Empirical Modelling principles to support learning in a cultural context

Chris Roe and Meurig Beynon

{croe,wmb}@dcs.warwick.ac.uk

Department of Computer Science, University of Warwick, Coventry CV4 7AL, U.K.

Much research on pedagogy stresses the need for a broad perspective on learning. Such a perspective might take account (for instance) of the experience that informs knowledge and understanding [Tur91], the situation in which the learning activity takes place [Lav88], and the influence of multiple intelligences [Gar83]. Educational technology appears to hold great promise in this connection. Computer-related technologies such as new media, the internet, virtual reality and brain-mediated communication afford access to a range of learning resources that grows ever wider in its scope and supports ever more sophisticated interactions.

Whether educational technology is fulfilling its potential in broadening the horizons for learning activity is more controversial. Though some see the successful development of radically new educational resources as merely a matter of time, investment and engineering, there are also many critics of the trends in computer-based learning who see little evidence of the greater degree of human engagement to which new technologies aspire [Tal95].

This paper reviews the potential application to educational technology of principles and tools for computer-based modelling that have been developed under the auspices of the Empirical Modelling (EM) project at Warwick [EMweb]. This theme was first addressed at length in a previous paper [Bey97], and is here revisited in the light of new practical developments in EM both in respect of tools and of model-building that has been targetted at education at various levels. Our central thesis is that the problems of educational technology stem from the limitations of current conceptual frameworks and tool support for the essential cognitive model building activity, and that tackling these problems requires a radical shift in philosophical perspective on the nature and role of empirical knowledge that has significant practical implications.

The paper is in two main sections. The first discusses the limitations of the classical computer science perspective where educational technology to support situated learning is concerned, and relates the learning activities that are most closely associated with a cultural context to the empiricist perspective on learning introduced in [Bey97]. The second outlines the principles of EM and describes and illustrates features of its practical application that are particularly well-suited to learning in a cultural setting.

1. Perspectives on educational technology for situated learning

There is a well-recognised danger of identifying education too closely and narrowly with abstract knowledge and understanding that can be construed as independent of situation and culture. This section discusses issues relating to providing technological support for learning in its broader context from both computer science and educational perspectives.

1.1 Computer support for the constructionist approach to learning

A desire to enrich the learning experience has been one factor in motivating constructionist rather than instructionist approaches to learning. The notorious "drill-and-kill" packages of the instructionist tradition reflect a concept of education as 'imparting established knowledge' that gives no scope for personal exploration of situation. The benefit that learners derive from these packages is seen as limited because of their lack of engagement with the material [Pap80]. Software developed with constructionist ideas in mind gives a different quality of learning experience: the student is actively engaged in a ‘microworld’ discovering knowledge as they explore. In this type of approach, the emphasis is on students actively constructing their own views of the world.

Learners who are enabled to construct their own models will benefit from approaches that can support different learning styles. With this in mind, Turkle and Papert call for a ‘revaluation of the concrete’ [Tur91], and appeal to developers to make software that encourages many learning approaches. They contrast the ‘planning approach’ of traditional computer programming with the ‘bricolage approach’ of craftworkers building artefacts [Lev68]. In our view, there are profound conceptual issues to be addressed before such a shift in emphasis in computer-based model-building can be achieved: the fundamental preconceptions about computation that inform classical computer science are ill-oriented for this purpose.

Central to the distinction between traditional computer programming and craftwork is the ontological distinction between computer programs, as classically conceived, and the artefacts that a craftworker creates in bricolage. A computer program is, of its essence, a rational product that supports planned user interactions and preconceived interpretations. Of course, the development of such a program may involve an energetic interleaving of prototypical use and consequent redesign (as for instance in the 'extreme programming' paradigm [Bec00]), but the quality of the program is evaluated with respect to fitness for its intended use and the extent to which it is optimised to serve this function. By contrast, bricolage involves an informal subjective interaction between a craftworker and the artefact he/she is creating that more closely resembles discovery than organised construction. The model-building activity has an experimental and creative quality: if it is successful, the character of the artefact itself changes in the mind of the discoverer as it develops – it is continuously being newly conceived and reinterpreted in stimulating ways.

In philosophical terms, the key issue that distinguishes the perspective of the classical programmer from that of the craftworker is the role that knowledge plays in the interpretation of human interaction with the object under development. Knowledge of the intended interaction and interpretation is a prerequisite for classical programming. In bricolage, the artefact under development is an embodiment of ignorance, and the craftworker's interaction – if successful – serves to shape the familiar from the unknown.

