6.4 shows a 3D model of the Wetli planimeter (developed by Care [Car04b]).

The potential of the planimeter models for learning about planimeters is what strikes me as most significant about Care's project. During a recent talk about planimeters given by Care, he mentioned the effect of his explorations building models of planimeters as leading to a better understanding of their workings than historians might usually obtain. Care described how on a trip to the Science Museum he met a curator with whom he discussed planimeters. To his surprise, he appeared to have a deeper understanding of the workings of the planimeters on display simply from this experience of constructing and playing with models of planimeters. This suggests that there is a special power in constructing models that brings about a deeper understanding than

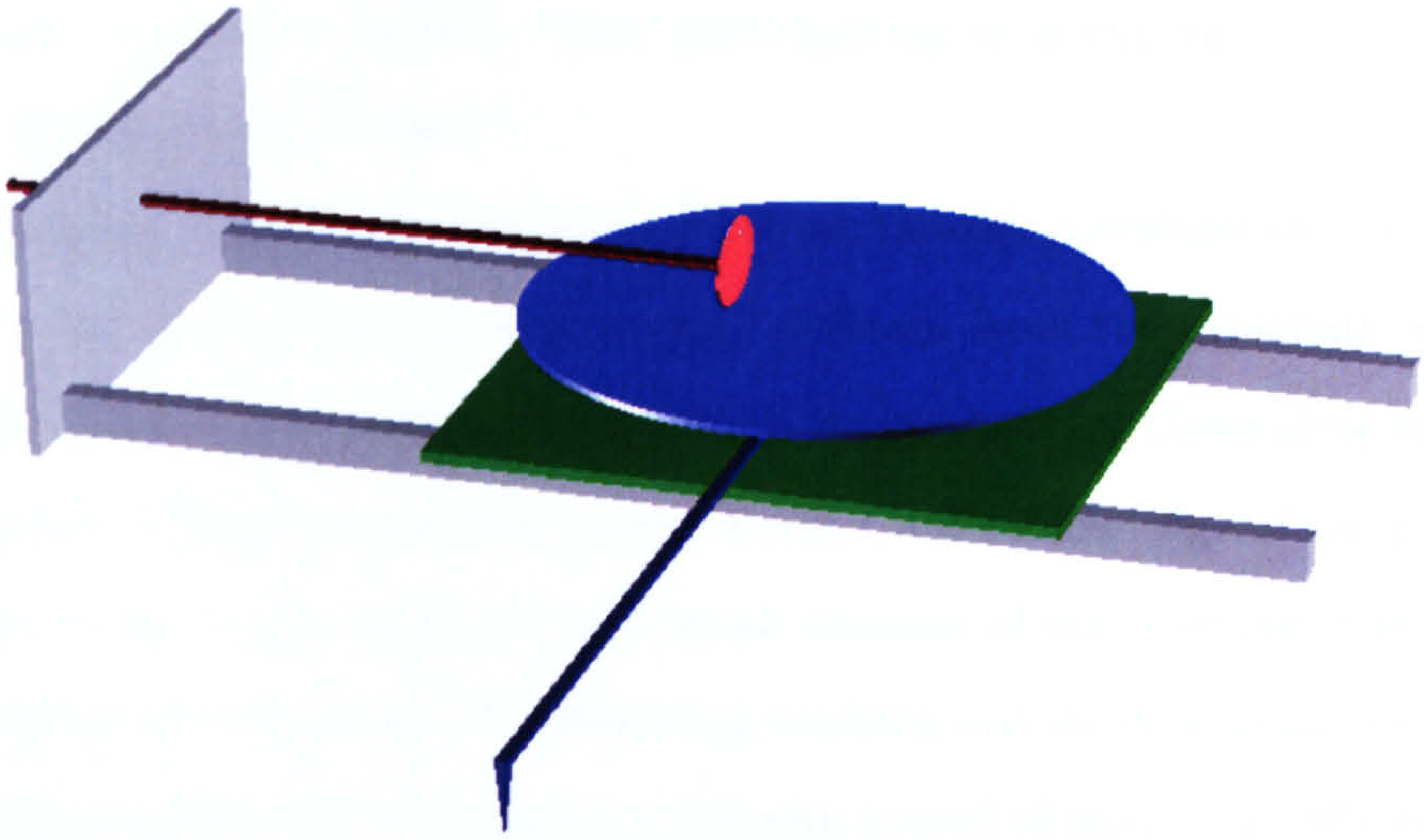


Figure 6.4: Care's model of a Wetli planimeter [EMP:planimeterCare2005].

can be grasped from language alone. The curator, in this case, had available to her at least as much information—in the form of books, historical accounts and records relating to the planimeter—as Care had available to him. Care was surprised because he expected a curator interested in planimeters to have developed a deeper knowledge of the subject. However, Care used the information about planimeters to construct an EM model and it is this process that might have led to a deeper understanding of the workings of planimeters and the important issues around planimeter design. In a paper written after his project, Care shares a similar conviction:

> "Descriptions of various planimeters, their construction and how they work are documented in a number of sources. However, these usually take the form of annotated diagrams with written explanations that are difficult to relate to. Such descriptions fail to give the reader anything like the experience of holding a real planimeter in their hands. My suggestion is that the models described do provide this kind of experience and that this property is not simply associated with the model but also with the modelling process employed." [Car04a]

The deeper understanding that comes from the modelling process is perhaps not as expressible as the objective information about a planimeter that is generally found in books. This deeper understanding might be referred to as 'tacit knowledge' as described by Polyani in his work on the nature of knowing [Pol62]. If it is tacit knowledge then I suspect it is more like the understanding that comes about from the experience of designing and the repeated experiences of using a planimeter. Care did consult technical documents on the design and construction of planimeters, which would fit Lave & Wenger's idea that learning involves a deepening process of participation in a community of practice [LW91]—Care participating in a community of planimeter engineers that no longer existed.

In discussions with Care, I have come to believe that his investigations into planimeters show a deep understanding that is quite different from the explanations given in books, such as the mathematical description in Aspray's book *Computing before computing* [Asp90]. There is something special about Care's knowledge of the planimeter. I can, at best, try to give a glimpse of a small element of the learning that has taken place during his investigations. The following explains one small 'eureka moment' that occurred within a few days early during a 6 month period of modelling. My exploration of the modelling activity is reconstructed from a history of scripts from 24th to 27th November 2003 that were carefully recorded by Care. Each script represents a point (or state) of significant change in his modelling activity, although I have identified from logs that in between these saved scripts there were a significant number of interactions. A complete analysis of these scripts is included in a separate document [Har07b].

6.2.1.1 The incident

Care's 'learning about planimeters' began after he became interested in planimeters during a module on the history of computing. His third year project was originally aimed at demonstrating that computer models of historical artefacts are beneficial for learning, although after a while the project became more of an investigation into the history of planimeters and their workings. In the initial stages of his project, Care was concerned with constructing a computer model (using EM) of a planimeter. He started considering a small part of the planimeter device, the wheel and the disc, which are essential to the integration function. In the Wetli planimeter (Figure 6.3), the small wheel sits on top of and at right angles to the disc. When the disc spins, friction causes the wheel to turn. The amount the wheel turns relative to the disc depends upon how close the wheel is to the centre of the disc. If the wheel is at the edge of the disc then it will turn more (when the disc spins) compared to if the wheel is nearer the centre. A typical diagram used to teach this principle (and the actual example in the History of Computing module) is shown in Figure 6.5. It is from this mathematical description that Care began his disc and wheel model.

At 12.59pm on the 24th November 2003, Care sat down at a workstation in the

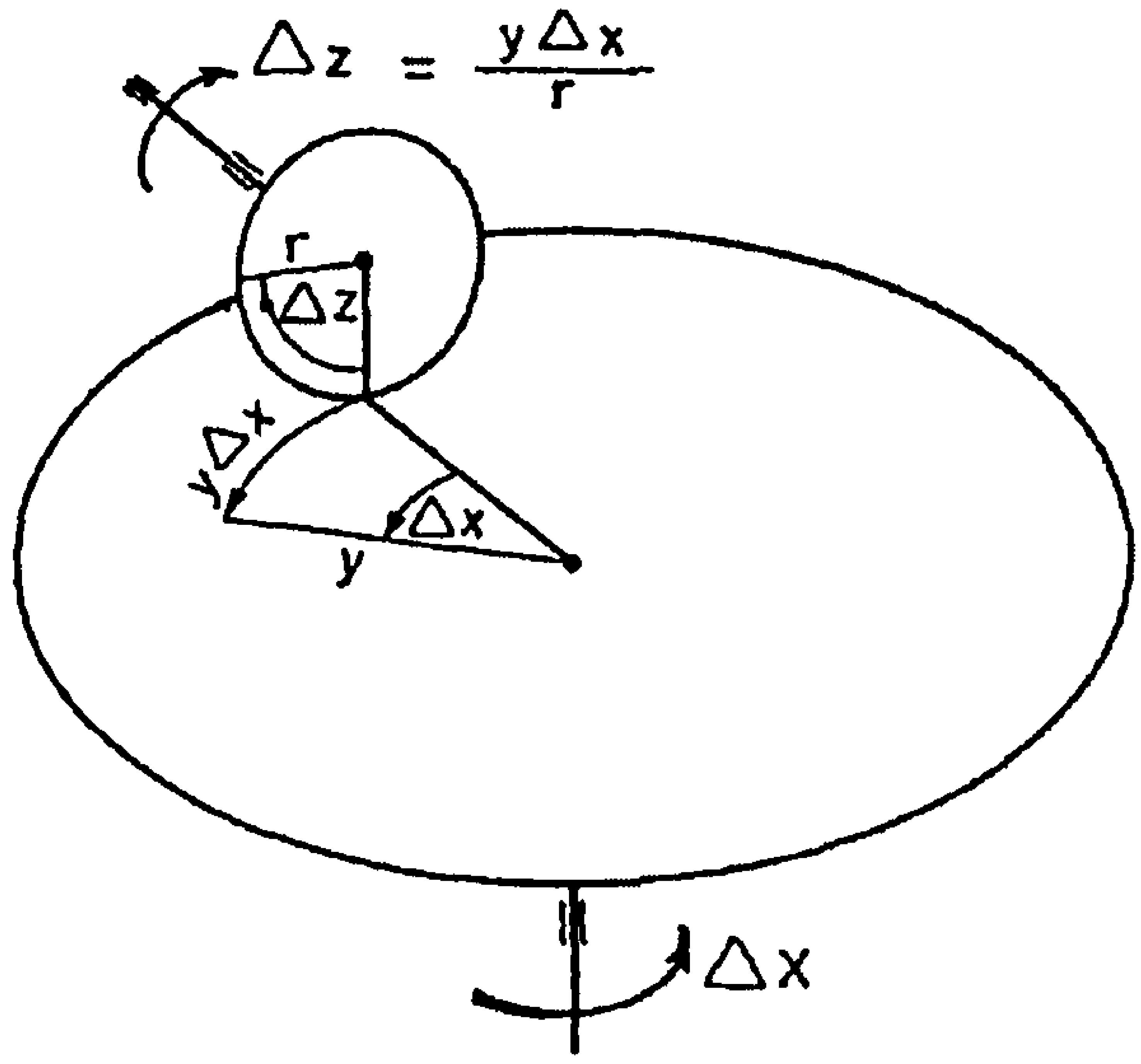
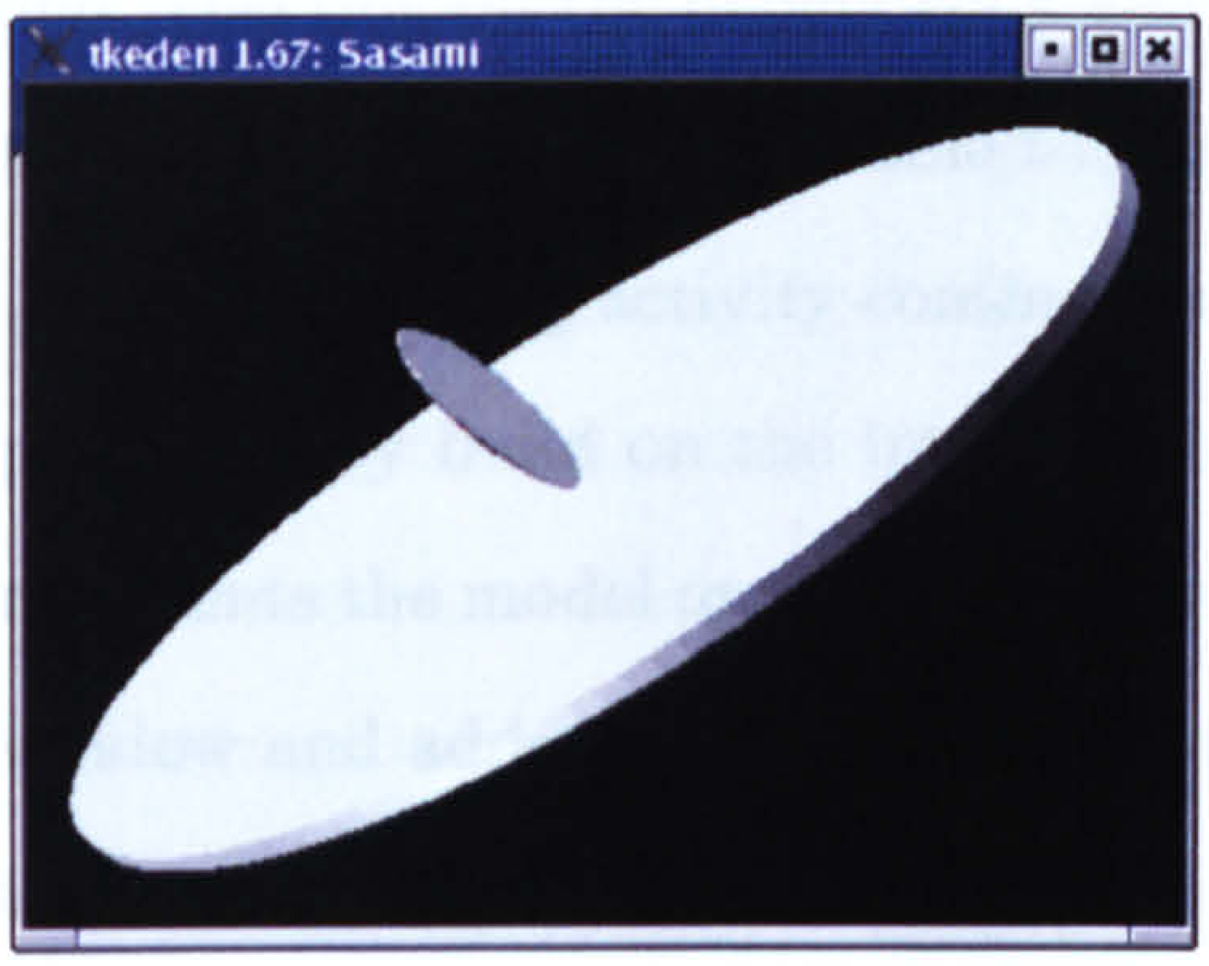


Figure 6.5: "The principle of operation of the Wetli disk-and-wheel integrating mechanism. The dependent variable, y, determines the distance of the integrating wheel from the center of the disk. If the disk is rotated through an angle, Δx, then the integrating wheel is turned by $\Delta z = [y\Delta x]/r$. Hence, after a period of action, $z = 1/r \int y\,dx$." (text and diagram taken from [Asp90:p167]).

Computer Science building, loaded tkeden and started experimenting with the model he had been working on the previous week. At this early stage in his project, his model consisted of simply a 3D disc and wheel. A screen-shot of his model and a selection of the relevant underlying observables and dependencies are shown in Figure 6.6(a). These observables have a direct correspondence to the variables explained in Figure 6.5. The observable 'wheelPos' is equivalent to y, 'discSpeed' is Δx (speed corresponds to the displacement in one clock tick), 'wheelSpeed' is Δz, and 'wheelRotation' is z. It is from here that Care began to explore the model further and eventually realise a subtle problem with the dependencies in his model's current state as of that day.

In the next hour he began to construct a mouse sensitive window with which he used to make the position of the wheel dependent on the position of the mouse. He took a break at 1.58pm, but returned at 4.05pm and continued where he left off. Care started adding marks to the disc and wheel (a blue square on the disc, and a red square on the wheel) which helped to indicate the relative speed and amount of rotation of both the disc and the wheel. Care stopped again at 4.46pm, and did not resume modelling until 9.22am the following day (25/11/2003). In the next script that was recorded, we can see that Care had been altering the disc speed and in particular slowing it down. I can only assume that this was to better observe its behaviour, and perhaps even at this stage there was a hint of suspicion in Care's mind that his model was not operating in the same way as the physical device. By 10am, Care had added another mouse sensitive scout box to control the speed of the wheel. The screen-shot in Figure 6.6(b) shows the state of the model at this time. This addition enabled Care to experiment with the position of the wheel and the speed of the disc at the same time. It was the combination of marks on the wheel for observing the speed, the sensitive box for controlling the speed of the disc, and the sensitive box for positioning the wheel that led to an important discovery. Care had worked until 10.04am when he left his workstation, and then he came back at 1.03pm. At 2.04pm he saved his current state once again and it shows that he had made a significant change to the way rotation was being calculated. During that time, Care had removed the discRotation and wheelRotation dependencies and replaced them with a triggered procedure (as defined in Figure 6.6(c)). The reason for this change is because Care noticed that when the disc was moving very slowly, dragging the wheel across the surface of the disc caused an unexpected rotation of the wheel. If the disc was stationary, then dragging the wheel across it should result in no rotation of the wheel, but this was not the case in the model. The change that Care made to his model was due to the realisation that the rotation of the wheel must be calculated cumulatively, by adding the current wheel speed (Δz) to the wheel rotation at each increment of unit time (hence the use of a triggered procedure running on every clock tick). This change produced a noticeably different behaviour (two videos demonstrating the before and after behaviour can be found in [Har07b]).

Care remembers this moment well and describes it as when he realised the flaw in his conception of the way rotation is calculated on the wheel. However, I do not think that it was particularly a flaw in his model because at that time he had been following mathematical explanations based on those described in Figure 6.5. His model did accurately model the mathematical description from the book. The problem was that the book did not fully represent the subtleties of the physical mechanism. That is, there was no mention in the mathematical description that there was a temporal element to disc speed, that the disc speed would vary over time, and that this would clearly effect the cumulative wheel rotation. Perhaps it was not possible to represent accurately in mathematics the precise nature of interaction between the disc and the wheel—there was a necessary 'agency' element. It was only through building a model,

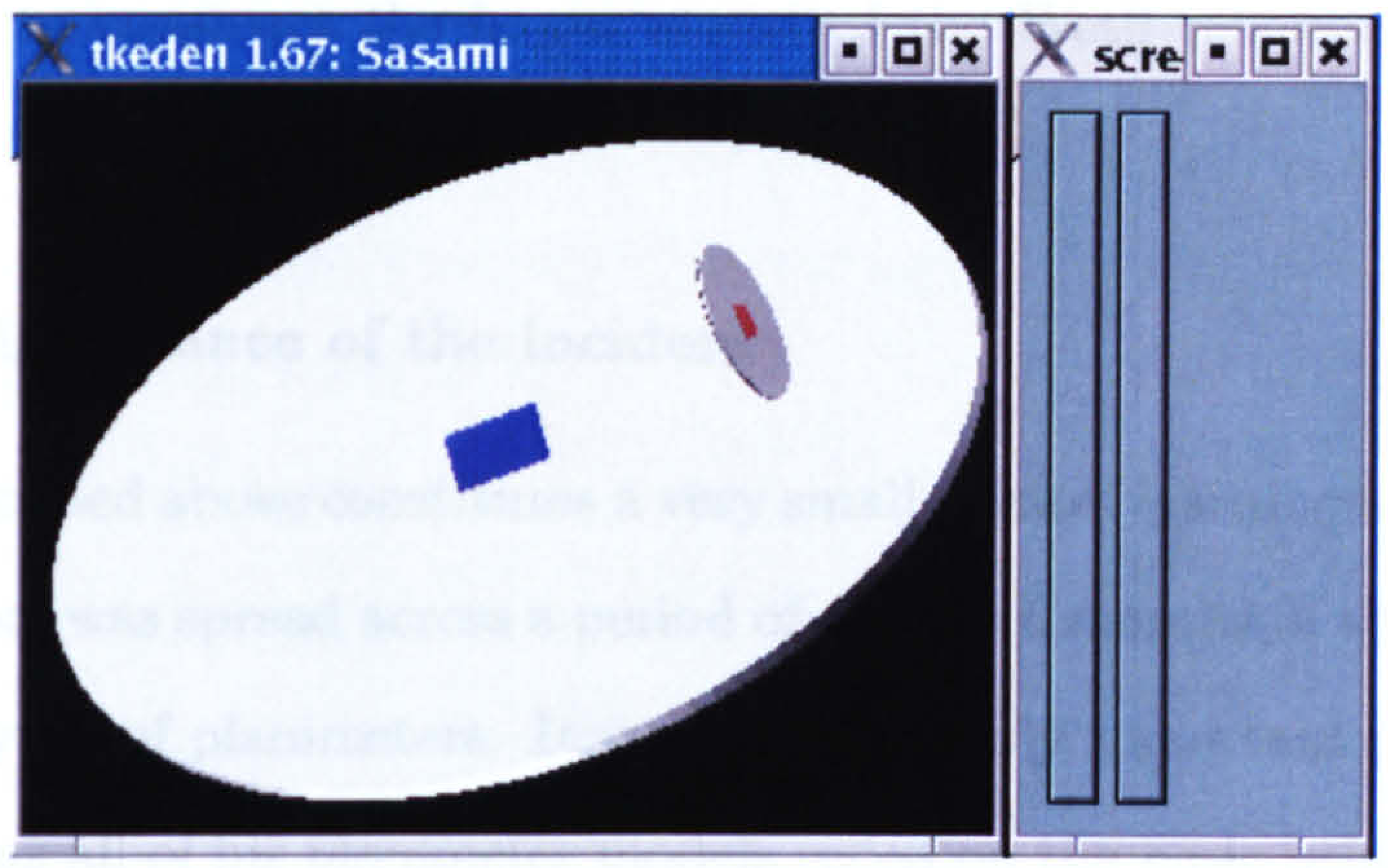



```
## The disc is rotating a constant speed
discSpeed = 1.0;

## The speed of the wheel is determined
## by the position of the wheel
wheelPos is discRadius*wheelRatio;
wheelSpeed is wheelPos*discSpeed;

## The amount of cumulative rotation on
## the disc and wheel can be calculated
## over time
discRotation is timeCount*discSpeed;
wheelRotation is timeCount*wheelSpeed;
```

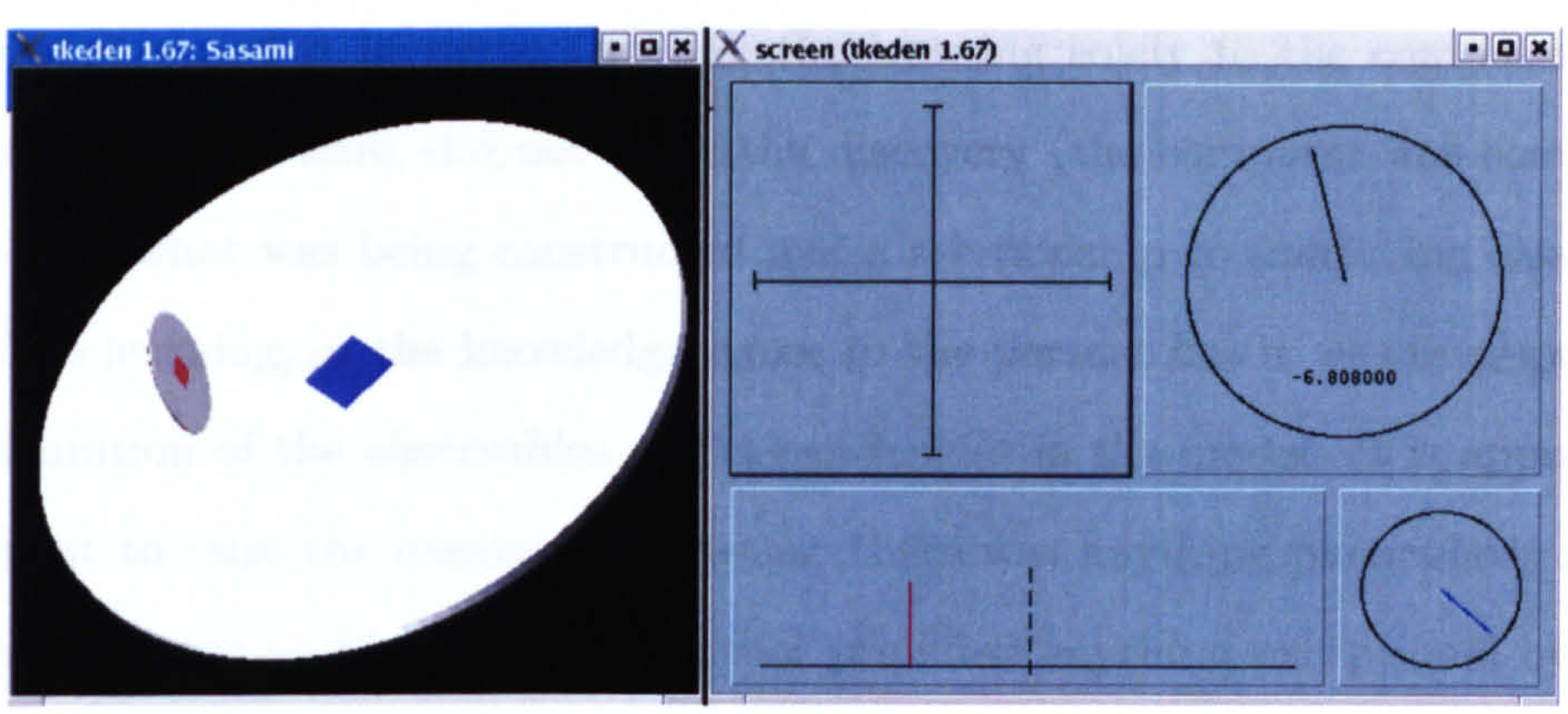
(a): The disc and wheel model on 24/11/2003 with relevant dependencies.



(b): The disc and wheel model on 25/11/2003 with two mouse sensitive bars to control the position of the wheel (wheelPos) and the speed of the disc (discSpeed).

```
proc clock: timeCount {
  discRotation = discRotation+discSpeed;
  wheelRotation = wheelRotation+wheelSpeed;
   todo("timeCount=timeCount+1;");
};
```

(c): A part of the script on 25/11/2003 after replacing the rotation dependencies in (a) with a triggered procedure for calculating the cumulative rotation.



(d): The disc and wheel model on 27/11/2003 with an interface for observing each part of the disc and wheel behaviour.

Figure 6.6: Stages in the disc & wheel model from 24/11/2003–27/11/2003.

and reflecting on the dependency and agency together, that a deeper understanding of the mechanical device was able to arise.

The modelling activity continued after the described incident, and Care went on to progressively build on the interface during the 26th & 27th November. Figure 6.6(d) represents the model on the 27th by which time Care had improved the mouse sensitive window and added two dimensional views of the disc and the wheel to closely observe their state. This model is used to describe the integration mechanism in his project report [Car04b]. The modelling process did not end here though, the disc and wheel model was used to construct the larger model of the Wetli planimeter as shown in Figure 6.4.

6.2.1.2 The significance of the incident

The incident described above constitutes a very small piece of learning when considering that Care's project was spread across a period of at least 6 months in which he explored many different types of planimeters. It was a particularly important piece of learning for Care though as all of his planimeter models relied on the basic principle of the disc and wheel for integration.

It is difficult to say precisely what caused the learning in this small incident. The evidence shows that Care's knowledge of the planimeter goes deeper than the information given in any book or other sources that he had encountered. His overall knowledge of planimeters was clearly heavily affected by the information given in explicit sources, but in this particular incident, it is difficult to link it directly to a source of information. Neither would it be correct to link the learning solely to the computer. The computer, or the software, did not make the discovery (the computer was completely unaware that what was being constructed had a relationship to something else in the world). The learning, or the knowledge, arose in the person, but it was in response to the configuration of the observables and dependencies in the model. It is appropriate at this point to raise the question of whether there was anything particularly special in the environment or whether the EM tool provided anything exceptional over any other environment. It is true, that the learning is not singularly the result of EM. However, Care's incident demonstrates that observables and dependencies, including the open environment in which different configurations of *observables* and *dependen-*

cies can be explored, can support the development of understanding that goes beyond explicit knowledge as digested information. The open-ended nature of the environment allows *agency*—making and attributing state changes—to play an important role in developing understanding. There are limitations in the mathematical descriptions of planimeters from textbooks, such as [Asp90]. Polyani views this as explicit knowledge and acknowledges that 'tacit knowing' is a deeper understanding that is difficult to describe in language [Pol62]. One solution to developing a deeper understanding is to construct an external model, as in constructionism. Papert's constructionism [PH91] is not concerned how the model or program is developed. EM specifically offers principles for constructing models (with observables, dependency and agency) that accord with experiential learning and tacit knowing in a way that differs from traditional model-building and programming techniques.

Care's interactions with the disc and wheel model show on a micro-level the way in which combined model construction and use can promote the development of a knowledge of the workings of planimeters. Furthermore, it has been observed that the knowledge of planimeters gained by Care is quite different from the book knowledge that is taught on history of computing courses. I have claimed that Care's experience is probably better described as similar to an engineer engaged in practical use and experimental construction of planimeters. This is supported by the idea that significant learning takes place when engaged in activities linked to a concrete situation.

6.2.2 Learning about ant navigation

The second project to be considered is about ant navigation and was undertaken by Daniel Keer as a final year project [Kee05]. The model, as shown in Figure 6.7, can be found in the EM archive [EMP:antnavigationKeer2005]. The aim of this project was to understand the method which a particular species of desert ants (genus cataglyphis) use to find food. Previous entymology research by Collett suggests that these ants, compared to other species, have relatively accurate colour vision and are able to recall and match images that they have seen before [CDGW92]. Ants, as well as other insects and animals, have a 'path integration' sense (a construal of the position of their nest) that enables them to relocate their nest after a long meandering forage for food. Collett has found that desert ants are able to maintain an accurate construal of their position

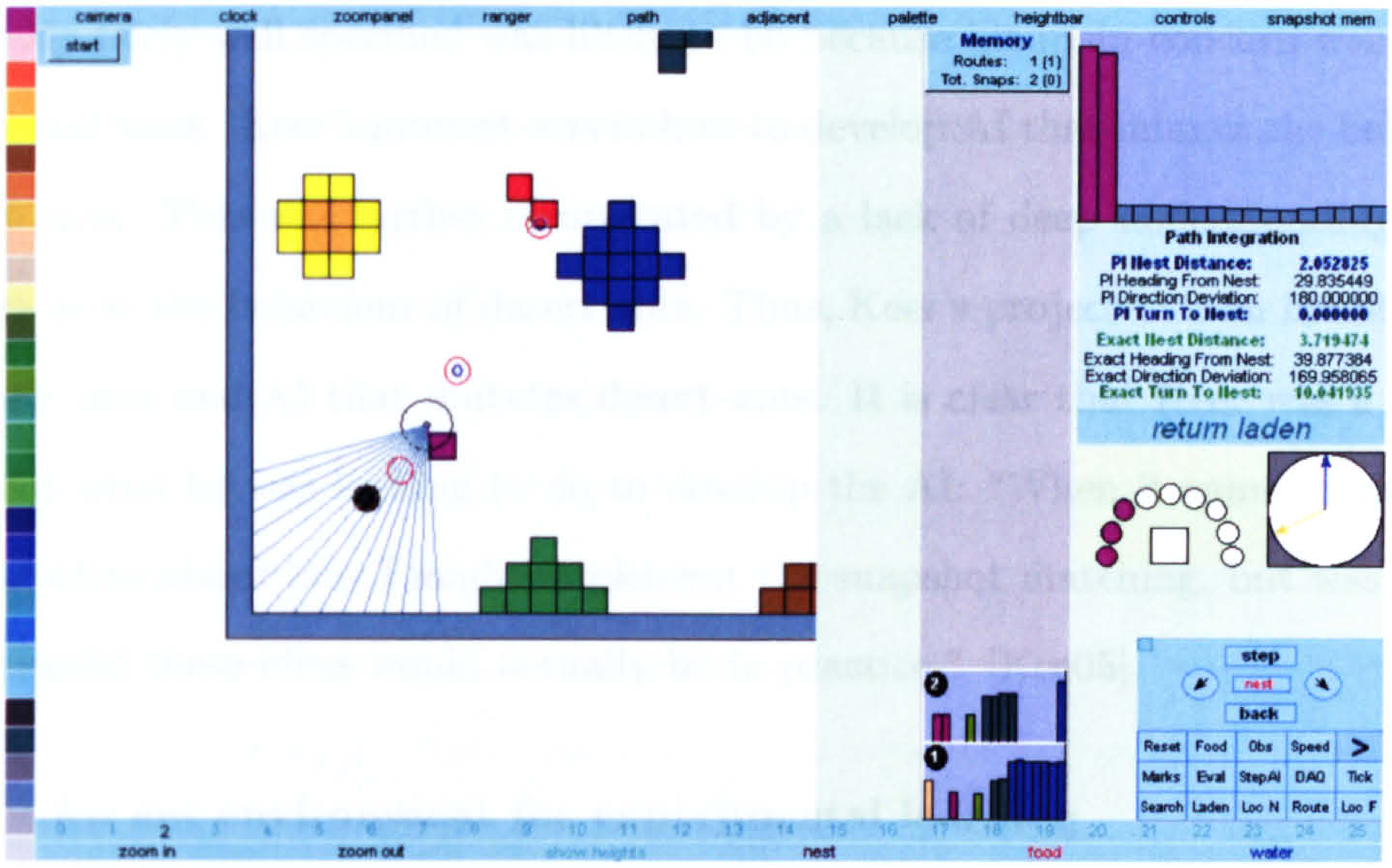


Figure 6.7: The ant navigation model by Keer.

on food runs of up to 200 metres [CDGW92]. He also suggests that when an ant has located a food source, it is able to remember a list of snapshot views on returning to its nest that allows it to find the food a second time [CDGW92].

6.2.2.1 Modelling process

Whereas the planimeter model was constructed from scratch, Keer's project was based around an existing ant model. Previously, K.C. Tan developed a simple model of an ant moving around a square space containing blocks [Tan99] (first in the Maintainer of Dynamic Dependencies (MoDD) tool, but later in `tkeden`). Tan's model [EMP:antsTan2004] formed the basis of Keer's ant navigation model, allowing Keer to begin experimenting with ant navigation at an early stage. This demonstrates the benefit of model reuse, that experimentation in the domain can begin at early stage if an existing model fits the needs of the model-builder.

Although Keer was able to utilise Tan's ant model, it was heavily extended during the modelling process to create a suitable environment for exploring ant navigation. Keer considers the initial part of the project to prepare the environment as the most straight-forward: "For the environment/interface, I had a clear plan of what was necessary, and developed [Tan's] model accordingly." Contrast this with the planimeter model where the modelling process was relatively open at the beginning in terms of how to develop the model. The possible reason for Keer working with an initial model that was already fairly well specified was likely to be because his main concern was for the experimental work. Keer's interest was in how to develop AI that mimics the behaviour of desert ants. This was further complicated by a lack of deep understanding in the literature as to the behaviour of desert ants. Thus, Keer's project was an investigation into desert ants and AI that imitates desert ants. It is clear that Keer was a lot less sure about what he was looking to do to develop the AI: "When it came to the AI, I had some ideas about how I might implement the snapshot matching, but was unsure how successful these ideas would actually be in practice." [Kee05].

6.2.2.2 An ant environment for experimental learning

To understand EM's potential for supporting the type of experimentation that Keer undertook, it is worth considering deeper issues regarding this type of learning. Beynon & Russ describe the usual type of experimentation supported by computers as being associated with "a stable objective context of observation in which parameters can be changed and the outcomes observed" [BR07]. Whereas, traditionally, usual scientific experimentation is more typically concerned with "identifying appropriate contexts for reliable observation, distinguishing between essential and accidental features of interaction, deciding what is deemed to be an outcome and what is deemed to have significant implications for this outcome" [RB07]. It is this type of exploratory sense-making activity that they call 'pre-theory' experimentation [RB07]. Such thinking is aligned with Gooding, a philosopher of science, who explains, in his book on 'Experiment and the making of meaning', that Faraday's knowledge of electromagnetism evolved through pre-theory experimentation [Goo90]. Gooding interprets that Faraday did not perform experiments to explain some aspect of reality but to find out what the reality was. Faraday developed experiments with his own artefacts and using his own procedures, and it is these 'construals' that contributed most to the science of electromagnetism [Goo90].