The purpose of the above discussion is to argue that our framework for thinking about computer programming and software development is ill-suited to the demands of constructionism. This is not to suggest that computer programming activity as practised has no element of bricolage. Many programmers see their work as a craft [Bro95], and can relate to the idea of wrestling with the meaning of software that is incompletely understood. The traditional view of software development broadly identifies two phases: one of engagement with the world, and of preliminary knowledge gathering, and a complementary phase where knowledge is deployed in program specification and design. In gathering knowledge, there is a clear role for interaction with artefacts (as represented by use cases, UML diagrams [Jac92], and prototypes of various kinds), and it is only in the deployment of this knowledge that the classical theory of computation has a fundamental role. The widely accepted view that – in practice – the activities associated with these phases must be interleaved does not do full justice to the intimate semantic relationship between them. Software development to support constructionist use demands more: a conceptual integration of the pre-articulate exploration and formalised knowledge that are associated with these two phases. The aim of EM, as will be discussed in section 2 below, is to propose concepts and principles that are better suited to this integration of human and computer centred activities (see also [Bey99]).

1.2 An empiricist perspective on learning in a cultural context

The connection between private pre-articulate experience (as in 'pre-articulate exploration of a domain') and formal communication of knowledge through language (as in 'knowledge that is sufficiently formalised to serve as a specification for software') is clearly of central relevance to learning. Education is broadly concerned with how hidden subjective individual experience can be revealed in ways that admit some objective (or quasi-objective) external evaluation. The empiricist perspective on learning (EPL) described in [Bey97] supplies a framework within which to conceive the constituent activities that are associated with the emergence of meaning and language out of interaction.

Within the framework of the EPL (see Box 1), we can speculatively trace learning from its origins in private experience to its expression in successful communication. Through interaction and observation we identify artefacts as integrated collections of observables, can observe their salient features and acquire skills in managing their state. We develop an understanding of our own agency in a situation by observing when things change and how they are related to the actions we perform. We can attribute certain changes to the responses of particular agents and so postulate agency, stimulus-response mechanisms and generic interactions to account for observed changes of state. By correlating our interactions with artefacts with those of other human agents, we can identify where it is appropriate to presume commonality of experience, and so develop languages for communicating, in the first instance about our common situation, and ultimately about more general and abstract domains. In this way, the formal elements that enjoy the greatest priority in traditional education systems are seen to rely upon experience and knowledge whose inter-subjective status is so well-established that it can be formalised.

private experience / empirical / concrete

interaction with artefacts: identification of persistent features and contexts
practical knowledge: correlations between artefacts, acquisition of skills
identification of dependencies and postulation of independent agency
identification of generic patterns of interaction and stimulus-response mechanisms
non-verbal communication through interaction in a common environment
directly situated uses of language
identification of common experience and objective knowledge
symbolic representations and formal languages: public conventions for interpretation

public knowledge / theoretical / formal

Box 1: An empiricist perspective on learning

As explained in more detail in [Bey97], the EPL is not to be interpreted as a template that applies to every act of learning. For instance, there is no necessary direct experience on the part of the learner to corroborate a result obtained using a calculator, or a fact acquired from a reference book. On the other hand, such a conceptual framework for learning is arguably essential if the intention is to give a learner access to concrete experience that informs received knowledge, to diagnose his conceptual difficulties in understanding, or to explore the extent to which comprehension is situated (cf. Lave's observation [Lav88] that some people can carry out calculations when shopping that they cannot perform in the classroom).

The relevance of the EPL is most clearly evident when we consider learning in a cultural context. Within a culture there are artefacts, conventions, rules, rituals, interpretations and skills, together with diverse languages whose semantics cannot be formally prescribed, but evolve through common usage and reflect both individual and collective behaviour. In such a context, the intimacy and subtlety of the relationship between interaction with artefacts, interaction with others within the same culture, and the development and use of language is conspicuous. A student of the piano, for instance, may be encouraged both to rehearse specific skills and to experiment with the instrument in ways that do not involve reading music; to play from progressively more sophisticated artefacts for representing music (such as pictorial representations of finger positions, then notations for notes for each and both hands, together with rhythmic patterns, key and time signatures, then dynamic markings, tempi markings and fingerings); to become familiar with the harmonic idioms and the standard repertoire of a particular musical tradition. For each level of attainment and genre of piano-playing, there is a pianistic competence and an appropriate level of sophistication in musical language (cf. "Play Middle C", "Play the harmonic scale of C sharp minor", "Play the octave passages in the coda of the Rondo in Beethoven's Waldstein sonata as glissandi"). It is significant that at its most sophisticated the language associated with a culture draws on such extensive experience and so many different sources of knowledge (e.g. in the above instance: music theory – *octave*; classical musical forms – *coda, Rondo*; musical history – *Beethoven, Waldstein*; and instrumental techniques – *glissando*) that it is only intelligible to the musical specialist.

An educational technology that could provide a framework sufficiently rich to support piano tuition in all the above aspects would be difficult to imagine with our current technologies. The virtue of the EPL is that it enables us to identify the conceptual distinctions that are an essential prerequisite for an effective implementation. For instance, within our empiricist framework, the unambiguous elements of musical theory (such as the interpretation of key and time signatures) can be distinguished from those that have a culturally determined meaning (e.g. 'How fast is Allegretto?', 'What frequency is concert A pitch?') and these in turn can be distinguished from artefacts such as fingerings that perhaps serve a purely private purpose for the performer, and can be adapted or added to the musical score at his/her discretion. It is also in principle possible to explore the empirical foundation that underlies languages and notations at every level of sophistication, as might be required for instance in calling up an electronic score of the particular passage of the particular Beethoven sonata to which the illustrative injunction to a pianist in the previous paragraph refers. As discussed in more detail in [Bey97], EM complements the EPL, and aspires to address the relevant technical challenges in implementation, some of which we discuss in section 2 below.

A significant question that has both implications and meta-implications for the theme of our paper is whether all learning can be construed as 'learning in a cultural or situational context'. It is self-evident that disciplines such as historical, social or religious studies are highly culturally dependent, but learning mathematics and science have perhaps traditionally been seen as more abstract objective topics that are culturally embedded only in a rather superficial sense. The following quotation, drawn from the preface of David Gooding's critique of established philosophies of science [Goo90, p. xi], promotes an interpretation of learning in science that is neither purely abstract, nor purely culturally defined:

> "Scientists' descriptions of nature result from two sorts of encounter: they interact with each other and with nature. Philosophy of science has, by and large, failed to give an account of either sort of interaction. Philosophers typically imagine that scientists observe, theorize and experiment in order to produce general knowledge of natural laws, knowledge which can be applied to generate new theories and technologies. This view bifurcates the scientist's world into an empirical world of pre-articulate experience and know-how and another world of talk, thought and argument. Most received philosophies of science focus so exclusively on the literary world of representations that they cannot begin to address the philosophical problems arising from the interaction of these worlds: empirical access as a source of knowledge, meaning and reference, and of course, realism."

Our critique of classical computer science as a basis for educational technologies echoes the sentiments in Gooding's quotation. The alternation between pre-articulate 'requirements capture' and language-driven program specification in iterative software development testifies to the bifurcation of the computer scientist's world. The role of the EPL is to redress the balance with respect to the focus on the literary world of representations to which Gooding refers. In the absence of a framework such as the EPL provides, it is unsurprising that educational technology has so far been at its most effective in respect of those elements of education that are perceived to be independent of culture, and that are most intimately linked with formal languages and theories.

In a subsequent passage from the same source [Goo90, p.5], Gooding observes:

> "The failure of computational approaches to deliver real discovery programs – to make discoveries with data that is as 'raw' as the stuff scientists work on – is largely due to the fact that most work with the impoverished notion of discovery still favoured by analytical philosophy."

This observation frames the challenge for educational technologies that aspire to support situated learning. Software to support learning in an exploratory idiom should do more than organise what is thoroughly well-understood so as to obscure elements of understanding. The scientist's discovery is not of the hide-and-seek variety: it involves the apprehension of new observables, and of previously unobserved relationships between familiar observables.

2. Empirical Modelling

Empirical Modelling (EM) is an approach to computer-based modelling that has been developed at the University of Warwick over many years [EmWeb]. Full details of the principles and tools that we have developed are beyond the scope of this short paper, but several key features that relate directly to educational support for learning in a cultural context will be discussed in this section. Some relevant case studies are also briefly reviewed. The interested reader can access screenshots of models and explore our research in more detail by consulting and/or downloading the web resources associated with these case studies. These can be accessed at [EmRep] by selecting the relevant project from the index. (For instance, to access the jugs model, select 'jugsPavelin2002'.)