While other examples in this thesis may evoke an image of post-theory experimentation (i.e. interactions with jugs), the ant navigation project undertaken by Keer has a distinct element of pre-theory experimentation. Keer had a rough idea for the kind of interface that might be needed for experimentation, in a similar way that Faraday may have had a rough idea of some of the artefacts he needed for understanding phenomena relating to electromagnetism. However, when it came to the procedures that Keer used for experimenting, these were very much influenced by the situation and could not have been preconceived beforehand. For example, Keer experimented with different ideas in the beginning to see what insights could be gained. In particular, he found that new ideas could be generated "by moving the ant myself and seeing what she could perceive of the environment" [Kee05]. Keer found that he "built up the complete working AI through this interaction and experimentation with the model" [Kee05].

As highlighted by Beynon & Russ, human engagement plays a central role in pre-theory experiment [RB07]. Keer confirms this sentiment: "Experimenting with the model from the 'point of view' of the ant was vital for generating ideas about how the ant could compare snapshots with current surroundings." Interaction such as this appears to be crucial to developing understanding and learning as Keer goes on to say, "I think it would be very difficult to develop such an AI without 'playing' with the model to generate such insights" [Kee05].

Gooding's account of scientific discovery emphasises the role played by developing artefacts and procedures for experimenting during the experiment itself. Although Keer developed some of the ant environment before experimentation began, there were some parts of the environment which developed during experimentation. Simple additions such as buttons for acting out a specific element of agency, like moving the ant to the nest, were added on-the-fly. The potential to change the construction environment whilst constructing is evident in the EM Presentation Environment discussed in §4.2.2.2 and §5.2, when changes were made to the underlying notation in response to newly identified needs in the current model (i.e. "in the stream of thought").

Beynon & Russ emphasise that while the traditional specification-led approach to programming is not well-suited to pre-theory experimentation, the EM approach has the right ingredients due to the close relationship with observation and experiment [RB07]. Furthermore, EM artefacts such as the ant navigation model have much in common with 'construals' in Gooding's sense [RB07]. Keer endorses these affinities in his conclusion that "the Empirical Modelling 'experimental' approach has significant merit for AI development." [Kee05].

6.2.2.3 Differences between ant navigation and planimeter modelling

In both the planimeter project and the ant navigation project, it should be clear from the above discussion that the students were learning about each of their respective domains (as well as improving their model-building skills). The nature of this learning is linked in that they were exploring areas unknown to themselves, but in terms of the areas themselves the nature of the learning is different. In the planimeter project it was the case that Care's referent in the world was a physical planimeter, something that (had he had one) he could have compared with his model in order to check it reliably followed its behaviour. A planimeter, having been designed and engineered, has a clear behaviour. When the problem with the corkscrewing wheel arose in the EM model, it was a discovered because the behaviour did not accord with the physical planimeter. In the ant navigation project Keer had a quite different task in that he was attempting to model a phenomena whose behaviour is not well understood. His exploration of building a model was experimenting to confirm possible theories, or even describe new theories, about the plausibility of desert ants navigating in particular ways.

6.2.2.4 Implications for learning

Not only is Keer using EM to construct a model of ant navigation, but he is also using EM to construct an environment for exploring and learning about the nature of ant navigation. A traditional educational technology approach to this would force the construction of the environment first, and later a learner could use the environment to explore ant navigation. This can be called post-theory experimentation. EM allows learners to be much less constrained by a specification (if indeed there is any specification at all), and enables learners to follow their own lines of inquiry, by considering all artefacts and procedures—observables and dependencies—in the environment to be open to change by the learner during any experiment. Thus, EM can be considered as supporting pre-theory experimentation.

6.3 The basis for further empirical work

The coursework for the 'Introduction to EM' module shows the general applicability of EM for learning in a wide range of subjects and domains. It offers evidence on a macro-level that EM has potential in domains that have yet to be explored in any detail. The third year projects in planimeters and ant navigation examined particular examples of learning through EM in a specific domain. These studies show on a micro-level that EM has potential for facilitating learning that is different to, and complements, the learning from language-based sources (whether they be written as in books, or spoken as in teacher/classroom sources).

Chapter 7

Constructivist computing†

The term constructivism does not occur with great frequency in this thesis before the current chapter, but many of the discussions are closely related to issues in constructivism. The aim of this chapter is to introduce some of the relevant constructivist literature and to show how this relates to the practice of EM. Rather than take a stance on any particular decomposition of constructivism, the idea of constructivism is taken as a complete attitude or approach, as supported by Bruno Latour in his attempt to rescue constructivism [Lat03]. EM will be shown as supporting the five guarantees sought by Latour. This has led Beynon to propose that the EM approach be referred to as a constructivist approach to computing, or 'constructivist computing' [Bey07c].

Discussions on and relating to constructivism in the educational literature are vast and seem to be ever expanding with the addition of new variations of constructivism as pointed out by Phillips in his paper "The good, the bad, and the ugly: the many faces of constructivism" [Phi95]. Phillips compares constructivism to a secular religion because it has many sects "each of which harbors some distrust of its rivals" [Phi95]. Educational theorists and practitioners, sociologists and scientists, have all popularised and criticised constructivism as well as interpreting it differently according to the specific needs of their area. It has been applied so broadly that to talk of constructivism is to enter into the debate of 'which constructivism?' and justification is sought for the adoption of a particular form. Constructivism has been popular in the area of computing and educational technology as can be judged, for example, by the research relating to Papert's 'constructionism' [PH91] (as described in §1.2.1). Such widespread

†This chapter is an extended exposition of part of a joint paper published in the Journal of Computers [BH07].

application (and mis-application) of the term 'constructivism' has led many people to question its status as a prominent theory of learning. The title of Phillips' paper—the good, the bad and the ugly—reflects the changing mood surrounding constructivism, with Phillips suggesting that currently we might be experiencing the 'ugly' side. However, as Phillips notes, ideas closely related to constructivism continue to be the subject of vigorous debate [Phi95].

The application of EM to education and learning can be seen as having much in common with constructivism (see for example the discussion by Roe [Roe03:c.4]). However, previous connections with constructivism have been criticised for not being clearly aligned to a particular group of constructivists. In a book entitled "Constructivism for education" [SG95], Steffe highlights some of the conflicting epistemologies discussed by different groups of constructivists. She also stresses the need for research relating to constructivism to be aware of these conflicts [Ste95]. One solution to this problem is to join a particular group of constructivists and adopt a specific attitude to constructivism (e.g. radical constructivism). Another solution is to neglect the connection with constructivism altogether and join the critics of 'ugly' constructivism. Neither of these solutions appear to offer any contribution towards understanding EM with respect to learning.

7.1 Introducing constructivism

7.1.1 The constructivist idea

According to the well-known constructivist von Glasersfeld, the basic tenet of constructivism is that individuals are not born into the world pre-installed with rules, mechanisms, skills, and knowledge [Gla90]—individuals are unlike a computer that comes installed with software and ready-loaded with data. Neither do individuals acquire skills and knowledge in neatly packaged boxes by some kind of transfer—unlike the way a program or files are installed into a computer. Instead, the majority (the amount depends on which constructivism you follow) of knowledge is constructed by each individual in response to personal experiences. Piaget is generally credited with first formalising this way of thinking about how we construct knowledge from his studies of children [Pia29] and his essays on biology and knowledge [Pia71]. He suggested that learners internalise knowledge in two ways: assimilation and accommodation. When a learner assimilates, she is interpreting events in terms of her own understanding, fitting the experience to the mental model, thus strengthening her understanding of the experience [Pia71]. When a learner accommodates, she is changing her understanding of the world to accommodate the event, fitting the mental model to the experience, thus learning something new from the experience [Pia71]. This implies that the knower, the known, and the outside world, all can play a part in any construction or reconstruction of knowledge within the learner (and the interplay of these influences is also important, as described further by Latour [Lat03]).

EM is not concerned with the transfer of 'prescribed' knowledge to a 'passive' individual. The aim of this work is to address a more fundamental issue that is close to the constructivist idea: How can EM support the active personal construction of knowledge by model-building? Sense-making plays an important role in model-building with EM. Likewise, Piaget typically depicts the child learner as an individual creative scientist struggling to make sense of phenomena in the world [Pia29]. EM is offering support for learners in this role.

7.1.2 The many forms of constructivism

In this section, some of the many forms of constructivism are reviewed. Ernst von Glasersfeld's *radical constructivism* [Gla90] is perhaps the most famous form of constructivism, and is seen as the most extreme. He states the basic principles as:

> *1. Knowledge is not passively received either through the senses or by way of communication. Knowledge is actively built up by the cognizing subject. 2: a. The function of cognition is adaptive, in the biological sense of the term, tending towards fit or viability; b. Cognition serves the subject's organization of the experiential world, not the discovery of an objective ontological reality.* [Gla90]

Von Glasersfeld goes on to argue that to adopt constructivism seriously, the principles have to be radical because they are "incompatible with the traditional notions of knowledge, truth, and objectivity" and therefore require "a radical reconstruction of one's concept of reality." [Gla90].

The emphasis is placed on the individual as the constructor by von Glasersfeld. The effects the social environment have on the construction are not seen as being different from the effects from the environment taken as a whole [Gla90]. In Ernst's article on radical constructivism [Ern95], he discusses seven other forms of constructivism that place emphasis on different factors in the environment as playing a crucial role in knowledge construction.

As described by Galloway in article on Lev Vygotsky, *social constructivism* is based on Vygotsky's findings in developmental psychology that social context plays a central role in the process of making meaning [Gal01]. In Vygotsky's Zone of Proximal Development (ZPD) theory [Vyg78], the individual has a set of already acquired skills (things that can be done on one's own) and a set of skills yet to be acquired (things that cannot be done even with help). In between these two groups is a set of skills that can be done with some help, and Vygotsky calls this the Zone of Proximal Development. Learning or development is the process by which things move from the ZPD to the set of acquired skills, a process which needs a More Knowledgeable Other (MKO) [Gal01]. As development progresses, new skills yet to be acquired move into the ZPD enabling the individual to continue to learn at an ever greater depth.

The basis of social constructivism is the idea that individuals, being inherently social beings, participate in the learning of a collective knowledge. The relationship is reciprocal in that individuals learn from the collective, just as the collective is constructed by the social group of individuals. Stronger forms of social constructivism view all knowledge as socially constructed—great scientific theories are the constructive work of groups of individuals comparable to great works of literature, and classroom textbooks do not contain the absolute truth but are a representation of the knowledge of many individuals. Such views are expressed in a book by Latour controversially entitled *Laboratory life: The social construction of scientific facts* [LW79].

In another form, Vanderstraeten describes Dewey's philosophy as 'transactional constructivism' [Van02], referring to learning as a process of interaction (or transaction) between organism and environment. Vanderstraeten argues that Dewey's approach is constructivist because it implies that knowledge is not external, mind or organism independent, but is a relationship between actions and their results.

In a survey of the varieties of constructivism, Dougiamas builds on the well-known forms of constructivism to introduce other forms [Dou98]. After social constructivism he introduces cultural constructivism, which builds on social constructivism by emphasising the role of culture in constructivist learning [Dou98]. Critical constructivism is another flavour [Dou98] further building on social and cultural constructivism. There are others.

Phillips attempts to bring some order and clarity to the 'which constructivism' debate by introducing three dimensions on which to locate the different forms of constructivism [Phi95]. The first dimension has at one extreme the constructivists who focus on individual understanding (such as Piaget) and at the other the constructivists who focus on public knowledge (social constructivists). The second dimension can be characterised roughly as "mind as master versus matter as master", where on one side is the view that when knowledge is constructed it is chiefly influenced by the knower (or knowers) and on the other side is the view that it is imposed by nature (outside influence). The third dimension is related to the nature of the knowledge construction activity; at one extreme the construction activity can be described in terms of individual cognition, and at the other extreme the activity can be seen in terms of social processes. Phillips maintains that all flavours of constructivism can be located on these three dimensions [Phi95].

As explain earlier, there is pressure in the educationalist community to be aligned with a particular constructivist camp. For example, an article by Confrey [Con95] (in the same book as Steffe's discussion on the problems of constructivism [Ste95] mentioned earlier) expresses serious doubts about the compatibility of radical constructivism and social constructivism, even going as far as to say that the underlying theories are contradictory. EM faces criticism for not claiming to be constructivist but not adopting a clear theoretical basis by aligning with any particular group of constructivists. However, as will be shown in the next section, EM can offer a unique perspective on constructivism that does not require association with a particular group of constructivists. Instead, the EM approach is supported by a theoretical basis provided by the philosophy of William James [Bey05a]. Following Latour's suggestion, the quality of EM as an approach to constructivism can be determined, not by how well aligned EM is to a particular flavour, but by how well it strengthens the core constructivist idea [Lat03].

7.2 An EM perspective on constructivism

The principles of EM have been strengthened by the connection Beynon makes with a philosophic attitude, termed Radical Empiricism, developed by William James at the beginning of the 20th century [Bey05a]. Both EM and RE promote an attitude towards the nature of knowing that is radically different from typical approaches to knowledge representation in computing [Bey05a]. The close relationship between EM and Radical Empiricism is highly relevant to, and offers some fodder for, a discussion of constructivism from an EM perspective.

7.2.1 Introducing Radical Empiricism and pure experience

Radical Empiricism (RE) and EM are connected through their shared interest in rooting knowledge in personal experience [Bey05a]. EM as an approach to learning focuses on personal exploration, sense-making and individual understanding, which leads on to the argument in this thesis that the three key characteristics of EM activity are experimentation, flexibility and meaning. Beynon describes the important role of experience in EM activity:

> "The product of an EM exercise is first and foremost to be regarded as a source of experience whose interpretation by the modeller is not preconceived, but is to be established in the mind of the modeller through an association between experience of the model and experience external to the model." [Bey05a]

The nature of EM activity, as described by Beynon, is partially evident pictorially in the illustration of the model-builder in relation to the artefact and the referent (see Figure 2.5 on page 39). One of the key points of EM is that there is a relationship between knowledge of the artefact and knowledge of the referent such that interaction with the artefact can inform knowledge of the referent and interaction with the referent can inform knowledge of the artefact. In essence, there is an intimate connection between experience of the artefact and experience of the referent.

As argued by Beynon, RE is useful for "developing a deeper understanding of the primitive concepts of EM" [Bey05a]. This is because the relation between one experience and another is fundamentally important to RE as it is important to the principles and practice of EM. James identifies RE with the notion that knowledge or knowing is made by relations between experiences, experiences that may arise from relations themselves [Jam12:p30]. This is how James describes "an experience that knows another" [Jam12:p32]. For James, all aspects of knowledge or knowing can be put into experiential terms, and thus he refers to it as "a philosophy of pure experience" [Jam12:p45].

The importance of experienced relations to EM is evident when considering the substitution of an experience of the artefact for an experience of the referent. From a Jamesian "pure experience" standpoint, such a substitution is not only natural (in terms of knowing) but also beneficial: "By experimenting on our ideas of reality, we may save ourselves the trouble of experimenting on the real experiences which they severally mean." [Jam12:p32].

7.2.2 Reconciling RE with constructivism

Constructivism is generally expressed within a dualistic framework. The differences amongst constructivist approaches, as introduced in §7.1, are described by Phillips in terms of dualistic issues: focus on private versus focus on public; mind as matter versus mind as master; and construction as individual versus construction as social [Phi95]. Such opposing viewpoints are not easily reconcilable with a mainstream philosophic attitude.

In RE, James offers a non-dualistic pragmatic perspective on the nature of knowing. By adopting the idea of 'pure experience' there is a wide scope for what is classed as 'real': "the relations that connect experiences must themselves be experienced relations, and any kind of relation experienced must be accounted as 'real' as anything else" [Jam12:p22]. Private knowledge (criticised as potentially being within a dream world of no substance) and public knowledge (criticised as a socially constructed phenomena of no substance) can be experienced and therefore have equal status from an experiential perspective. The dualistic debate, common in constructivism [Phi95], on the status of the mind as either all powerful or subject to external forces of nature is irrelevant as experienced relations can be seen as caused by both internal states of consciousness [Jam12:p7] and external perceptions [Jam12:p28].

If constructivism were expressed in Jamesian terms then there might be fewer arguments amongst the opposing constructivist camps. By taking a RE attitude, the classification of constructivism, into Phillips's three dimensions for example, is an experienced relation. Whichever view of constructivism is observed, it is an experienced relation as 'real' or valid as any other. Given the large body of research on each of the many flavours of constructivism, there is clearly some benefit in the differentiations that are made between them. It seems quite natural, from an RE perspective, that scientists may wish to take a different view of constructivism from educationalists.

In an article that directly examines the relationship between RE and radical constructivism [Phi02], Phillips takes the view that RE is more *radical* in terms of its conception of the nature of knowing, whereas radical constructivism has a more mainstream (not so radical!) philosophical attitude. Equally though, Phillips believes that James would not have any difficulty with von Glasersfeld's suggestions for education implied by radical constructivism, so although they disagree on fundamentals, they can agree about the practical issues in education [Phi02].

7.2.3 Relevance for EM

As explained by Beynon [Bey05a], EM is built on the philosophy of RE from the initial idea that the development of knowledge is firmly rooted in personal experience, through to the precise details of James's attitude of 'pure experience' that all knowing and learning can be seen as a continuous stream or transition of 'experienced relations'. From this, the fundamentals of EM can be seen as radically different from the nature of knowing typically referred to in conventional approaches to knowledge representation in computing [Bey05a].

The significance of the connection between EM and RE can be appreciated when considering the potential of EM as an approach to constructivism. Due to RE's non-dualistic attitude, the classification of approaches to constructivism is just another experience, and therefore is not susceptible to the concerns surrounding which constructivist approach is 'correct' or 'better'. In following RE's attitude of 'pure experience', EM can likewise be seen as not primarily concerned with being associated with a particular constructivist camp.

There is still a need to show that EM is constructivist in spirit. The eight characteristics of EM, classified into experimental, flexible and meaningful characteristics, share many links with constructivism, from being concerned with 'learning through physical construction' to 'learning being a personal experience'. However, to evaluate EM as a possible constructivist approach, a stronger method for appraisal should be considered. For this it is helpful to refer to Latour later in this chapter.

7.3 Constructivism and Latour's guarantees

If Phillips attempts to draw out the differences and incoherence in the varieties of constructivist thought [Phi95] then Bruno Latour takes a different approach in emphasising their collective integrity for strengthening the constructivist idea [Lat03]. For Latour, the arguments around constructivism are a natural process in coming to understanding: "Everywhere, building, creating, constructing, laboring means *to learn how to come sensitive* to the contrary requirements, to exigencies, to the pressures of conflicting agencies where none of them is really in command." [Lat03]. Latour considers it a naturally constructivist tendency to be arguing about 'which constructivism'—where none really fit every situation.