2.1 From behaviour-as-programmed to state-as-experienced (and back again)

As explained above, traditional ways of thinking about computation give least support to the most primitive activities in the EPL. Computer programming as practised might be more aptly characterised as "computer-and-user programming"; the user has a role that is closely prescribed and has no choice but to adapt to the software. In complete contrast, human engagement with the artefact in the earliest stages of learning – as in bricolage – is stimulated by its openness to many possible modes of interaction and interpretation, rather than by pre-packaged automated responses to stimuli. Initially – until the learner has developed familiarity on his/her own terms – too sophisticated behaviour in an artefact presents an obstacle to the imagination, obstructing the projection of possible exploratory interpretations.

Formal computer science encourages the view that the only significant semantics of computer programs resides in the abstract patterns of behaviour and interaction that they support. This is a limited view: a computer offers direct experience of state whether or not it is executing a preconceived program, and (in principle, if not so obviously in practice) it is possible to shape this state to evoke different interpretations in much the same way that the craftworker moulds an artefact. Whilst the manipulation of program behaviours that an expert programmer effects when refining or debugging a program is within the scope of our concern in EM, our primary focus is on more primitive modelling activity. For instance, the archetypal role of the modeller in EM resembles that of a spreadsheet user, or an interface designer who is reconfiguring the state of a computer-generated screen to satisfy some practical or aesthetic requirement.

Our tools emphasise modelling that is state-based, where the term 'state' is to be understood as referring to 'state-as-experienced' rather than abstract computational state. A spreadsheet user always interprets the spreadsheet with reference to its current state and in relation to the external situational state to which it refers. The values in the spreadsheet are a snapshot of a set of observed external values at a particular moment in time. The definitions of values in the cells of the spreadsheet reflect the spreadsheet user's knowledge – or expectations – of dependencies between the values of the external observables to which they refer. Any change made to the value in a cell triggers updating of other cells that respects these dependencies, and the spreadsheet then represents a new external state. In the user's interaction, there need be no presupposition about the nature of future actions. A spreadsheet user is free to make any modification of cell values and definitions they wish. In practice, any constraints on how the spreadsheet user modifies the state are at the user's discretion. Typically – making allowance for the occasional bold but unjustified experimental interaction – the spreadsheet user only changes the spreadsheet state in ways that are meaningful in relation to its external interpretation (such as entering a new price for an item) or that assist its apprehension (such as highlighting the balance for the day in bold). The benefits of this open style of interaction with a spreadsheet in the exploratory investigation of a partially understood system are well-recognised [Nar93].

The principal modelling tool we have developed (the TkEden interpreter) uses scripts of definitions to model state. A definitive script records the current values of variables and dependencies relating them. The variables in the script represent observables in the application, and can be of a far wider range of types than in a conventional spreadsheet (cf. [EMRep, spreadsheetRoe2002]). The modeller is always free to make any redefinition that they wish, or to introduce new definitions into a model. This makes a model flexible and extensible.

Though the modeller always has discretion to interact with a definitive script directly, it is possible to set up program-like behaviours. Such behaviours are typically developed by identifying certain reliable patterns of change manually in an exploratory fashion. These can then be automated by introducing agents (implemented as triggered procedures in TkEden) that act according to specified stimulus-response patterns relating to the observables in the script.

The essential principles of constructing models in this fashion can be simply illustrated with reference to an EM version of Jugs, an educational program written for schools use to introduce elementary ideas of number theory. The screen display for Jugs (see Fig 1 on the next page) comprises two jugs (A and B) of integer capacities (capA and capB respectively) together with a button interface via which jugs can be filled and emptied, and liquid poured from one jug to another. The purpose of the model is to illustrate that, by performing sequences of Fill, Pour and Empty operations, it is possible to achieve any target integer quantity of liquid that is divisible by the greatest common divisor of capA and capB. In the EM Jugs model [EMRep, jugsPavelin2002], typical observables are the capacity of jug A, the content of jug A, the availability of the menu option 'Fill A', and the characteristics (e.g. location, size, content, background colour and sensitivity) of the associated menu button. These observables are represented by variables in a definitive script and the dependencies that link them by appropriate definitions. For instance, the colour of the ‘Fill A’ button depends on whether filling jug A is a valid option, which is turn dependent on whether or not the capacity of A currently exceeds its content. The 'fill jug A' action in the model is implemented by invoking an agent that increments the quantity unit-by-unit whilst the capacity of jug A exceeds its content, thereby imitating the manual actions of a human agent who introduces units of liquid into jug A one by one until the menu option 'Fill A' is seen to be invalid.

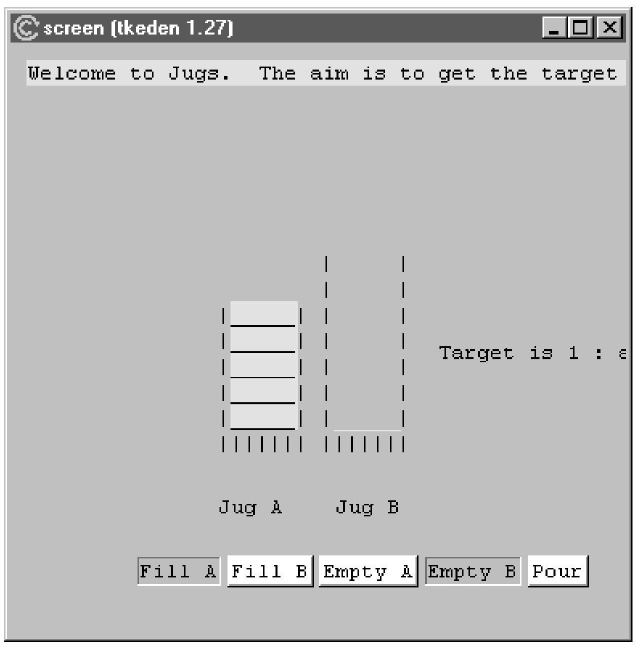

Fig 1 - A screenshot from the interface of the Jugs model

2.2 Applying the principles of EM in the EPL framework

From the discussion in section 1.2, it is apparent that the scope and diversity of learning activities associated with a cultural context is exceptionally broad. In learning, the human mind can seemingly draw from knowledge and experience of many different kinds, yet maintain a high degree of conceptual coherence and control. In this context, the use of educational technology is problematic because it inhibits the integration of diverse sources of insight both at the conceptual level and in practical terms. Conceptually, the auxiliary technology introduces a superfluous complexity of its own through its representations of our understanding. Practically, this complexity creates an inertia that in effect tethers the learner's mind to the limited range of perspectives that the technology dictates; modifying, extending and customising such technologies – if it is feasible at all – involves a disproportionate effort, and is only occasionally possible without significant redesign.

EM adopts an 'observation-oriented' approach to the challenge of developing educational software that can be adapted in diverse and unpredictable ways without sacrificing conceptual coherence. The following subsections review some practical studies and techniques in EM that relate to the key question – how can we integrate the diverse modes of observation that are associated with all the significant perspectives on the learning domain?

2.2.1 Integrating the current observations of the learner

One of the characteristics of successful learning, to which the EPL gives prominence, is the ability to bring together concepts and experiences from many different sources and to clarify the relationship between them. By way of illustration, the free distributive lattice on 4 generators (FDL4) is an abstract mathematical object with a formal characterisation that can also be viewed in many different concrete ways. It can be identified with the set of logically distinct boolean propositions in 4 variables that can be constructed using the 'and' and 'or' operations, ordered by $p \leq q$ if p implies q. Each element can be identified with the assignments of truth values to the 4 basic variables that make the corresponding proposition true, or with the family of subsets of {1,2,3,4} for which these assignments are the characteristic vectors. Under another interpretation, each element can be interpreted as a mapping from permutations of 4 symbols ("the symmetric group S4") into the natural numbers in the range 1 to 4. The entire lattice can also be represented by a traditional Hasse diagram with 166 nodes. Each view of FDL4 demands a different kind of background knowledge and mathematical sophistication, and makes different visual and cognitive demands on the human interpreter.