Before Latour attempts to rescue constructivism, he explains some of the reasons why it has become a dangerous, and often despised, word. First Latour attacks the misinterpreted use of 'social' often prepended to 'constructivism'. He argues that the word is taken to mean that "the construction is made of social stuff ... a material so light that the slightest wind would dismantle it". Therefore opponents considered social constructivism not strong enough to build "the house of science ... made of solid walls of facts" [Lat03]. Second, Latour argues that the idea of construction in social science wrongly implies an all powerful maker or builder. Whereas in reality, "if there is one thing toward which 'making' does not lead, it is to the concept of a human actor fully in command ... the constructor has to share its agency with a sea of actants over which they have neither control nor mastery" [Lat03]. Third, Latour describes how the 'construction' metaphor has led to opponents incorrectly assuming that constructions can easily be de-constructed. He explains that the 'de-constructionists' are all too eager to see a construction as a sign of weakness and as something to be reduced to ruins "in order to give way to a better and firmer structure untouched by human hands".

Latour aims to rescue constructivism, from the three problems above, by drawing on the core idea of constructivism and showing that different approaches can all coherently agree with the core idea. In determining the extent to which the constructivism is achieved, approaches to constructivism can be evaluated in terms of how each strengthens these five guarantees, *taken together* [Lat03]:

1. Certain ideas or things should not be allowed to be disputed, and should be acknowledged as a stable reality.

2. A revision process should be maintained to make sure new claimants are able to challenge areas which the established order has not taken into account.

3. Understanding of the world is composed progressively, it is not already there once and for all.

4. It is essential that there is no clear separation between the human and non-human (words and worlds, nature and culture, facts and representation).

5. It should be possible to differentiate between good and bad construction, and specify the quality of the 'good common world'.

7.4 Modes of application for EM construals

In order to reflect on EM's constructivist ingredients, it is helpful to consider the contrast between a traditional program and an EM construal (or model). This is best appreciated by considering five different but interrelated modes of application:

- Realising an established construal
- Developing and critiquing a construal
- Exploring speculative construals
- Blending mind and machine in construals
- Auditing a construal

These modes of application will be briefly discussed in turn, and related to exercises in EM that have been associated with an extended study of Sudoku puzzles and their solution by a number of student authors over the last two years. Screen-shots taken from various models that have been developed are depicted in Figure 7.1 and Figure 7.2. Figure 7.1 shows a basic EM model of Sudoku puzzles that was developed by King

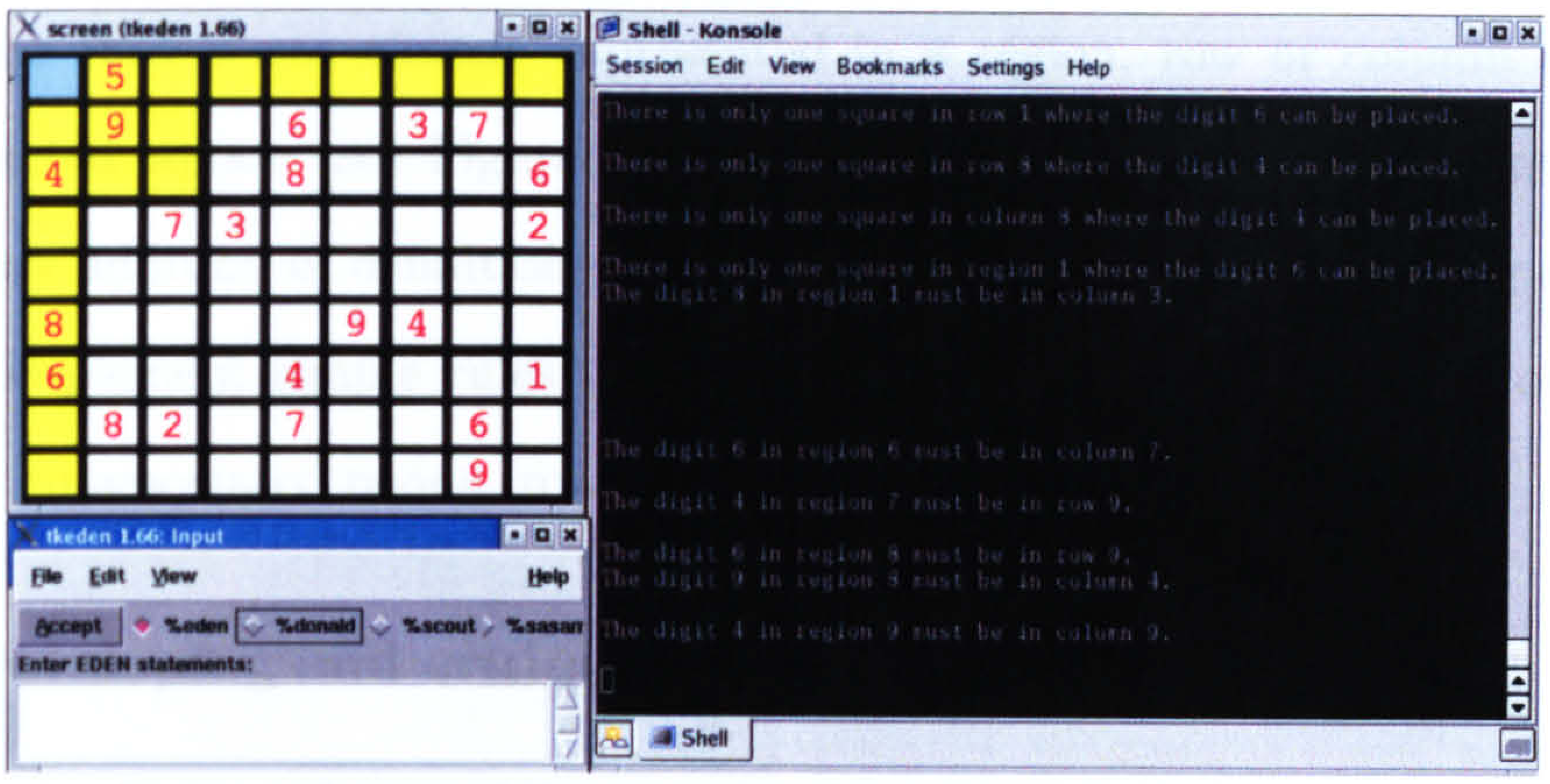


Figure 7.1: The original sudoku model by King [EMP:sudokuKing2007].

[EMP:sudokuKing2005]. Those depicted in Figure 7.2 (b) and (d) subsume King's original model simply by extending its basic set of definitions. In the next section the relevance of these modes of application will be formalised in comparison to Latour's guarantees.

7.4.1 Realising an established construal

EM can be conducted with an explicit referent and goal in mind. It may be that the nature of the relevant observables, dependencies and agencies at work in the application is well-accepted and understood. The objective of the modelling may be to achieve realism by some criterion: whether to be a good likeness or to fulfil a recognised function of an object.

In King's model [Kin07], dependency serves to maintain the relationships between observables that are characteristic of the Sudoku puzzle: for instance, ensuring that the displayed information about current possibilities is kept up-to-date when a new digit is entered. King's model is a 'vanilla' model whose primary function is to give book-keeping support to the manual solution of a puzzle. In its most primitive form, this entails being able to set up and store positions electronically, to record the sequence of steps and recover configurations. The state of the Sudoku puzzle can either be manipulated in a 'user/designer mode' through mouse interaction with the grid, or changed by typing new definitions into the interpreter input window. (The latter being the way in the model was originally constructed, and through which it remains wide open for further modification in much more radical ways.) The model is readily extended to perform functions that require automation; for instance, displaying the list of digits that is not already represented in a region, row or column associated with a selected square (see Figure 7.1). Introducing such functionality involves the discretionary addition of definitions to the 'vanilla' Sudoku model. Adding automatic agents to implement simple rules (such as entering a digit where every other digit appears in an associated region, row or column) is straightforward.

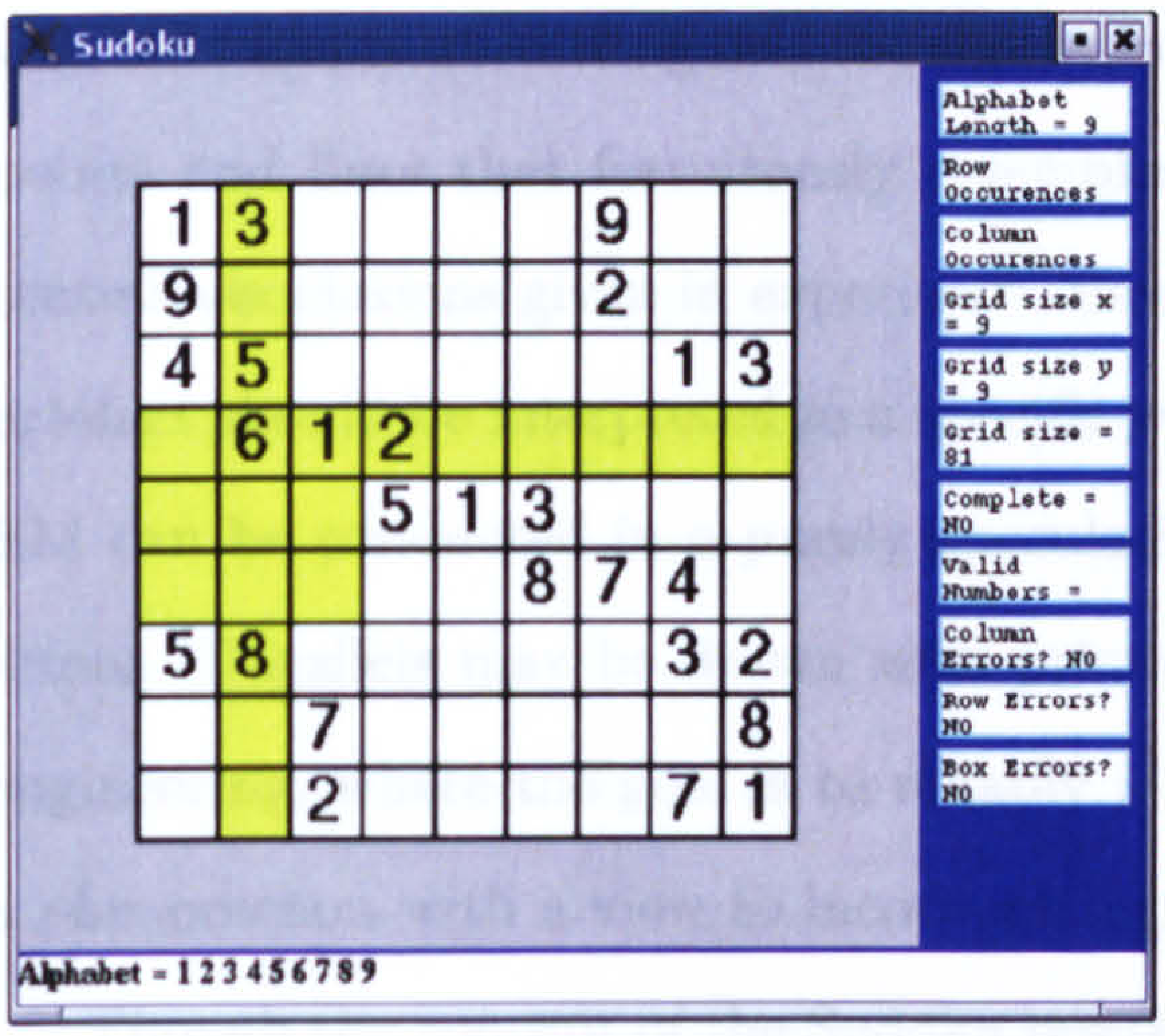
7.4.2 Developing and critiquing a construal

As construals, EM artefacts exhibit relationships between observables, dependencies and agents that embody a form of explanation. It may be that such explanations for the current state of affairs are uncontroversial, perhaps to the point of being seemingly beyond question. In some cases, certain features may be the very features that are deemed to define a referent. Nonetheless, building an EM artefact makes it possible to explore what might be termed "the neighbourhood of its referent in the space of sense" [Bey01]. This may entail adopting different viewpoints on the referent, probing accepted hypotheses and exposing alternative explanations. This is especially useful in a design context, where such investigation can lead to innovation.

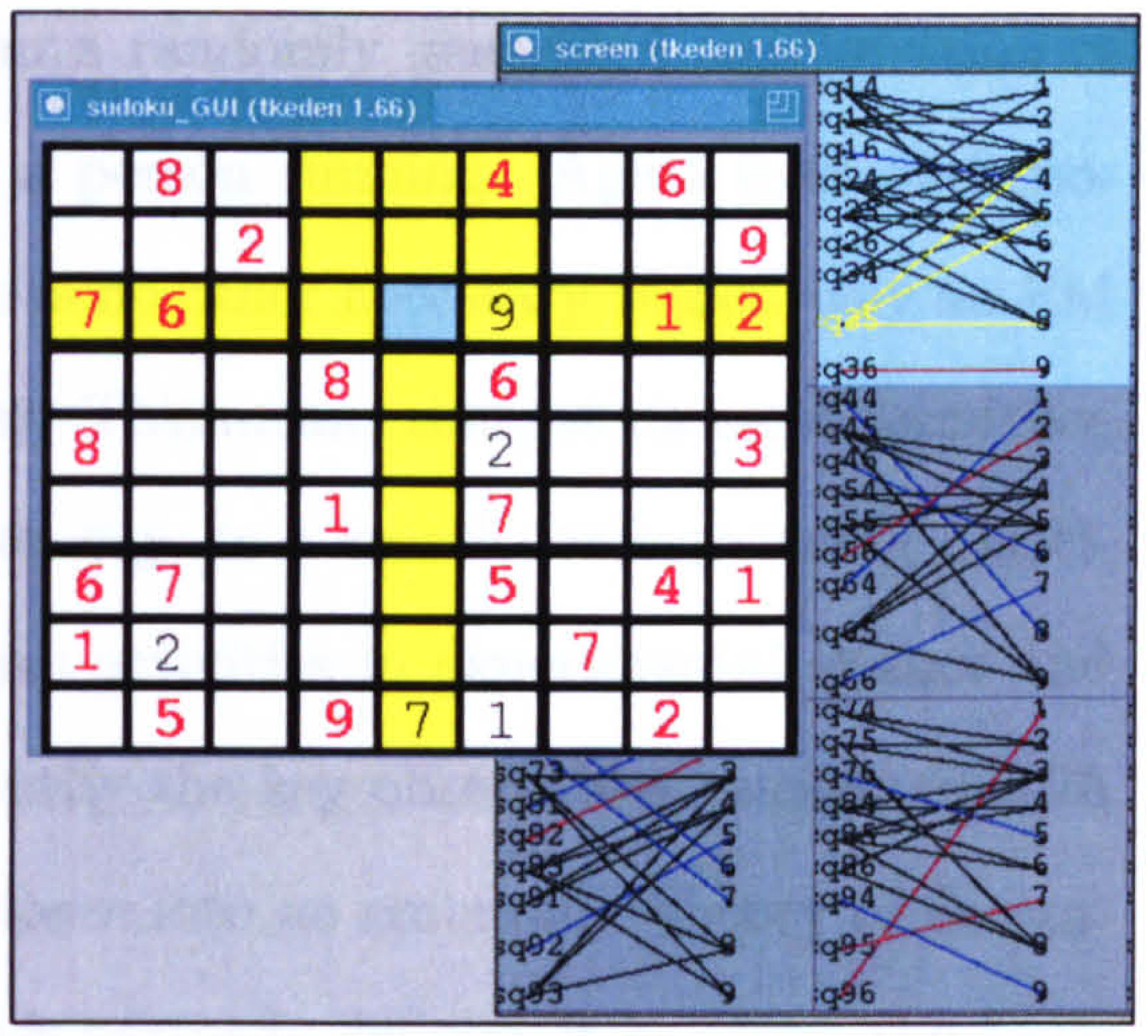
Rumsey's Doku builder (Figure 7.2(a)) demonstrates some of the potential for developing King's initial Sudoku construal. It allows a modeller to set up grids of different sizes and to generalise the principle of the Sudoku puzzle by using a different alphabet or modifying the solution constraints. Rumsey's model is oriented towards puzzle-building rather than solution, so that it deals with states of affairs of peripheral interest in the conventional model. For instance, the possibility of overwriting given entries and of processing states where there are conflicting entries is routinely considered. Note that the distinction between Rumsey's model and King's model is primarily concerned with how the modeller has exercised his discretion in developing and registering different paths through possible changes of state. For instance, it is quite possible to change the initially fixed digits through the graphical user-interface in King's model, but this is not consistent with the established construal of 'solving a Sudoku puzzle'.

7.4.3 Exploring speculative construals

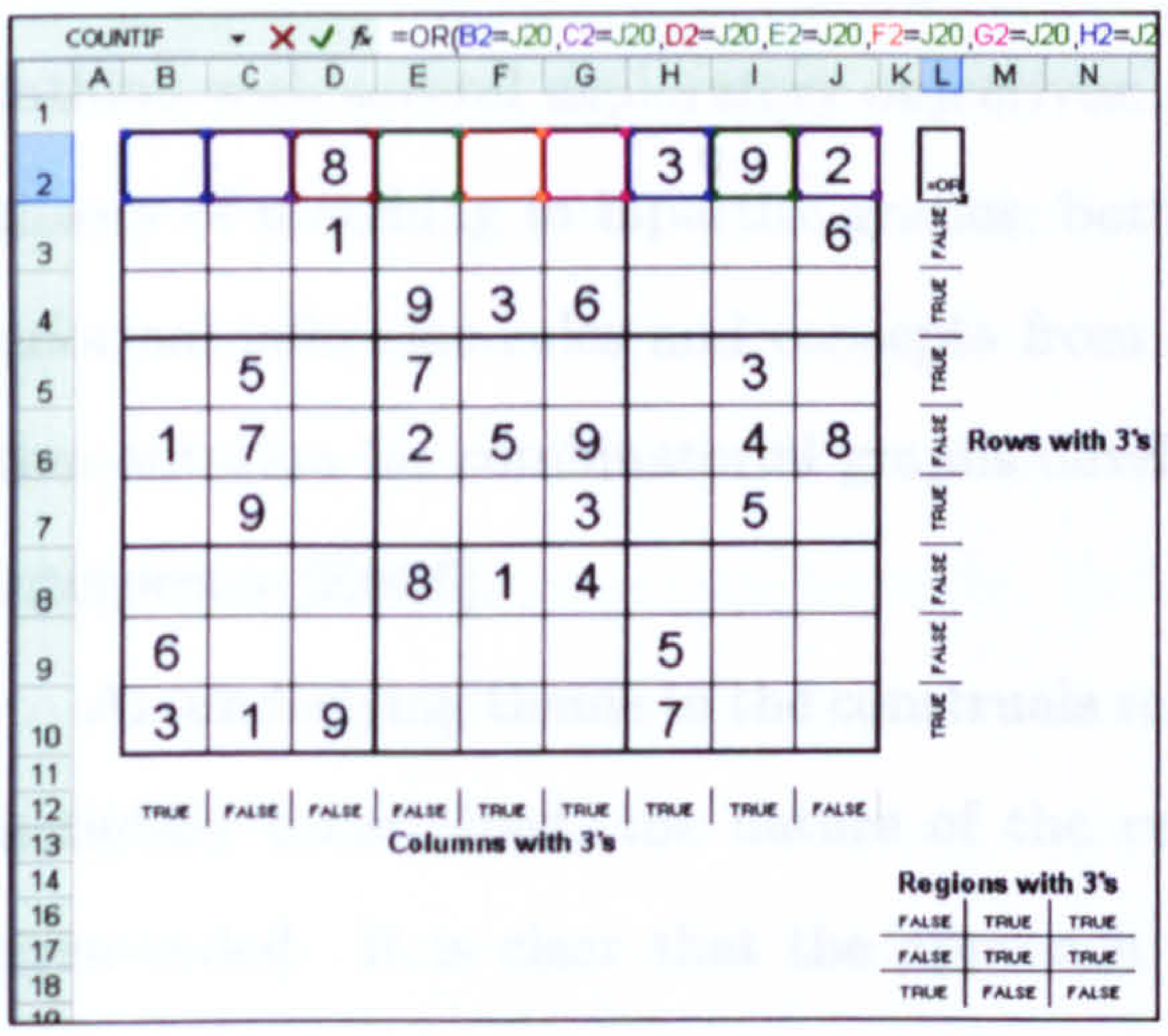
The characteristic primitive activity in EM is the construction of artefacts that exhibit patterns of dependency that invoke some external experience in the modeller's mind.