An EM model of FDL4 [EMRep, s4fdl4Beynon2002] illustrates how all of these different perspectives can be integrated within a single environment for visualisation and interaction. In the construction of the model, other forms of 'knowledge integration' relevant to learning activity are evident. The model includes a representation of S4 by a Cayley diagram that was extracted and adapted from another model [EMRep, cubesymWong2001] that exhibits S4 as the symmetries of a cube. Such a concrete representation for S4 could easily be incorporated in the model of FDL4, and typifies the kind of external situated familiarity a learner might have with an unfamiliar abstract mathematical object. As a further illustration, the points and lines in the Hasse diagram were entered by referring to a diagram previously created by John Buckle [Buc89]. Developing the full functionality of the FDL4 model involved integrating the diagram with an abstract algebraic model of FDL4 that was generated as an internal model and was represented by a table of boolean vectors. The process of correlating rows in this table with nodes in the diagram combined off-line interactions with Buckle's Hasse diagram with calculations using the internal model. Hybrid activity of this nature that combines interaction with an artefact outside the computer and interactions with a computer model – and leads to an eventual association of the two – illustrates another practical aspect of situated learning. It is also significant that this activity could be carried out despite the presence of errors – typical of a learning situation – introduced in translating Buckle's diagram into a definitive script.

In view of Gooding's concern about bifurcation in the scientist's world, the integration of interaction with artefacts with the use of language is of particular interest. The scope for such integration in EM is discussed at length in [Bey00], and illustrated with reference to an EM model that allows a formal specification of Heapsort to be interpreted as a special mode of observation within an exploratory environment in which variants of Heapsort can be developed. A related recent innovation is the development (by a final-year computer science undergraduate Chris Brown [Bro00]) of an observation-oriented parsing technique that enables us to develop and use new notations within the interactive environment of the TkEden interpreter. The clown-in-maze environment developed by the first author [EMRep, krustyRoe2002] shows how this parser can be exploited by introducing and refining simple languages dynamically in conjunction with interactive model-building. Depending on the context dynamically established by the modeller, the terminology available for specifying motion in this environment can encompass standard commands in Logo syntax, or be restricted to sets of actions such as 'up/down' and 'left/right' (optionally followed by a specified distance), or 'forward/backward' and 'turn'. The precise character of the notation available in the current context is constrained only by the skill of the modeller in parser specification, and can reflect alternative or complementary ways in which the learner observes and conceives the environment for interaction. For instance, 'up/down' and 'left/right' reflect a knowledge of orientation within the maze that is characteristic of the external observer, whereas 'forward/backward' and 'turn' more aptly describe the perspective of the clown.

2.2.2 Integrating observation associated with learning as it develops

The same principles that allow models to be adapted to suit the different perspectives of the learner in relation to his/her current understanding of a topic can be exploited by the teacher who wishes to create models that can scaffold learning. The clown-in-maze environment was originally conceived in this way. A more highly developed application of similar principles is to be found in the sqleddi environment [EMRep, sqleddiBeynon2001], which was created for the principal database module for computing-relating undergraduate students at the University of Warwick. The languages available within the sqleddi environment include eddi, a relational database query language closely modelled on Todd's ISBL notation [Tod76], and a partial implementation of standard SQL. The interaction within the environment is sufficiently rich and open-ended to support a range of behaviours suited to different learning objectives. Variants of SQL that use the same syntax, but have different evaluation strategies, serve to expose the discrepancies between SQL as commercially established and the underlying mathematical model on which it is based (cf. [Cod70],[Dat00]). The open experimental qualities of the environment are reflected in several ways: the same database can be interrogated and manipulated in both eddi and SQL at any stage; the evaluation strategy can be dynamically changed; and the environment also supports an SQL to eddi translation mode that shows the relational algebra equivalents of SQL queries (in so far as these exist!).

There is an important distinction between the family of artefacts that is associated with the sqleddi environment and user environments (such as ORACLE) that have a rich functionality that is typically progressively disclosed to the users as they become more advanced. The possible modes of interaction with sqleddi are much less rigidly specified, and each variant of the environment corresponds to a different observational perspective on how relational database queries are – or can be – framed and evaluated. The sqleddi environment is interesting in relation to the overall goals of EM in respect of educational technology in that the functionality of the environment as introduced in the associated worksheets was not preconceived, but was developed as the module itself was being delivered, and tailored to the educational agenda as it evolved. For instance, it was not originally apparent that query evaluation in standard SQL deviates so far from pure relational algebra that special syntactic conventions and pre-processing of SQL are required for successful translation. Future extensions of sqleddi, which in principle can be developed by the modeller in one continuous interaction within the environment itself, might address issues concerning integrity constraints and the data definition language that have yet to be given serious attention.