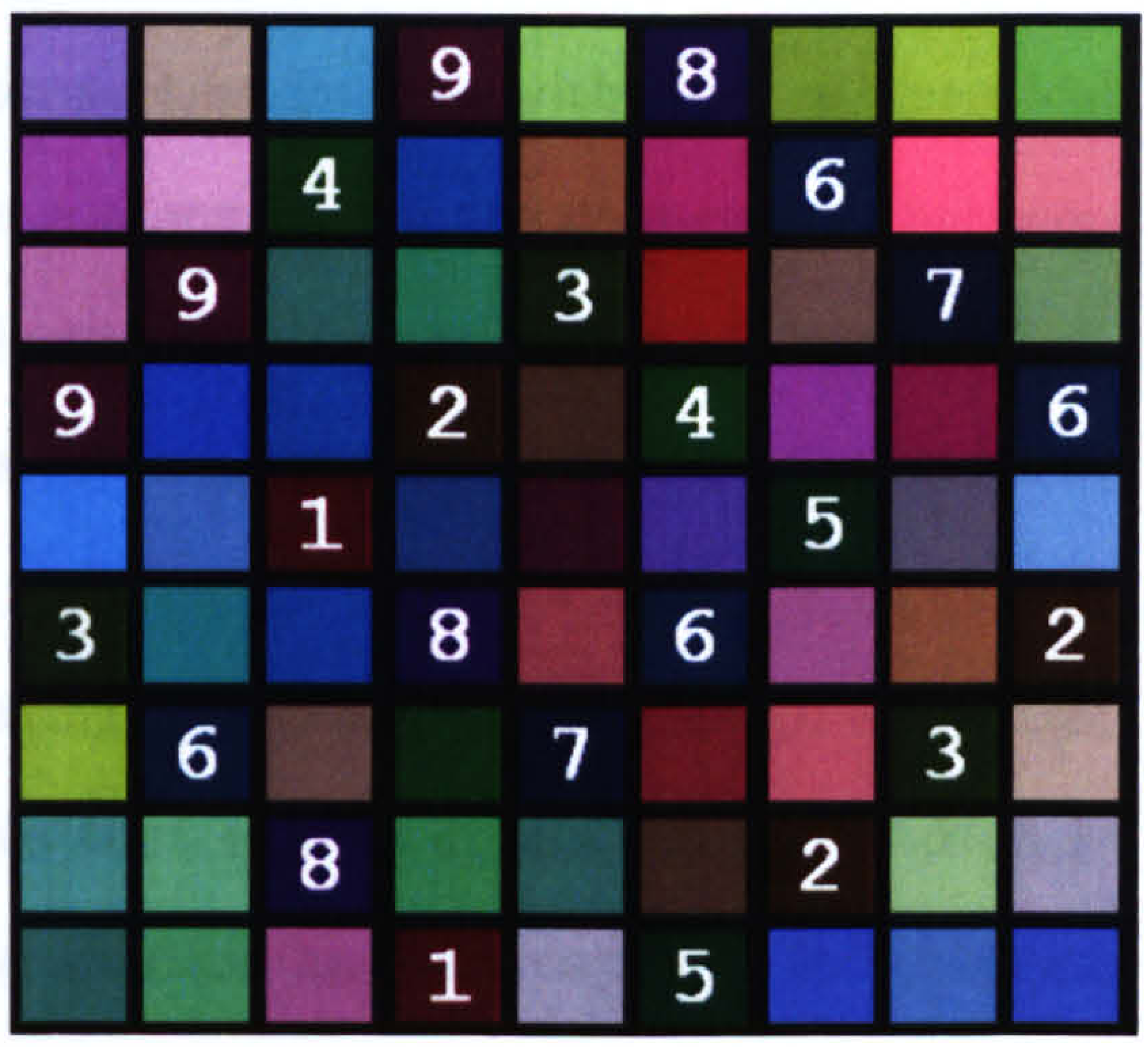
(a) The Doku builder for generalised Sudoku puzzles



(b) Linking Sudoku with combinatorial graphs



(c) Studying Sudoku solving with a spreadsheet



(d) Exploring solving strategies based on colour

Figure 7.2: EM construals based on or inspired by King's Sudoku model.

For instance, a few simple geometric objects, such as lines and circles, when appropriately animated, can invoke a person running. Though no animation may be strictly necessary for this association to be made, the distinctive semantic foundation for EM rests on the observation that EM artefacts admit open interactions that disclose semantically significant dependencies between the positions of geometric elements. The fact that in the geometric figure "the head" moves with "the body" in a characteristic way is evidence to distinguish the figure from a randomly generated configuration of points and lines that fortuitously resembles a person running. Apart from such potential associations given in experience, there is no other necessary reason why an EM artefact should be interpreted in a specific way. This means that, at its most primitive, EM can be conducted in a purely speculative way, as a search for convincing associations. Parallels may be drawn with primitive activities in experimental science and engineering, where the goal is to reliably identify the key observables associated with a phenomenon with a view to incorporating them into an embryonic theory or design.

Though the concept of the Sudoku puzzle is narrowly defined, it has been the focus of a number of open-ended investigations. Figure 7.2(b) is a model developed by Efstathiou with several exploratory objectives: connecting Sudoku with the mathematical theory of matching in bipartite graphs; better understanding the connections between informal inference rules and concepts from discrete mathematics; evaluating a definitive notation for combinatorial graphs developed as an extension to the standard EM interpreter [Efs06].

An underlying theme in the construals relating to Sudoku is that, though the puzzle is tightly constrained, the nature of the rules that can be applied in its solution is open-ended. It is clear that the approach to solution and the capacity to recognise rules differs from solver to solver. All strategies rest on being able to build upon simple self-evident observations about state to derive useful consequences. The Sudoku model in Figure 7.2(c) was developed (using a general spreadsheet environment as described in §3.4) in a speculative manner by the author with a view to better understanding how self-evident observations can cumulatively inform steps in solution. In this context, the associated dependencies were modelled using a conventional spreadsheet application. Elementary observations that can be made about the current state of the puzzle, such as 'row 1 contains 2, 3, 8 and 9', were maintained on separate layers of the spreadsheet, and significant inferences were subsequently derived by constructing other dependencies to link entries across several layers.

7.4.4 Blending mind and mechanism in construals

The concept of EM is predicated on human engagement, perception and interpretation. The notion of a dependency appeals to the idea that one change entails another in the view of a specific agent. The ground for such a notion of indivisibly coupled change is either the direct experience of the modeller ("pressing the switch puts the light on") or an experimentally informed construal that projects such coupling of change into the environment of an independent agent ("in the view of the engine management system the car is moving when the engine is running and the clutch is engaged"). That changes are perceived as indivisibly coupled has conceptual, physiological and technological components. On this account, sense-making in EM necessarily involves a blending of human and non-human agency. Some applications of EM may be primarily concerned with investigating this blending of aesthetic and experiential with mechanical and symbolic worlds.

The distinction between human and non-human perspectives on construal is highlighted in Sudoku by the fact that—in a well-posed puzzle—the precise content of any square can be inferred. In effect, the content of every square is logically dependent on the digits in the initial grid. There is no way that such dependency is directly mediated in the experience of the solver however. Even an experienced solver can only appreciate how a few specific inferences apply in a current situation to infer a new value in the grid. The boundary between what the machine automatically supplies by way of support for the solver and what the solver might be expected to observe can be adjusted by deploying different strategies for dealing with rules and inferences.

The colour variant of Sudoku depicted in Figure 7.2(d), developed by the author [EMP:coloursudokuHarfield2007], illustrates a subtle variation on this theme. A specific colour is associated with each digit, and the background colour for each blank square is a mixture of the colours associated with digits that do not already appear in the same region, column or row. The distribution of colours in the grid provides global information about the possible entries in locations that proves to be a valuable aid to solution. At each step, dark squares offer the best prospects for making a new entry.

A black square indicates that an error has been made. Being obliged to place a digit in a brightly coloured square suggests that a speculative step has been taken. Strategies for solution can be suggested by looking at the disposition of hues in regions, rows and columns. The process of choosing the colours to be associated with the nine digits was necessarily empirical in nature, and remains subject to further refinement. Subsequent extensions of this model allow the solver to manipulate the association of colours to digits and their luminance in a dynamic fashion, potentially enabling richer strategies.

7.4.5 Auditing a construal

EM is as much—or more—concerned with the processes of construction as with the product. McCarty, writing in the context of humanities computing [McC05], has emphasised the importance of modelling in helping us to appreciate 'how we know what we know'. The incremental construction of an EM artefact is associated with step-by-step empirical validation of how its states correspond to those of its referent. Where appropriate, these steps can be retraced in auditing a construal. The closely parallel role played by 'informal' artefacts in the exposition of a mathematical proof highlights the complexity and subtlety of the relationship between abstract propositions and the experiences that can convince us of their validity. The premise that underlies this aspect of EM is that espoused by William James in his philosophic attitude of Radical Empiricism: "Everything real must be experiencable somewhere, and every kind of thing experienced must somewhere be real" [Jam12:p160].

As the Sudoku exercises illustrate, EM activity broadly informs the quality of the construal from all the above perspectives. Problematic elements in a construal reveal themselves in interaction. They may be associated with imprecise or incorrect definitions in the construal itself. For instance, a mistake in defining the initial colour mix within the colour Sudoku model was disclosed when two squares that patently admitted the same possibilities had different colours. Significantly, such a problem could be resolved by redefining the colour mix function on-the-fly, then resuming the Sudoku solving activity without abandoning the stream of thought [Kin07]. The problems in developing a construal may also be attributable to the referent. For instance, Efstathiou's model led to the identification of a puzzle that was apparently not well-posed in that its solution required guessing and an extended back-tracking search. In this context, the way in which non-obvious facts about a state in solving a Sudoku puzzle can be inferred from simple self-evident observations can itself be viewed as an integral part of the EM activity. Note that EM offers no magic wand for conjuring dependencies; it only supplies the conceptual framework within which they can be most effectively exploited. This is illustrated by the difficulty of recovering from a mistaken step in Sudoku without exploiting access to information not directly accessible to experience.

There is no clear separation between the various modes in which EM construals are applied. As the Sudoku modelling exercises illustrate, many different interpretative aspects can be represented in one and the same activity, potentially simultaneously. A similar ambiguity arises in experiment, when what was first carried out with uncertain expectations of the outcome is routinely performed to confirm what—it thereafter seems—could hardly be otherwise.

7.5 EM and constructivism

The aim of this chapter is to demonstrate EM's capability for supporting the constructivist thought and practice. In Latour's vision for strengthening the core values of constructivism, he proposes that constructivist approaches be evaluated in terms of their ability to strengthen five guarantees. In each of the five subsections below, a guarantee is discussed with reference to construction using EM. According to Latour's criterion, the quality of EM as a constructivist approach can be gauged by how far it strengthens all five guarantees *when taken together* [Lat03].

7.5.1 Acknowledging a reality

The first of Latour's guarantees is that approaches to constructivism should acknowledge stable aspects of knowledge or reality where certain ideas, beliefs, and reasonings should not be allowed to be disputed. For Latour this means, "once there, and no matter how it came about, discussion about X should stop for good" [Lat03].

From a model-building point of view, it is essential that certain ingredients in a model should be accepted as stable (i.e. as a reality) in a model-builder's personal experience. During the construction of the Sudoku model, it was acknowledged that the rules of the game (i.e. played on a 9x9 grid, only one of each digit in each row, column and region) must remain stable in order for any further reasoning to take place. The activity of 'constructing a reality' involves the identification of stable patterns of interaction in terms of observables, dependencies and agent actions. Once constructed, further preliminary model development on top of the stable *observables, dependencies & agency* (ODA) can take place, and as regular patterns emerge the stable aspects of the model (in terms of ODA) may increase. The extent to which ODA are considered stable corresponds to the experience of the model-builder's interactions with the model in relation to the referent.

In terms of learning, acknowledging a reality is related to the characteristic that learning involves *exercising the familiar* aspects of our understanding (see §1.2.3). The reality is that which is deemed to be stable, that which is deemed to be familiar and fairly well understood. These familiar aspects of knowledge can be framed in terms of observables, dependencies and agent actions, in order to mediate between the model and the referent (see Figure 5.14). A further aspect to this is model reuse, where a model is considered stable it can effectively be taken as a basis for further model-building, by different model-builders and in different contexts. As discussed by Latour, a stable basis is needed to encourage 'builders' to make further constructions—without an acknowledged reality constructions are susceptible to the exploits of de-constructionists [Lat03].

7.5.2 Admitting the possibility of revision

Despite the definite stance on reality taken by the first guarantee, the second admits the possibility of revision. Latour specifies that "a revision process should be maintained, an appeal of some sort, to make sure that new claimants—which the former established order had not been able to take into account—will be able to have their voices heard" [Lat03].

Where EM and traditional programming may stand on a fairly equal footing in terms of constructing a reality, EM has many advantages over programming when it comes to the possibility of revision. As has been discussed in Chapter 3 and Chapter 4, one of EM's powers is that within a model 'everything is up for grabs' at any time during construction and use (although no distinction is made between the two—see §3.1.5). This open-ended support for revision of a construction guarantees that appeals to the established order of things can be taken into account.

For Latour, constructivism does not mean that everything constructed can be 'deconstructed to dust', but that when experiences do not fit into the established order it should be possible to revise ideas, beliefs and reasoning to take account of new findings. It is common for our conceptions of the world to be slightly different, often a conception is *good enough* to work (just as the colour model was good enough to work for solving Sudoku) and at other times a conception might not correspond with our expectations of the world. The latter case is when it is important to be able to revise ideas, beliefs and reasoning—not just for the sake of it—and it is when learning is most likely to occur. Latour states that this guarantee is a complement of the first, and in terms of learning it is related to exercising the familiar and examining experiences that do not fit the familiar, as introduced in the characteristic of 'learning results from realising the unknown' in §1.2.3.

7.5.3 Progressive composition of the common world

The third guarantee states that "the common world is to be composed progressively; it is not already there once and for all" [Lat03]. The emphasis in this guarantee is on "the unified world as a thing of the future, not of the past"

A learner constructing a model with EM can be compared to a scientist, constantly experimenting with phenomena leading to progressively deeper understanding. When the referent of a model is unclear or unstable in the reality of the model-builder, then a certain amount of experimentation must take place to develop a closer correspondence between the world (referent) and the model. EM environments encourage experimentation, as discussed in Chapter 2, because changes to observables, dependencies and agency are made on-the-fly and their effects can be immediately noticeable. Furthermore, as the model (and the understanding in the learner) is progressively composed, reliable patterns of ODA become stable realities from which further experimentation can take place.

The third guarantee insists that there is no one specific understanding of the common world, but that understanding is to be built up progressively through experimentation. The range of use of the Sudoku models shows that there is no one specific use for a model in terms of learning. EM supports the development of understanding through progressive model-building founded on experimentation. The flexible characteristics of EM discussed in Chapter 2 ('creating EM construals is not a pre-thought-out activity' and 'an EM construal is never considered finished') describe the open-ended nature of model-building in supporting learning that does not necessarily follow a prescribed path and does not lead to a pre-specified outcome.

7.5.4 The essential union of the human and the non-human

The next guarantee emphasises the importance of the inseparable interaction between human and non-human agencies in the process of construction. For Latour this means that approaches to constructivism should "ensure that there is no ... clear separation between words and worlds, nature and culture, facts and representation" and recognise that "humans and non-humans are engaged in a history that should render their separation impossible" [Lat03].

The immediate support that EM provides for Latour's fourth guarantee is that model-building is a personal activity, as discussed in the characteristic of 'an EM construal is personal to the model-builder' (see §2.2.6), that brings together the human and the non-human. Constructing a model involves the model-builder finding a correspondence between an experience in the common world and an experience with the model on the computer. EM necessitates a correspondence between the human model-builder and the non-human modelling environment, and between facts and representations. As can be seen by the number of students who prefer to start model-building from scratch, models often do not transfer easily to other human model-builders because they are part of a history that makes their separation difficult. Models are more special to their owners because they represent an acknowledged reality, a history of revisions, and a progression of experimentation.

As discussed in Chapter 2, EM supports learning that is motivated by personal interest, situated in a culture or context, and intimately connected with everyday experience. These three characteristics involve an essential correspondence between the learner, the referent, and the model (see Figure 2.5 on page 39). In this way, EM activity as characterised necessarily supports Latour's guarantee for respecting the inseparability of human and non-human agencies in the process of construction.

7.5.5 Differentiating between good and bad construction

The fifth and final guarantee ensures that constructivist approaches should enable the builders to appraise the quality of the construction. In Latour's words, "institutions assuring due process should be able to specify the quality of the 'good common world' they have to monitor" [Lat03].

EM as an approach to constructivism places the responsibility of auditing a model in the hands of the model-builder. Auditing constructions does not mean that at the end of the model-building exercise the model will be evaluated on its quality (although this may be relevant for education), auditing constructions is a process that goes on throughout the model-building where a model-builder is slowly developing an understanding of familiar interactions. At the beginning of a model-building exercise this might involve testing primitive dependencies, such as the observation of the number 9 in a Sudoku square. Later on, confidence in these primitive foundations may lead to stable interactions on a more complex level, such as the solving of the complete Sudoku puzzle. As discussed in section §2.2.4, EM is more concerned with the process than it is with the product. The quality of a model is constantly appraised by the model-builder with respect to how well it imitates elements of the referent that the model-builder's attention is drawn towards. The accessibility of ODA in a model make it possible to mediate between states in the computer and states in the world, and therefore the quality of a model can be assessed by comparing observed dependencies in the model with those in the world.

In terms of learning, Latour's fifth guarantee is related to the second—it is when the quality of a construction is considered that there is potential for revision to take place. The fifth guarantee also ensures that the realities acknowledged (by the first guarantee) are reasonable and sensible constructions. By ensuring the quality of constructions, approaches to constructivism such as EM can reliably build upon previous constructions that have been historically stable. The extensions to Sudoku described above demonstrate the quality of the original Sudoku model for exploring a wide range of learning domains.

7.6 Towards a 'constructivist computing'

The five modes of application for EM construals in §7.4 were conceived with Latour's five guarantees in mind. The fact that all five modes can be represented within a single modelling exercise is evidence that EM is well-matched to strengthening each of Latour's five guarantees *taken together.*

Despite only mentioning constructivism in the final chapter of this thesis, it should be evident that constructivism is a topic intertwined with many of the issues throughout the thesis. The characterisation of learning in Chapter 1 pulls together constructivist ideas, the exposition of EM develops the idea of 'constructionism', and the other chapters are full of examples of construction. In the light of Latour's attempt to rehabilitate constructivism, Beynon has proposed that EM as a computer-supported approach to learning be renamed 'constructivist computing' [Bey07c]. There is potential for such a label to be misconstrued given the concern surrounding constructivism, but Latour's vision for the notion of constructivism and the five guarantees offer a solid support for 'constructivist computing'.

The reason for this exposition on constructivism is to show that EM is a constructivist approach without the usual baggage or arguments that surround the term. Without ignoring the many different forms of constructivism, EM has a unique perspective on constructivism—due to the foundation of Radical Empiricism—that means the many forms need not be seen as in opposition to the core idea. EM need not necessarily be aligned to any one constructivist camp—to show that it is a constructivist activity we need only show that it supports the five guarantees suggested by Latour. To enter into the constructivist debate is to miss the point that, as with all knowledge constructions, constructivism cannot be absolutely defined. Entering the debate is definitely not going to achieve a definition that will be accepted by all, forever; as von Glasersfeld suggests, the best constructivist practitioners can do is take on what seems most viable [Gla90].

Conclusion

The research aim outlined at the beginning of the thesis was to answer: *How*, *where* and *why* can EM benefit learning and education? The broad scope of these three questions has been answered in three sections. In Chapter 1 I have shown 'why' there is a need for educational technology that is better aligned to learning, and in Chapter 2 I have offered EM principles and tools as a suitable approach and technology. Chapter 3 has explained 'why' EM offers a better solution over conventional software development and use. In Chapter 4 I have shown 'where' EM can be of benefit in computer science education. Chapter 5 has highlighted the potential 'where' EM can be of benefit in other subjects, in teaching, in lifelong learning and in collaborative learning. Finally, I have shown 'how' EM can be of benefit in Chapter 6 by examining evidence from projects and coursework undertaken by students in computer science at Warwick.

In Chapter 7 I have brought together the motivations (the 'why'), the principles and tools (the 'where') and the practical evidence (the 'how') in acknowledging the qualities of EM as an approach to constructivism. This leads to a vision for EM as computing that is constructivist in spirit, drawing on all aspects of EM activity, as depicted in Figure 0 on page 3. The contribution of the thesis is the complete approach to learning and education that is offered by EM. The significance of the contribution is that this complete approach overcomes the paradigmatic conflict between formalised educational technology and everyday sense-making and between the rich potential for enhanced learning afforded by new technology and the constraints of old-style educational practice, as discussed in Chapter 1.

To evaluate the contribution of the thesis, it is helpful to take a step back from the points raised in relation to EM and consider a broader agenda. There has been much discussion of different approaches to technology enhanced learning from a technology perspective. However, any technology enhanced learning is going to be heavily affected by the education system, whether it be schools, universities, or adult education. It seems to have been a common problem for technologists that they can get too excited about the innovation in technology and forgetful of the goal of innovation in learning.

In concluding the thesis I want to think about the situation 'on the ground' (e.g. in classrooms in schools) and consider the contribution that work such as this can potentially make. The average school teacher would no doubt regard the content of this thesis as concerned with problems largely disjoint from those experienced in a classroom. In the UK, the Teaching and Learning Research Programme (TLRP) is attempting to highlight the problems in our classrooms by putting practitioners and researchers together, in a hope that improvements can be made to teaching and learning practice. In September 2007 the TLRP sent out to all UK schools a guide to teachers entitled 'Principles into practice' highlighting ten principles for effective teaching and learning [TLRP07]. In an article about the ten principles, Andrew Pollard, director of the TLRP, proposes that schools forget targets and imagine aims that are 'much broader, more interesting and more intellectually challenging' [TLRP07]. The ten principles are introduced as a guide to effective teaching and learning that are not target-oriented, but are evidence-informed based on the collaboration of practitioners and researchers around the country [TLRP07]. The ten principles say teaching and learning should:

> "(1) equip learners for life, in its broadest sense; (2) engage with valued forms of knowledge; (3) recognise the importance of prior experience and learning; (4) require the teacher to 'scaffold' learning (support pupils as they move forwards); (5) make assessment congruent with learning; (6) promote the active engagement of the learner; (7) foster both individual and social processes and outcomes; (8) recognise the significance of informal learning; (9) depend on teacher learning; and (10) demand consistent policy frameworks, with support for teaching and learning as their main focus." [TLRP07]

There are some connections between the above ten principles and the eight characteristics of learning through EM. Both are suggesting a more *experimental*, *flexible* and *meaningful* approach to learning. There are also some indications to suggest that learning supported by model-building with EM may go some way to satisfying the ten principles of effective teaching and learning as set out by the TLRP.

The principles are much concerned with making teaching and learning more *meaningful* by *engaging with valued forms of knowledge* (2), *recognising the importance of prior experience and learning* (3) and *recognising the significance of informal learn-*

ing (8). Model-building supported by EM has been shown to be useful for exploring artefacts that are related to well-known topics and prior experience outside of school. Model-building is a personal experience and takes account of personal experience in the way that the ten principles encourage learning that takes account of personal and cultural experiences of different individuals and groups. The activities in the sudoku (§7.3), planimeter (§6.2.1) and Thai language (§5.1.2) models demonstrates that, as Pollard suggests [TLRP07], informal learning, such as learning out of school, is at least as significant as formal learning, and could be used within formal education. There is scope for using EM to support learning in an open-ended manner that could not be expected of conventional educational technology that is designed for a specific purpose. The value of EM is its openness to incorporate a wide variety of sources in any learning activity, whether they are traditional classroom sources such as books or more informal sources such as a student's experience from an activity outside the classroom. EM can support the TLRP call for more practical meaningful experiences in classrooms in ways that much traditional educational software cannot.

A requirement for teaching and learning to be *flexible* is expressed within Pollard's ten principles [TLRP07]. The first principle is concerned with *equipping learners for life in its broadest sense* (1) meaning that a broad view be taken of learning outcomes because learning is about developing people's intellectual, personal and social skills to equip them for their lives. To consider the curriculum as the most relevant subject-matter is a narrow view of teaching and learning. Educational technology that is closely based on aspects of the curriculum (e.g. a microworld for teaching Newton's laws of motion) engages with outcomes that are typically prescribed when building the software. EM is not constrained by a particular topic or outcome, as can be seen from the history of the 3D room model, which started life as simple DoNaLD example and later became a specialised component for teaching and learning about 3D to 2D transformations in computer graphics (see Figure 4.10 on page 105). A further aspect of flexibility relates to another of the ten principles which states the importance of *fostering both individual and social processes and outcomes* (7). The case studies in databases and computer graphics indicate the potential for EM to support both individual and teacher-student model-building, and the section on 'collaborative learning' (§5.4) has introduced the possibility of model-building in a classroom-like social environment. A topic to which this has given little consideration is assessment. One of the ten principles is that learning and teaching *needs assessment to be congruent with learning* (5) and an important point here is that assessment should help to advance learning. Assessment in model-building rests on Jonassen's maxim that if you can build a model of something then that is a good indication that you understand it [Jon06]. EM is well-aligned to this idea, as assessment could also rely on the history of interactions with a model, as well as the model itself. In this way, assessment using EM offers much more flexibility over conventional educational technology whose methods of assessment are constrained by a pre-specified functionality.

The third focus of the thesis on learning's *experimental* characteristics is also conveyed in Pollard's ten principles for effective teaching and learning [TLRP07]. According to the TLRP, a chief goal of teaching is to *promote the active engagement of the learner* (6) especially in terms of encouraging independent and autonomous learners. EM supports active engagement on the part of the learner by making model-building a personal activity that encourages model-builders to think about their own experience. The TLRP also acknowledge the essential part a teacher can play in experimentation by stating that effective teaching and learning *requires the teacher to scaffold learning* (4). EM offers a mode of interaction where the roles of student, teacher and developed are integrated. The section on presentations and lectures (§5.2) highlights the potential to support the teacher and at the same time provide scaffolding for the learner. Moreover, collaborative learning with EM offers another environment where the integration of student and teacher roles might help scaffold learning. The TLRP have another principle stating that effective teaching also *depends on teacher learning* (9), meaning that teachers should be actively engaging in developing their knowledge and experimenting with their skills, in order to benefit the learner. Beynon's experimentation with the 3D room model (§4.2) shows how EM can be equally beneficial to teachers and students.

The final principle to be covered says that effective teaching and learning *demands consistent policy frameworks with support for teaching and learning as their primary focus* (10). This principle is aimed at policy-makers in government, local education authorities and schools, asking for policies "to be designed to make sure everyone has access to learning environments in which they can thrive" [TLRP07]. Although technology might develop and styles of learning might change, it suggests that a main focus should be placed on maintaining consistent learning environments. As evident from the wide variety of models in this thesis, there is potential for EM to adapt to different technologies and styles of learning, whilst offering a consistent environment focussed primarily on learning.

The future of the work in this thesis depends on bringing the principles and tools closer to practitioners in education by taking it to teachers and schools, lecturers and universities, and more generally to lifelong learners. The successful application of EM as a support for learning requires more resources and better support from educational institutions and organisations such as the TLRP. In order to achieve such targets, there is potentially beneficial development of the tools to be considered. Some of the features I envisage for a state-of-the-art EM tool are: a web-based application for global access; a graphical interface for building models to bring the benefits of spreadsheet-style model-building; interface variations suitable for school children, university students, and teachers; built-in version control (for model-builders and researchers); a more accessible repository for sharing models amongst learners; collaborative capabilities for discussing, sharing and building models.

The benefits of these additional features in an EM tool would not only attract students and teachers but would also help researchers conducting empirical studies. Given the resources and interaction with students and teachers, the potential for EM to support learning and education as expounded in these pages might then be realised.

Bibliography

[Alb07] Albirini, A. (2007) The crisis of educational technology, and the prospect of reinventing education. *Educational Technology & Society*, 10(1), 227–236. (Cited on page 8, 9, 10, 11)

[Ari12] Aristotle (1912) *A treatise on government (Politics)*. Translated by William Ellis. London: J M Dent & Sons Ltd. (Cited on page 16)

[Asp90] Aspray, W. et al (1990) *Computing before Computers*. Iowa State University Press. (Cited on page 201, 203, 204, 208)

[Aus68] Ausubel, D.P. (1968) *Educational psychology: A cognitive view*. New York: Holt, Rinehart and Winston. (Cited on page 20)

[Bad85] Badgett, T. (1985) Lotus Symphony; a full-blown big brother for 1-2-3. *Creative Computing*, 11(2), 88. (Cited on page 76)

[BBR06] Beynon, W.M., Boyatt, R.C. & Russ, S.B. (2006) Rethinking Programming. In *Proceedings IEE Third International Conference on Information Technology: New Generations* (ITNG 2006), 10th-12th April 2006, Las Vegas, Nevada, USA, 149–154. (Cited on page 59)

[BBRW03] Beynon, W.M., Bhalerao, A.H., Roe, C. & Ward, A. (2003) A computer-based environment for the study of relational query languages. In *Proceedings of the Teaching, Learning and Assessment in Databases Workshop*, Coventry, UK, July 2003, 104–108. (Cited on page 4, 86)

[BC89] Brown, J.S. & Collins, A., & Duguid, P. (1989) Situated cognition and the culture of learning. *Educational Researcher*, 18, 32–42. (Cited on page 24, 25)

[BC03] Bartsch, R.A. & Cobern, K.M. (2003) Effectiveness of PowerPoint presentations in lectures. *Computers & Education*, 41, 77–86. (Cited on page 149)

[BC06] Beynon, W.M. & Chan, Z.E. (2006) A conception of computing technology better suited to distributed participatory design. NordiCHI Workshop on Distributed Participatory Design, Oslo, Norway, October 2006. (Cited on page 169, 170)

[BCC96] Beynon, W.M., Cartwright, R.I. & Cartwright, A.J. (1996) Abstract Geometry for Design in an Empirical Modelling Context. Research Report CS-RR-319, Department of Computer Science, University of Warwick. (Cited on page 98)

[BCY95] Beynon, W.M., Cartwright, A.J. & Yung, Y.P. (1995) Databases from an Agent-oriented Perspective. Research Report CS-RR-278, Department of Computer Science, University of Warwick. (Cited on page 86)

[Bea92] Beare. R.A. (1992) Software Tools in Science Classrooms, *Journal of Computer Assisted Learning*, 8, 221–230. (Cited on page 76, 77)

[Bea96] Beare. R.A. (1996) A Spreadsheet System for Data Analysis and Mathematical Modelling, *Computer Education*, 82, 11–15. (Cited on page 75)

[Bec00] Beck, K. (2000) *Extreme programming explained: embrace change.* Addison-Wesley: Boston. (Cited on page 59, 61)

[BEC04] BECTA (2004) ICT and e-learning in Further Education: embedded technology, evolving practice. A report to the Learning and Skills Council. British Educational Communications and Technology Agency (BECTA). (Cited on page 155)

[BEC07] BECTA (2007) Learning in the 21st century: The case for harnessing technology. British Educational Communications and Technology Agency (BECTA), Publication Ref: BEC1-15507. (Cited on page 8)

[Bey01] Beynon, W.M. (2001) Liberating the Computer Arts. In *Proceedings of First International Conference on Digital and Academic Liberty of Information (DALI 2001)*, University of Aizu, Japan, March 2001 (invited paper). (Cited on page 47, 225)

[Bey05a] Beynon, W.M. (2005) Radical Empiricism, Empirical Modelling and the nature of knowing. In I.E. Dror (Ed.) *Cognitive Technologies and the Pragmatics of Cognition: Special Issue of Pragmatics and Cognition*, 13(3), 615–646, December 2005. (Cited on page 51, 198, 218, 219, 221)

[Bey05b] Beynon, W.M. (2005) Computational Support for Realism in Virtual Environments. In *Proceedings of 11th International Conference on Human-Computer Interaction* (HCII 2005): Volume 10 - Internationalization, Online Communities and Social Computing: Design and Evaluation, Las Vegas, NV, 22-27 July 2005 (published on CD). (Cited on page 35)

[Bey07a] Beynon, W.M. (2007) Lecture notes on Empirical Modelling for concurrent systems. Department of Computer Science, University of Warwick. URL: `http://www2.warwick.ac.uk/fac/sci/dcs/research/em/teaching/cs405/concsys/` (Cited on page 30, 31, 32, 33, 37, 38, 40, 41, 86)

[Bey07b] Beynon, W.M. (2007) CS324 Computer Graphics course material 2006/7. Department of Computer Science, University of Warwick. URL: `http://www.dcs.warwick.ac.uk/~wmb/CS324/` (Cited on page 98, 99, 102, 103, 104, 106, 152)

[Bey07c] Beynon, W.M. (2007) Evaluating constructivist computing for education. A research proposal submitted to the TLRP Technology Enhanced Learning Call 2. (Cited on page 214, 235)

[Bey97] Beynon, W.M. (1997) Empirical Modelling for Educational Technology. In *Proceedings of Cognitive Technology*, University of Aizu, Japan, IEEE, 54–68. (Cited on page 2, 51)

[BH05a] Beynon, W.M. & Harfield, A. (2005) Empirical Modelling in support of constructionist learning: a case study. Deparmental Research Report No. 412, Department of Computer Science, University of Warwick. (Cited on page ix)

[BH05b] Beynon, W.M. & Harfield, A. (2005) Empirical Modelling in Support of Constructionist Learning: a Case Study from Relational Database Theory. In *Proceedings of 5th IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, Kaohsiung, Taiwan, July 2005, 396–398. (Cited on page ix, 39, 86, 156)

[BH06] Beynon, W.M. & Harfield, A. (2006) Rethinking Life-long Learning: the Empirical Modelling Approach. In *Proceedings of 6th IEEE International Conference on Advanced Learning Technologies (ICALT'06)*, Kerkrade, The Netherlands, July 2006, 854–858. (Cited on page ix, 146, 147, 154, 156)

[BH07] Beynon, W.M. & Harfield, A. (2007) Lifelong Learning, Empirical Modelling and the Promises of Constructivism. Journal of Computers, Volume 2, Issue 3, 43–55. (Cited on page ix, 154, 214)

[Bha06] Bhalerao, A. (2006) CS324 Computer Graphics course material 2005/6. Department of Computer Science, University of Warwick. URL: `http://www.dcs.warwick.ac.uk/~abhir/dcs/cs324/` (Cited on page 99, 111)

[BHB06] Boyatt, R., Harfield, A. & Beynon, W.M. (2006) Learning about and through Empirical Modelling. In *Proceedings of 6th IEEE International Conference on Advanced Learning Technologies (ICALT'06)*, Kerkrade, The Netherlands, July 2006, 662–666. (Cited on page ix, 185, 192)

[BHC05] Beynon, W.M., Harfield, A. & Chang, S. (2005) Alternative model-building for the study of socially interactive robots. In *Proceedings of AISB'05 Symposium on Robot Companions Hard Problems and Open Challenges in Human-Robot Interaction*, University of Hertfordshire, UK, April 2005, 5–15. (Cited on page ix, 117, 118, 119, 124, 128)