The observation-oriented analysis that guides development in EM establishes an intimate connection between the structure of an EM model and the modeller's understanding of the domain to which it refers. In this respect, such a model closely resembles what Gooding [Goo90] characterises as the 'construals' that the experimental scientist creates as an essential aid to understanding phenomena. The layers of understanding that are represented in the EPL and are prominent in learning in a cultural setting are mirrored in many EM models. Our model of the game of noughts-and-crosses [EMRep, oxoGardner1999] illustrates this by introducing the concepts that are required to play the game (or to develop a computer program to play the game) layer-by-layer in a manner that reflects their interdependency. This involves successively modelling the 3 by 3 grid, the introduction of Os and Xs to the grid, the static interpretation of a configuration of Os and Xs, the identification and evaluation of possible moves, the notion of player and turn etc. In other models, such as [EMRep, racingGardner1999], the layers of the model are associated with systematically introducing additional functionality that reflects ever more refined observation of a situation.

Such 'cognitive layering' of EM models has benefits for the student, the modeller and the teacher. The student can first gain experience of the domain at a simple level and understand the basic ideas. Where more complex issues are addressed in subsequent levels, the student has a good foundation to build on. By developing models layer-by-layer, the modeller gains access to a tree of possible design options, since the scripts at each layer can be reused, and refined or extended to suit different objectives. For instance, the model of the geometry in noughts-and-crosses can be elaborated to explore issues concerning the presentation of positions to the player, such as the style, location, size, shape and colour of elements of the grid display. For the teacher, adaptation of layered models of this nature can in principle be the basis for diagnosing a learner's difficulties.

2.2.3 Integrating observation of the learning context

The previous subsections have focused on the application of EM principles as they relate to the personal experiential aspects of learning. Within the EPL, these aspects are complemented by other activities that have a strong empirical component, but are concerned with the communication and collaboration in a cultural setting that spawns 'shared knowledge' and the emergence of 'objective' standards.

Many problems with existing educational technology stem from its relationship to the broader educational environment, not its use in isolation. Whether EM will – in due course – generate polished packages to compete with the best existing commercial products in special-purpose applications is unclear, but it offers clear potential for models to support learning that are better suited to the management of educational technology to meet pedagogical needs. A good teacher will customise resources to suit his/her teaching style, the nature of the class and the curriculum, and sometimes the special needs of individuals. This customisation may involve adapting the presentation of a topic, developing different types of artefact, or using different types of language. EM models are well suited to such customisation. Definitive scripts can be added to supply alternative interfaces that visualise appropriate parts of the underlying model, and can be tailored to particular types of student and display hardware. As is illustrated in several presentations that have been developed within TkEden (see e.g. [EMRep, introtodeppresentRoe2002]), the scripts associated with several models can be drafted into a common environment, and links between models created by introducing additional definitions. An important feature of the synthesis of EM models is that it can be managed in such a way that the constituent models require little revision and that they can still be accessed within the new environment in their original forms.

Communication between different cultural frameworks and across boundaries within a particular cultural framework is often the catalyst for learning. To give support for this, it is essential to focus not on established and uncontroversial knowledge, but on the interaction that leads to consensus and common understanding. For this purpose, educational software that allows the concurrent representation of several potentially conflicting viewpoints is required.

The potential for applying EM in collaborative working has been explored in our previous research on concurrent design [Bey94]. Practical support for such activity can be supplied using a distributed variant of the TkEden interpreter developed by Pi-Hwa Sun [Sun99]. Several experimental models that exploit distribution to set up multi-user environments for learning have been developed. These include a Virtual Electrical Laboratory that can be configured to allow students to collaborate in circuit construction and analysis under the discretionary supervision and direction of a tutor. The Clayton Tunnel simulation [EMRep, claytontunnelSun1999] provides a distributed environment to simulate the context for a historic railway accident. In this environment, students can simulate playing the roles of participants in a real-world situation. This role-playing represents an explicit enactment of situated learning that has proved motivating for students.