[BHJ05] Beynon, W.M., Harfield, A. & Jormanainen, I. (2005) Varieties of Concretisation: an illustrative case-study. In Proceedings of 5th Annual Finnish/Baltic Sea Conference on Computer Science Education (Koli Calling 2005), 17th-20th November 2005, 153–156. (Cited on page ix, 4, 97)

[BHV07] Beynon, W.M., Harfield, A. & Vesisenaho, M. (2007) Contextualising information and communications technology in developing countries. Accepted for 7th Baltic Sea Conference on Computer Science Education (Koli Calling 2007), 15th-18th November 2007. (Cited on page ix, 4)

[BNR89] Beynon, W.M., Norris, M.T., Russ, S.B., Yung, Y.P. & Yung, Y.W. (1989) Software construction using definitions: an illustrative example. Deparmental Research Report No. 147, Department of Computer Science, University of Warwick. (Cited on page 65)

[BNY94] Beynon, W.M., Ness, W.M. & Yung, Y.P. (1994) Applying Agent-oriented Design to a Sail Boat Simulation. In *Proceedings of ESDA'94*, Vol.6, 1–8. (Cited on page 130)

[BR04] Beynon, W.M. & Roe, C. (2004) Computer Support for Constructionism in Context, In *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, Joensuu, Finland, August 2004, 216–220. (Cited on page 39)

[BR07] Beynon, W.M. & Russ, S.B. (2007) Experimenting with Computing (preliminary version of extended abstract). European Computing and Philosophy (ECAP'07), Twente, The Netherlands, June 21-23, 2007. (Cited on page 1, 128, 210)

[Bri80] Bringuier, J.C. (1980) Conversations with Piaget. Chicago: Chicago University Press. (Cited on page 18)

[BRM06] Beynon, W.M., Russ, S.B. & McCarty, W. (2006) Human Computing: Modelling with Meaning. Literary and Linguistic Computing 21(2), 141–157. (Cited on page 38, 50, 53, 131, 132)

[Bro00] Brown, C. (2000) An agent-based parsing system in EDEN. Third year project report, Department of Computer Science, University of Warwick. (Cited on page 35)

[Bru66] Bruner, J.S. (1966) Toward a Theory of Instruction. Cambridge, MA: Belkapp Press. (Cited on page 18)

[Bru96] Bruner, J.S. (1996) The Culture of Education. Cambridge, MA: Harvard University Press. (Cited on page 24)

[BS03] Baker, J.E., and Sugden, S.J. (2003) Spreadsheets in Education: The First 25 Years. Spreadsheets in Education 1(1): 18–43. (Cited on page 76, 77)

[BS99] Beynon, W.M. & Sun, P-H. (1999) Computer-mediated communication: a Distributed Empirical Modelling perspective, Proceedings of CT'99, San Francisco, August 1999. (Cited on page 170)

[Bur86] Burton, J. (1986) Robert Fitzroy and the Early History of the Meteorological Office. *British Journal for the History of Science*, 19, 147–76. (Cited on page 42)

[Car00] Carr, S. (2000) As Distance Education Comes of Age, the Challenge Is Keeping the Students. The Chronicle of Higher Education. Issue published: February 11th, 2000. URL: `http://chronicle.com/free/v46/i23/23a00101.htm` (last accessed 27/06/07). (Cited on page 155)

[Car04a] Care, C. (2004a) Models that manage user experience of analog devices, Paper given at Psychology of Programming Interest Group (PPIG) Postgraduate Workshop, December 2004. (Cited on page 202)

[Car04b] Care, C. (2004b) Illustrating the History of the Planimeter. Third year project report, Department of Computer Science, University of Warwick. URL: `http://empublic.dcs.warwick.ac.uk/projects/planimeterCare2004/Docs/report.pdf` (last accessed 11/05/07). (Cited on page 132, 201, 207)

[Car99] Cartwright, R.I. (1999) *Geometric Aspects of Empirical Modelling: Issues in Design and Implementation.* PhD thesis, May 1999. (Cited on page 98)

[Cha04] Chang, S. (2004) Intelligent house. Third year project report, Department of Computer Science, University of Warwick. (Cited on page 124)

[CDGW92] Collett, T.S., Dillmann, E., Giger, A. & Wehner, R. (1992) Visual landmarks and route following in desert ants. *Journal of Comparative Physiology A*, 170, 435–442. (Cited on page 208, 209)

[CFP04] Collins-Sussman, B., Fitzpatrick, B.W., & Pilato, C.M. (2004) *Version Control with Subversion.* O'Reilly Media, Inc. (Cited on page 168)

[CHM04] Colley, H., Hodkinson, P., and Malcolm, J. (2004) Informality and formality in learning. A report to the Learning and Skills Research Centre from the Lifelong Learning Institute at the University of Leeds, UK (http://www.guidance-research.org/EG/LLLtop/workplacelearning/wrl/fnfl/iandf). (Cited on page 137)

[CKW05] Chen, N.S., Kinshuk, & Wang, Y.H. (2005) Cyber Schooling Framework: Improving Mobility and Situated Learning. ICALT 2005: 290–292. (Cited on page 168)

[Con95] Confrey, J. (1995) How Compatible are Radical Constructivism, Sociocultural Approaches, and Social Constructivism? In Steffe, L. and Gale, J. (eds.). *Constructivism in Education.* Lawrence Erlbaum Associates, Publishers. Hillsdale. New Jersey. (Cited on page 218)

[Dar59] Darwin, C. (1859) *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.* London: John Murray. (Cited on page 21)

[Dau07] Dautenhahn, K. (2007) Methodology and Themes of Human-Robot Interaction: A Growing Research Field. *International Journal of Advanced Robotic Systems,* 4(1), 103–108. (Cited on page 117, 118)

[DeB02] De Bra, P. (2002) Adaptive educational hypermedia on the web. *Communications of the ACM,* 45(5), 60-61. (Cited on page 136)

[Del00] Delcloque, P. (2000) History of CALL. An online presentation originally made at EuroCALL 2000. URL: `http://www.ict4lt.org/en/History_of_CALL.pdf` (last accessed 24/09/2007). (Cited on page 134, 145)

[Den89] Denzin, N.K. (1989) *Interpretive biography.* Newbury Park, CA: Sage. (Cited on page 195)

[Den98] Dening, G. (1998) *Readings/Writings.* Melbourne: Melbourne University Press. (Cited on page 133)

[Dew59] Dewey, J. (1959) *Experience and Education.* New York: Macmillan. (Cited on page 6, 21, 22, 24, 25, 27)

[DG05] Davies, J., Graff, M. (2005). Performance in e-learning: online participation and student grades. *British Journal of Educational Technology,* 36(4), 657–663. (Cited on page 155)

[Dij89] Dijkstra, E.W. (1989) On the Cruelty of Really Teaching Computer Science. *Communications of the ACM,* 32(12), December 1989. (Cited on page 61)

[Don91] Donald, M. (1991) *Origins of the modern mind: three stages in the evolution of culture and cognition.* Cambridge, MA: Harvard University Press. (Cited on page 10)

[DOr98] D'Ornellas, H.P. (1998) *Agent oriented modelling for collaborative group learning.* Masters thesis. Department of Computer Science, University of Warwick, UK. (Cited on page 131, 166, 167, 168)

[Dou98] Dougiamas, M. (1998) A journey into constructivism. URL: `http://dougiamas.com/writing/constructivism.html` (last accessed 28/02/07). (Cited on page 217, 218)

[Dwo59] Dworkin, M. (Ed.) (1959) *Dewey on Education.* New York: Teachers College Press. (Cited on page 16, 23, 24, 190)

[Efs06] Efstathiou, G. (2006) *C-GRAPH: A case study in the design, implementation and application of a definitive notation.* Masters thesis, Department of Computer Science, University of Warwick. (Cited on page 227)

[EJRV02] Eronen, P., Jarvela, J., Roe, C. & Virnes, M. (2002) Using Empirical Modelling to simulate robotics in Kids Club. In *Proceedings of 2nd Annual Finnish/Baltic Sea Conference on Computer Science Education (Koli Calling)*, Joensuu, Finland, 18-20 October 2002, 63–68. (Cited on page 4)

[Ehr00] S.Ehrmann, S. (2000) Technology & Revolution in Education: Ending the Cycle of Failure. *Liberal Education*, Fall 2000, 40-49. (Cited on page 96)

[EMD] Empirical Modelling Research Group. (2007) EDEN Documentation: Guides to EDEN, DoNaLD, SCOUT, SASAMI, AOP, EDDI. URL: `http://www.dcs.warwick.ac.uk/modelling/guides` (last accessed 07/08/07). (Cited on page 35, 43, 107, 119)

[EMP] Empirical Modelling Research Group. (2007) Empirical Modelling project archive. URL: `http://empublic.dcs.warwick.ac.uk/projects/` (last accessed 25/06/07). Individual projects referenced by [EMP:titleAuthorYear]. (Cited on page 1, 33, 35, 65, 66, 94, 97, 98, 99, 104, 106, 108, 112, 117, 129, 130, 131, 132, 133, 140, 146, 148, 149, 161, 166, 168, 170, 176, 178, 185, 200, 202, 208, 209, 224, 228)

[EMW] Empirical Modelling Research Group. (2007) Empirical Modelling website. URL: `http://www.dcs.warwick.ac.uk/modelling/` (last accessed 07/08/07). (Cited on page 1, 31, 37)

[Ern95] Ernest, P. (1995) The one and the many. In L. Steffe & J. Gale (Eds.). *Constructivism in education*, Hillsdale, NJ: Lawrence Erlbaum, 459–486. (Cited on page 217)

[ESV02] Eronen, P.J., Sutinen, E., Vesisenaho, M., & Virnes, M. (2002) Kids' club as an ICT-based learning laboratory. *Informatics in education*, 1(1), 61–72. (Cited on page 182)

[FB01] Fischer, C.N. & Beynon, W.M. (2001) Empirical Modelling in Product Design. In *Proceedings of 16th Brazilian Congress of Mechanical Engineering*, November 2001, 218–225. (Cited on page 130)

[FB02] Frey, B.A. & Birnbaum, D.J. (2002) Learners' perception on the value of PowerPoint in lectures. Pittsburgh: University of Pittsburg (ERIC Document Reproduction Service No. ED467192). (Cited on page 149, 151)

[Fis05] Fischer, G. (2005) Social creativity: making all voices heard. In *Proceedings of Human Computer Interaction International Conference (HCII)*, Las Vegas, July 2005. (Cited on page 166)

[FND02] Fong, T., Nourbakhsh, I. & Dautenhahn, K. (2002) A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42, 143–166. (Cited on page 118)

[Fri04] Friesen, N. (2004) Three Objections to Learning Objects and E-learning Standards. In McGreal, R. (Ed.), *Online Education Using Learning Objects*, London: Routledge, 59-70. (Cited on page 157)

[Gal01] Galloway, C.A. (2001) Vygotsky's learning theory. In M. Orey (Ed.), *Emerging perspectives on learning, teaching, and technology.* URL: `http://www.coe.uga.edu/epltt/vygotskyconstructionism.htm` (last accessed 28/02/07). (Cited on page 217)

[GdC05] Green, D., de Waal, A. & Cackett, B. (2005) Education: Better results or declining standards? Online Briefing, The Institute for the Study of Civil Society (CIVITAS), December 2005. URL: `http://www.civitas.org.uk/pdf/educationBriefingDec05.pdf` (last accessed 31/05/07). (Cited on page 8)

[Geh96] Gehring, D.K. (1996) Spreadsheets and programming. In *Proceedings of the First Psychology of Programming Interest Group Postgraduate Student Workshop*, ISBN 1 874152 04 7. (Cited on page 82, 85)

[Gen97] Gendlin, E.T. (1997) *Experiencing and the creation of meaning: a philosophical and psychological approach to the subjective.* Evanston, IL: Northwestern University Press. (Cited on page 26)

[Gib06] Gibbs, M. (2006) EditGrid, an excellent Web 2.0 spreadsheet. Web Applications Newsletter, Network World, 17th July 2006. (Cited on page 167, 182)

[Gil07] Gilbert, C. (2007) 2020 Vision: Report of the Teaching and Learning in 2020 Review Group (The Gilbert Report). Department for Education and Skills. URL: `http://www.teachernet.gov.uk/docbank/index.cfm?id=10783` (last accessed 10/06/07). (Cited on page 7, 8)

[Gil95] Gilbert, N. (1995) Emergence in social simulation. In N. Gilbert & R. Conte (Eds.), *Artificial Societies*, London: UCL Press, 144–156. (Cited on page 127)

[GKN01] Goldstein, R., Kalas, I., Noss, R. & Pratt, D. (2001) Building rules. In *Proceedings of Cognitive Technology: Instruments of Mind*, Coventry, UK, August 2001, 265–281. (Cited on page 70)

[Gla90] von Glasersfeld, E. (1990) An exposition of constructivism: Why some like it radical. In R.B. Davis, C.A. Maher & N. Noddings (Eds.), *Constructivist Views on the Teaching and Learning of Mathematics*, Reston, VA: NCTM, 19–29. (Cited on page 215, 216, 217, 235)

[GM95] Goodson, I. F. & Mangan, J. M. (1995) Subject Cultures and the Introduction of Classroom Computers. *British Educational Research Journal*, 21(5), 613–628. (Cited on page 9)

[Goo90] Gooding, D. (1990) *Experiment and the making of meaning.* Kluwer Academic. (Cited on page 31, 48, 210)

[Gua07] The Guardian. (2007) Sudoku classic, Tuesday July 17th 2007, Number 683 medium. (Cited on page 79)

[Gui03] Guillou, L. (2003) Empirical Modelling and math-based tutorial software. Final year project report, Department of Computer Science, University of Warwick. (Cited on page 130)

[Gul03] Gulati, S. (2003) Opening up lifelong learning. Paper presented at SCUTREA, 33rd annual conference, University of Wales, Bangor, 1-3 July, 2003. URL: `http://www.leeds.ac.uk/educol/documents/00003098.htm` (Cited on page 155)

[Hal66] Hall, E.T. (1966) *The Hidden Dimension.* Doubleday. (Cited on page 127)

[Har03] Harfield, A. (2003) Agent-oriented parsing. Third year project report, Department of Computer Science, University of Warwick. (Cited on page ix, 4, 35, 104, 185)

[Har04] Harfield, A. (2004) Exploratory language environments. Presentation at Psychology of Programming Interest Group (PPIG) Unroll Your Ideas Workshop 2004, University of Nottingham, 15th-16th December 2004. (Cited on page ix)

[Har06] Harfield, A. (2006) Learning about and learning through Empirical Modelling. Presentation at British Computer Society Coventry Branch Evening, University of Warwick, 19th April 2006. (Cited on page ix)

[Har07a] Harfield, A. (2007) Graphics presentation using 3D room. EM Project Archive. URL: `http://empublic.dcs.warwick.ac.uk/projects/graphicspresHarfield2007/` (Cited on page 109)

[Har07b] Harfield, A. (2007) Analysis of planimeter model week 9 revisions. Published on Google Docs & Spreadsheets and Google Video. URL: `http://docs.google.com/View?docID=w.dd3q4nm4_11wtbhkq` (last accessed 18/05/2007). (Cited on page 203, 205)

[HB86] Hearn, D. & Baker, M.P. (1986) *Computer Graphics.* Prentice-Hall, Englewood Cliffs, NJ. (Cited on page 99, 100)

[HCW05] Harfield, A., Chan, Z.E., & Ward, A. (2005) Recreating an experience of the Clayton Tunnel railway accident in 1861. Poster for Kaleidoscope Showcase 2005, Germany. URL: `http://empublic.dcs.warwick.ac.uk/projects/kaleidoscopeBeynon2005/posters/ClaytonTunnelPoster.pdf` (last accessed 03/07/07). (Cited on page ix, 176)

[Hoa85] Hoare, C.A.R. (1985) *Communicating Sequential Processes.* Prentice Hall. (Cited on page 38)

[Jac05] Jackson, M. (2005) Problem Frames and Software Engineering. *Information and Software Technology*, special issue on the 1st International Workshop on Advances and Applications of Problem Frames, 47(14), 903–912. (Cited on page 60, 70)

[Jac06] Jackson, M. (2006) What can we expect from program verification? *Computer*, IEEE Computer Society, October 2006, 39(10), 65–71. (Cited on page 56)

[Jam12] James, W. (1912) *Essays in Radical Empiricism.* London: Longmans, Green. Republished by New York: Dover Publications, 2003. (Cited on page 51, 165, 220, 229)

[Jam25] James, W. (1925) *Talks To Teachers On Psychology; And To Students On Some Of Life's Ideals.* New York: Henry Holt. (Cited on page 23, 190)

[Jam92] James, W. (1892) *Psychology.* London: MacMillan. (Cited on page 20, 23, 25, 26)

[JISC04] JISC (2004), Effective Practice with e-Learning. Online guide. Joint Information Systems Committee (JISC). URL: `http://www.elearning.ac.uk/effprac/index.htm` (last accessed 15/06/07). (Cited on page 7, 11)

[Jon06] Jonassen, D.H. (2006) *Modeling with technology: Mindtools for conceptual change.* Columbus, OH: Merrill/Prentice-Hall. (Cited on page 11, 12, 14, 17, 53, 77, 101, 102, 120, 124, 141, 194, 239)

[Kah96] Kahn, K. (1996) ToonTalk—An animated programming environment for children. *Journal of Visual Languages and Computing*, 7, 197–217. (Cited on page 68, 74)

[Kan06] Kanchanawan, N. (2006) Romanization, Transliteration, and Transcription for the Globalization of the Thai Language. *The Journal of the Royal Institute of Thailand*, 31(3), 832–841. (Cited on page 141)

[Kay84] Kay, A. (1984) Computer Software. *Scientific American*, September 1984, 251(3), 41–47. (Cited on page 76)

[KB00] Kalas, I. & Blaho, A. (2000) Imagine... a new Generation of Logo: Programmable Pictures. In *Proceedings of Conference on Educational Uses of Information and Communication Technologies*, World Computer Congress 2000 (WCC2000), Beijing, 427–430. (Cited on page 68, 74)

[Kee05] Keer, D. (2005) Modelling Navigation and Landmarking in Ants. Third year project report, Department of Computer Science, University of Warwick. (Cited on page 208, 210, 211)

[KI75] Karmiloff-Smith, A., & Inhelder, B. (1975) If you want to get ahead, get a theory. *Cognition*, 3, 195–212. (Cited on page 52, 53)

[Kin07] King, K. (2007) *Uncovering Empirical Modelling.* Masters thesis. Department of Computer Science, University of Warwick, January 2007. (Cited on page 1, 4, 37, 65, 71, 106, 108, 133, 197, 198, 224, 229)

[Kni07] Knight, J. (2007) Transforming Education Through Technology. Speech at BETT Conference on 10th January 2007. URL: `http://www.dfes.gov.uk/speeches/search_detail.cfm?ID=468` (last accessed 10/06/07). (Cited on page 7)

[Kno84] Knowles, M. S. et al (1984) *Andragogy in Action: Applying modern principles of adult education.* San Francisco: Jossey Bass. (Cited on page 155, 158)

[Kol84] Kolb, D.A. (1984) *Experiential Learning: Experience as The Source of Learning and Development.* New Jersey: Prentice Hall. (Cited on page 19, 20, 21, 22)

[Kon07] Kong, X.L. (2007) Guitar tutor. Final year project, Department of Computer Science, University of Warwick. (Cited on page 133)

[KS05] Knobelsdorf, M. and Schulte, C. (2005) Computer Biographies - A Biographical Research Perspective on Computer Usage and Attitudes Toward Informatics. In *Proceedings of Koli Calling 2005*, Conference on Computer Science Education, Koli, Finland, November 2005, 139–142. (Cited on page 195, 196)

[Kyn07] Kynigos, C. (2007) Half-baked Logo microworlds as boundary objects in integrated design. In *Proceedings of EuroLogo 2007*, Bratislava, August 2007. (Cited on page 143, 144)

[Lat03] Latour, B. (2003) The Promises of Constructivism. In D. Ihde & E. Selinger (Eds.), *Chasing Technoscience: Matrix for Materiality*, Indiana University Press, 27–46. (Cited on page 81, 214, 216, 218, 222, 223, 230, 231, 232, 233, 234)

[Lib06] Liber, O. (2006) Technological challenges for the re-design of the education system. Keynote address at 6th IEEE International Conference on Advanced Learning Technologies (ICALT'06), Kerkrade, The Netherlands, July 2006. Unpublished presentation slides. (Cited on page 157)

[LV04] Lund, H.H. & Vesisenaho, M. (2004) I-Blocks in an African Context. In Sugisaka (Ed.) *Proceedings of 9th International Symposium on Artificial Life and Robotics*, Oita, Japan. (Cited on page 182)

[LW79] Latour, B. & Woolgar, S. (1979) *Laboratory life: the social construction of scientific facts.* London: Sage. (Cited on page 217)

[LW91] Lave, J. & Wenger, E. (1991) *Situated Learning: legitimate peripheral participation.* Cambridge University Press. (Cited on page 25, 166, 203)

[Mac98] MacDonald, A. (1998) 3D Physical Modelling with Definitive Notations. Final year project report, Department of Computer Science, University of Warwick. (Cited on page 97, 98)

[Mar04] Martin, C. (2004) Empirical Modelling for crowd behaviour. Final year project report, Department of Computer Science, University of Warwick. (Cited on page 119, 120)

[McC03] McCarty, W. (2003) Knowing true things by what their mockeries be: modelling in the humanities. *CH Working Papers*, A.24, September 2003. URL: `http://journals.sfu.ca/chwp/index.php/chwp/article/view/A.24/`. (Cited on page 19)

[McC05] McCarty, W. (2005) *Humanities Computing.* London: Palgrave. (Cited on page 229)

[McP05] McPherson, M (2005) Editorial: Developing innovation in e-learning: lessons to be learned. *British Journal of Educational Technology*, 36(4), 585–586. (Cited on page 155)

[Mic04] Microsoft Corporation (2004) NetMeeting Software. url-http://www.microsoft.com/windows/NetMeeting/ (Cited on page 168)

[Mil02] Milton, J. (2002) Literature review in languages, technology and learning. A report for NESTA Futurelab, UK (http://www.futurelab.org.uk/research/reviews/lang01.htm). (Cited on page 137)

[MM99] Morgan, M.S. & Morrison, M. (Eds.) (1999) *Models as Mediators.* Ideas in Context Series, Cambridge University Press. (Cited on page 102)

[Mur04] Murphy, R.R. (2004) Human-robot interaction in rescue robotics. *IEEE Transactions on Systems Man and Cybernetics Part C–Applications and Reviews*, 34(2), 138–153. (Cited on page 123)

[Nar93] Nardi, B.A. (1993) *A small matter of programming: perspectives on end user computing.* MIT Press. (Cited on page 75)

[New69] Newell, A. (1969) Heuristic programming: Ill-structured problems in progress. *Operations Research*, 3, 360–414. (Cited on page 170)

[Nor03] Norvig, P. (2003) Powerpoint: shot with its own bullets. URL: `http://norvig.com/lancet.html` (accessed: 21/03/2007) (Cited on page 149)

[Nov84] Novak, J.D. and Gowin, D.B. (1984) *Learning How to Learn.* Cambridge University Press. (Cited on page 20)

[NP04] Negus, K. & Pickering, M. (2004) *Creativity, Communication and Cultural Value.* London: Sage. (Cited on page 17)

[OED] Oxford English Dictionary (1989) 2nd edition. Oxford University Press. (Cited on page 31)

[Pap80] Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas.* New York: Basic Books. (Cited on page 16, 24)

[PH91] Papert, S. & Harel, I. (1991) Situating Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism,* Norwood, NJ: Ablex, 1–12. (Cited on page 2, 16, 157, 208, 214)

[Phi02] Phillips, D.C. (2002) From radical empiricism to radical constructivism, or William James meets Ernst von Glasersfeld. In J. Garrison, R. Podeschi, & E. Bredo (Eds.), *William James and education,* New York: Teachers College Press, 115–129. (Cited on page 221)

[Phi95] Phillips, D.C. (1995) The good, the bad, and the ugly: The many faces of constructivism. *Educational Researcher,* 24(7), 5–12. (Cited on page 214, 215, 218, 220, 222)

[Pia29] Piaget, J. (1929) *The Child's Conception of the World.* Translation by J. & A. Tomlinson. London: Routledge & Kegan Paul. (Cited on page 154, 215, 216)

[Pia50] Piaget, J. (1950) The Psychology of Intelligence. (Cited on page 97)

[Pia71] Piaget, J. (1971) Biology and Knowledge: An Essay on the Relations between Organic Regulations and Cognitive Processes. Translation by B. Walsh. Edinburgh University Press. (Cited on page 18, 215, 216)

[Pia74] Piaget, J. (1974) La prise de conscience. Paris: Presses Universitaires de France (English translation by S. Wedgwood as The grasp of consciousness, Cambridge, MA: Harvard University Press, 1976). (Cited on page 52)

[Pla55] Plato (1955) The Republic. London: Penguin (translated by H. P. D. Lee). (Cited on page 20)

[Pol62] Polanyi, M. (1962) Tacit knowing: its bearing on some problems of philosophy. Reviews of Modern Physics, 34 (4), 601-616. (Cited on page 202, 208)

[Pop72] Popper, K.R. (1972) Objective Knowledge: An Evolutionary Approach. Oxford, UK: Oxford University Press. (Cited on page 19)

[Pow04] Power, D.J. (2004) A Brief History of Spreadsheets, DSSResources.com, URL: `http://dssresources.com/history/sshistory.html`, version 3.6, 08/30/2004. (Cited on page 75)

[Pre05] Pressman, R.S. (2005) Software Engineering: A Practitioner's Approach (International Edition). McGraw-Hill. (Cited on page 58, 61)

[PS05] Papert, S. & Strohecker, C. (2005) Catalyzing Debate about Fundamental Change in Education. In Price, S. (Ed.), Extended Abstract Proceedings of Technology and Change in Educational Practice, London: Knowledge Lab, Institute of Education. (Cited on page 17)

[RB07] Roe, C. & Beynon, W.M. (2007) Dependency by definition in Imagine-d Logo: applications and implications. In Ivan Kalas (Ed.) *Proc. of the 11th European Logo Conference*, 19-24 August 2007, Bratislava, Slovakia (published on CD ISBN: 978-80-89186-20-4). (Cited on page 179, 210, 211)

[Rep93] Repenning, A. (1993) AgentSheets: A tool for building domain-oriented dynamic, visual environments. PhD dissertation, Department of Computer Science, University of Colorado, Boulder. (Cited on page 74)

[Res96] Resnick, M. (1996) Distributed constructionism. In Proceedings of the International Conference on the Learning Sciences 1996, Northwestern University, pp. 280-284. (Cited on page 167, 182)

[Rey04] Reynolds, D. (2004?) Title. Final year project report. Department of Computer Science, University of Warwick. (Cited on page 65)

[Ril07] Riley, D.P. (2007) Educational technology and practice: types and timescales of change. Educational Technology & Society, 10 (1), 85–93. (Cited on page 10, 11, 17)

[Roe03] Roe, C. (2003) *Computers for learning: An Empirical Modelling perspective.* PhD thesis, University of Warwick, November 2003. (Cited on page 2, 37, 39, 51, 52, 53, 74, 129, 190, 196, 215)

[Roe99] Roe, C. (1999) The EMPTY project. Masters project, Department of Computer Science, University of Warwick, UK. URL: `http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/mscprojects/croe/`. (Cited on page 129)

[Rog61] Rogers, C.R. (1961) On becoming a person. Boston: Houghton Mifflin. (Cited on page 12, 13, 21, 22)

[Rou11] Rousseau, J.J. (1911) Emile or Education. Translation by B. Foxley. London: J.M. Dent & Sons. (Cited on page 24)

[RPJ05] Roe, C., Pratt, D. & Jones, I. (2005) Putting the learning back into e-learning. Presented at the CERME 4 conference, Spain, February 2005. URL: `http://cerme4.crm.es/Papers%20definitius/9/Roe-Pratt-Jones.pdf` (Cited on page 73, 74, 129)

[Run02] Rungrattanaubol, J. (2002) *A treatise on modelling with definitive scripts.* PhD Thesis. University of Warwick, April 2002. (Cited on page 1, 2, 37, 58, 59, 62, 67, 68, 77, 144, 160, 164)

[SB05] Swann, J. & Burgess, T. (2005) The usefulness of Karl Popper's selectionist theory of learning for educational practice. Learning for Democracy, 1 (3), pp. 7-22. (Cited on page 19, 20)

[SC98] Sun, P.H. & Cartwright, R. (1998) The Not-So-Quick Guide to dtkeden. urlhttp://www2.warwick.ac.uk/fac/sci/dcs/research/em/software/eden/dtkeden/. (Cited on page 170, 171)

[Sch99] Schrage, M. (1999) *Serious Play: How the World's Best Companies Simulate to Innovate.* Harvard Business School Press. (Cited on page 45)

[Sel99] Selwyn, N. (1999) Students' attitudes towards computers in sixteen to nineteen education. *Education and Information Technologies*, 4(2), 129–141. (Cited on page 9)

[Sfa98] Sfard, A. (1998) On two metaphors for learning and the dangers of choosing just one, *Educational Researcher*, 27(2) 4–13. (Cited on page 25)

[SG95] Steffe, L.P. & Gale, J. (1995) *Constructivism in Education.* Hillsdale, NJ: Lawrence Erlbaum. (Cited on page 215)

[SH00] Szabo, A. & Hastings, N. (2000). Using IT in the undergraduate classroom: should we replace the blackboard with PowerPoint? *Computers & Education*, 35, 175–187. (Cited on page 149)

[She97] Sheetz, S.D., Irwin, G., Tegarden, D.P., Nelson, H.J., Monarchi, D.E. (1997) Exploring the Difficulties of Learning Object-Oriented Techniques. *Journal of Management Information Systems*, 14 (Fall 1997) 103–131. (Cited on page 69)

[She98] Sheth, C.R. (1998) An investigation into the application of the distributed definitive programming paradigm in a teaching environment: the development of a virtual electrical laboratory. Masters thesis. Department of Computer Science, University of Warwick, UK. (Cited on page 166, 168)

[Shi86] Shinners-Kennedy, D. (1986) Using Spreadsheets to Teach Computer Science. Technical Symposium on Computer Science Education, Cincinnati, Ohio, United States, ACM Press. (Cited on page 76)

[SK99] Shaffer, D.W. & Kaput, J.K. (1999) Mathematics and virtual culture: an evolutionary perspective on technology and mathematics education. *Educational Studies in Mathematics*, 37(2), 97–119. (Cited on page 9, 10)

[Ski74] Skinner, B.F. (1974) *About Behaviorism.* New York: Knopf. (Cited on page 19)

[Smi96] Smith, B.C. (1996) *On the origins of objects.* Cambridge, MA: MIT Press. (Cited on page 137)

[Smi99] Smith, M.K. (1999) Andragogy. The encyclopaedia of informal education, `http://www.infed.org/lifelonglearning/b-andra.htm`. Last update: January 28, 2005. (Cited on page 154)

[Ste94] Steward, A. (1994) Spreadsheets in Mathematical Education. *International Journal of Mathematical Education in Science and Technology.* 25(2), 239. (Cited on page 77)

[Ste95] Steffe, L.P. (1995) Alternative epistemologies: an educator's perspective. In Steffe, L. P. & Gale, J. (eds), *Constructivism in Education*, Hillsdale, NJ: Lawrence Erlbaum. (Cited on page 215, 218)

[Ste99] Stein, L.A. (1999) Challenging the Computational Metaphor: Implications for How We Think. *Cybernetics and Systems*, 30(6), September 1999. (Cited on page 156)

[SP85] Strike, K.A. & Posner, G.J. (1985) A conceptual change view of learning and understanding. In L.H.T. West & A.L. Pines (Eds.), *Cognitive structure and conceptual change*, New York: Academic Press, 211–231. (Cited on page 11, 138)

[Sun99] Sun, P.H. (1999) *Distributed Empirical Modelling and its Application to Software System Development.* PhD thesis, Department of Computer Science, University of Warwick. July 1999. (Cited on page 110, 170, 176, 178)

[Tan99] Tan, K.C. (1999) Ant Simulation in Java. Third year project report, Department of Computer Science, University of Warwick. (Cited on page 209)

[Tay97] Taylor, J.M. (1997) *Investigating the applications of Empirical Modelling techniques to historical railway accidents.* Masters thesis. Department of Computer Science, University of Warwick, UK. (Cited on page 178)

[The05] Theodorou, A. (2005) Reviving ancient greek battle through the use of Empirical Modelling. Final year project report, Department of Computer Science, University of Warwick. (Cited on page 132)

[Tho03] Thomas, M. (2003) US Patent for Language teaching system. Patent number: 6565358. URL: `http://www.google.com/patents?id=ZzMNAAAAEBAJ` (last accessed 25/06/07). (Cited on page 134)

[TLRP07] TLRP (2007) Principles into practice: A teacher's guide to research evidence on teaching and learning. Produced by the Teaching & Learning Research Programme (TLRP). Sent out to every UK School in September 2007. URL: `http://www.tlrp.org/pub/documents/Principles%20in%20Practice%20Low%20Res.pdf`. (Cited on page 237, 238, 239)

[TSO98] The Scottish Office. (1998) Opportunity Scotland: A Paper on Life-long Learning. Presented to Parliament by Secretary of State for Scotland, The Stationery Office, September 1998. (Cited on page 154)

[Tuf03] Tufte, E.R. (2003) *The Cognitive Style of PowerPoint.* Chesire, CT: Graphics Press LLC. (Cited on page 149, 150, 151, 152, 153)

[Ull82] Ullman, J.D. (1982) *Principles of Database Systems.* Computer Science Press. (Cited on page 87)

[UoW07] University of Warwick (2007). E-learning Showcase Day 2007, University of Warwick. URL: `http://www2.warwick.ac.uk/insite/newsandevents/events/elearningshowcase/` (last accessed 25/06/07). (Cited on page 136, 145)

[Van02] Vanderstaeten, R. (2002) Dewey's transactional constructivism. *Journal of Philosophy of Education*, 36(2), 233–246. (Cited on page 217)

[Vin93] Vincenti, W.G. (1993) *What engineers know and how they know it: analytical studies from aeronautical history.* The Johns Hopkins University Press. (Cited on page 56)

[Vyg78] Vygotsky, L. (1978) *Mind in Society: The Development of Higher Psychological Processes.* MA: Harvard University Press. (Cited on page 20, 27, 44, 217)

[War04] Ward, A. (2004) *Interaction with meaningful state: implementing dependency on digital computers.* PhD thesis, University of Warwick, May 2004. (Cited on page 1, 4, 35, 37, 43, 68, 201)

[War96] Warschauer, M. (1996) Computer-assisted language learning: An introduction. In S. Fotos (Ed.), *Multimedia language teaching.* Tokyo: LogosInternational, 3–20. (Cited on page 135, 136, 145)

[War00] Warschauer, M. (2000) The death of cyberspace and the rebirth of CALL. *English Teachers' Journal*, 53, 61–67. (Cited on page 136)

[WEBEM1] WEB-EM-1 (2005) The First Warwick Electronic Bulletin on Empirical Modelling (WEB-EM-1), 2005. http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/web-em/01/ (last accessed 10/05/07). (Cited on page 4, 131, 145, 185, 186)

[WEBEM2] WEB-EM-2 (2006) The Second Warwick Electronic Bulletin on Empirical Modelling (WEB-EM-2). http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/web-em/02/ (last accessed 10/05/07). (Cited on page 121, 122, 192)

[WEBEM3] WEB-EM-3 (2007) The Third Warwick Electronic Bulletin on Empirical Modelling (WEB-EM-3). http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/web-em/03/ (last accessed 10/05/07). (Cited on page 43, 192)

[Wei88] Weinberg, G.M. (1988) *The psychology of computer programming.* New York: Van Nostrand Reinhold. (Cited on page 59, 65)

[Wen99] Wenger, E. (1999) *Communities of practice: learning, meaning and identity.* Cambridge University Press. (Cited on page 166)

[Wik07a] Wikipedia (2007) Microsoft PowerPoint. Wikipedia. URL: http://en.wikipedia.org/wiki/Microsoft_PowerPoint (accessed: 21/03/2007). (Cited on page 149)

[Wik07b] Wikipedia (2007) Sudoku. Wikipedia. URL: http://en.wikipedia.org/wiki/Sudoku (accessed: 21/03/2007). (Cited on page 78)

[Woo00] Woodforth, J. (2000) Interactive VRML front-end to dtkeden. Final year project, Department of Computer Science, University of Warwick. (Cited on page 178)

[WRSC99] Wiedenbeck, S., Ramalingam, V., Sarasamma, S., & Corritore, C.L. (1999) A comparison of the comprehension of object-oriented and procedural programs by novice programmers. *Interacting with Computers*, 11, 255–282. (Cited on page 69)

[Yun90] Yung, Y.W. (1990) *EDEN: An Engine for Definitive Notations.* Masters thesis, Department of Computer Science, University of Warwick. (Cited on page 33, 97)

[Yun93] Yung, Y.P. (1993) *Definitive Programming: a Paradigm for Exploratory Programming.* PhD thesis, Department of Computer Science, University of Warwick, January 1993. (Cited on page 33, 97)