Conclusion

Whilst we recognise the current limitations of our tools in delivering large-scale practical solutions to the problems raised in this paper, we believe that EM is a natural approach to developing computer resources that are well-matched to everyday learning activities. One of the most important challenges to be addressed in future research is that of making our principles and tools more widely accessible and useful across the educational spectrum. To this end, we have developed a suite of models for a project aimed at introducing EM to UK schools. These show that our modelling approach can in principle be applied to a wide range of subjects (e.g. physics, geography, mathematics, and law), but there is much consolidation and further investigation to be done before this can have a practical impact in many subject areas. EM provides a framework within which a conceptual unification of programming paradigms is possible, but this does not provide simple solutions to the practical problems of linking our models to existing educational software or electronic resources. Perhaps the key question for the longer term is to what extent teachers and learners will be able to make direct use of EM principles and tools. There is some evidence that EM reduces the conceptual obstacles to the adaptation of models, and perhaps requires less technical programming skill than conventional styles of programming. There remain several technical problems to resolve in design and implementation of our tools, and for the present it seems likely that – to make full use of the proposals reviewed in this paper – teachers and learners would have to work in collaboration with expert modellers.

Acknowledgements

We are indebted to several members of the EM research group for material that has contributed to the content of this paper, and to Ashley Ward in particular for his recent work on TkEden and the associated EM repository.

References

[Bec00] K.Beck, eXtreme Programming explained, Addison-Wesley, 2000.
[Bey97] W.M.Beynon, Empirical Modelling for Educational Technology, Proc. Cognitive Technology’97, University of Aizu, Japan, IEEE, 1997, 54-68, 1997.
[Bey99] W.M.Beynon, Empirical Modelling and the Foundations of Artificial Intelligence, Computation for Metaphors, Analogy and Agents, LNAI 1562, Springer, 322-364, 1999.
[Bey94] W.M.Beynon, V.D.Adzhiev, A.J.Cartwright and Y.P.Yung, A Computational Model for Multiagent Interaction in Concurrent Engineering, Proc. CEEDA' 94, Bournemouth University, 227-232, 1994.
[Bey00] W.M.Beynon, J.Rungrattanaubol, J.Sinclair, Formal specification from an observation-oriented perspective, JUCS, Vol. 6 (4), 407-421, 2000.
[Bro95] F.Brooks, The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley, 1995.
[Bro00] C.Brown, An agent-based parsing system in EDEN, 3rd year project report, Department of Computer Science, University of Warwick, 2000.
[Buc89] J.F.Buckle, Computational aspects of lattice theory, Ph.D. Thesis, University of Warwick, U.K., 1989.
[Cod70] E.F.Codd, A relational model of data for large shared data banks, CACM 13, No.6, 1970.
[Dat00] C.J.Date, H.Darwen, Foundation for Future Database Systems: The Third Manifesto, Addison-Wesley, 2000.
[EMRep] http://empublic.dcs.warwick.ac.uk/projects
[EMWeb] http://www.dcs.warwick.ac.uk/modelling
[Gar83] H.Gardner, Frames of Mind: The Theory of Multiple Intelligences, New York: Basic Books, 1983.
[Goo90] D.Gooding, Experiment and the Making of Meaning: Human Agency in Observation and Experiment, Kluwer Academic, Dordrecht and Boston, 1990.
[Jac92] I.Jacobson, M.Christeron, P.Jonson, G.Overgaard, Object-orienetd Software Engineering: A Use Case driven approach, Addison-Wesley, 1992.
[Lav88] J.Lave, Cognition in Practice: Mind, Mathematics and Culture in everyday life, Cambridge University Press, 1988.
[Lev68] C.Levi-Strauss, The savage mind, University of Chicago Press, 1968.
[Nar93] B.Nardi, A Small Matter of Programming: Perspectives on End-user Computing, MIT Press, 1993.
[Pap80] S.Papert, Mindstorms: Children, computers and powerful ideas, Brighton - Harvester, 1980.
[Roe99] C.Roe, The EMPTY project : Empirical Modelling Principles for Teaching Youngsters, MSc dissertation, University of Warwick, 1999.
[Sun99] P-H.Sun, Distributed Empirical Modelling and its Application to Software System Development, Ph.D. Thesis, University of Warwick, U.K., 1999.
[Tal95] S.Talbott, The Future does not Compute: Transcending the Machines in our midst, O' Reilly and Associates, 1995.
[Tod76] S.J.P.Todd, The Peterlee relational test vehicle, IBM system journal, 15, 4, 1976, 285-308, 1976.
[Tur91] S.Turkle, S.Papert, Epistemological Pluralism and the revaluation of the concrete, In Constructionism, I.Harel & S.Papert (eds) (Ablex Publishing Corporation, 1991), 161-191.