# A Cognitive Tool for Classification Learning[1]

Tom Conlon
School of Education, University of Edinburgh, Edinburgh EH8 8AQ, Scotland
tom.conlon@education.ed.ac.uk
February 2000

**Abstract** This paper describes InterModeller, a computer program intended to assist children in primary and secondary schools to learn concepts and skills associated with classification. The design of the program consolidates and extends previous work by including support for multiple forms of representation, provision for automatic transformation between representational forms, and the ability to convert an inefficient model into an efficient one. A 'seven steps' methodology for classroom model-building is proposed and justified. Evaluation evidence suggests that the program engages learners effectively in constructive thinking and that its incorporation of a variety of forms of representation enriches the model-building process.

**Keywords** classification, expert systems, knowledge-based systems, modelling, learning.

## 1. Introduction

The aim of this paper is to present and justify the design of InterModeller, a computer program intended to support children in building models of classification-oriented subject matter [1]. With InterModeller, a child who is learning about spiders, dinosaurs or planets can build a model to represent his or her developing knowledge of the domain. Models once constructed can be run as small-scale expert systems that perform interactive classification. InterModeller differs radically from earlier systems by providing a choice of multiple forms of representation. These are readily interchangeable, so that a model can be converted automatically into a different form of representation if the originally selected form turns out to be unsuitable.

A major difference between classroom and business contexts of software use is that in classrooms the main goal is to achieve a rich process of development rather than a useful product. This paper will recommend a general approach to the process of classroom model-building. However, finished InterModeller models do have some value. They can be demonstrated and discussed, shared as files across networks, pasted into word processors and graphics programs, or just printed out to create classroom display material.

### 1.1 Classification learning

Education theorists have repeatedly stressed that classification skills are central to many forms of learning [2,3,4]. For instance, Bruner wrote:

> We map and give meaning to our world by relating classes of events rather than individual events. The moment an object is placed in a category, we have opened up a whole vista for 'going beyond' the category by virtue of the superordinate and causal relationships linking this category to others. [2, p13]

---

1 This paper appears in the International Journal of Continuing Engineering Education and Lifelong Learning, pp189-201 Vol 11 No 3, 2001. It is a revised version of a paper originally presented to the Ninth International Peg Conference (PEG99) on Intelligent Computer and Communications Technology, University of Exeter, UK, 10th-12th July 1999.

The influential characterisation of child development proposed by the psychologist Piaget gave a central place to the growth of skills in classification and reasoning [5]. In the so-called preoperational stage, at perhaps the age of four, children typically begin to show ability to perform simple sorting of physical objects, for example the grouping of toys by shape (e.g. round toys versus square toys). In the concrete operational stage, roughly from age six to twelve, children develop an understanding of the principle of class inclusion, enabling hierarchical classifications (i.e. nested categories) to be constructed from collections of things provided these are presented concretely (e.g. pictures of toys can be categorised into round toys, square toys, round red toys, round yellow toys, square red toys ….). Finally, in the formal operational stage roughly from twelve onwards children learn how to classify conceptual or abstract entities, things that can be imagined but are not concretely presented. This stage also sees the growth of capability of deductive logic — the use of if/then relationships to make predictions and solve problems.

Piaget's characterisation has been contested by more recent research which has tended to challenge the coherence and 'strictness' of the stages. For example, it has been argued that children understand hierarchical classification by age seven or eight [6] whilst at the other end of the scale it has been estimated that only 50-60 percent of 18-20-year olds in Western countries use formal operations at all, let alone consistently [7]. Such findings however do not undermine the general importance of classification learning.

Schools today have a practical interest in classification, as surveys conducted by the author [8] confirmed. Curriculum documents for the 5-14 age range in Scotland have a relatively high number of references to classification. Teachers for this age range across several subjects give a fairly high value to classification-oriented classroom tasks.

## 1.2 Considerations for a cognitive tool

At present there seems to be very little technological support for learning in classification. InterModeller is the latest fruit of a project that has run for several years with the aim of developing such support. More specifically the aim has been to construct a cognitive tool, a class of software defined by Jonassen [9, p2] as 'generalisable tools that can facilitate cognitive processing'. Cognitive tools share with tutoring programs the aim of promoting thinking and learning but they differ from tutoring programs by putting the learners in charge. According to Salomon [10], cognitive tools:

> … are *tools* inasmuch as their operation depends on learners' operations; they are *cognitive* inasmuch as they serve to aid students in their own constructive thinking, enabling them to engage in cognitive operations they would not have been capable of otherwise [10, p180]

The underlying hypothesis of the present project is that effective support for constructive thinking in classification can be provided by a model-building environment in which learners create software representations (or models) of classification structures. The environment provides editors for building these models and an interpreter which can run any model as an interactive classifier.

Importantly, this approach combines constructivist learning with great flexibility. Constructivist theories such as cognitive apprenticeship [11] stress that a learner's articulation of knowledge about a subject is likely to engage thinking and refinement of knowledge. Model-building is a demanding form of articulation. The flexibility of the approach stems from the fact that the tool itself is not committed to any subject domain. The 'content' comes from the learner whose grasp of abstract classification principles should be strengthened by experiencing them in a variety of domain contexts.

## 1.3 Primex, the predecessor of InterModeller

The first attempt to build such a cognitive tool was not very successful. Named Primex, this program provided a language of if/then rules similar to that provided by the first generation of commercial expert system shells. Although theoretically such rules have great expressive power, to children struggling to develop a capacity for abstract deductive logic they presented obvious difficulties in practice. It was concluded that other, more accessible forms of representation should be adopted that are more specific to the requirements of classification [12].

There followed a period of experimentation with a variety of representational forms which were developed initially as software extensions to Primex. Inspiration for this work, which has been reported previously [13,14], came from three main sources. The first was an immensely rich artificial intelligence literature on knowledge representation that provided a fund of implementable theories and techniques [e.g. 15,16]. Second, established teaching materials and interviews with teachers yielded insights into current classroom practice. Third, and most importantly, a programme of classroom-based research using the Persistent Collaboration Methodology [17] guided the refinement of prototype tools.

# 2. Design of InterModeller

The findings of that earlier work have now been consolidated and extended in the design of InterModeller. Available for both Macintosh and Windows computers, this program which is implemented in Prolog++ [18,19] offers learners a variety of forms of representation for building classification models. Figure 1 illustrates a session with InterModeller and shows the program's most significant feature, the provision of multiple forms of representation. There are two other novel capabilities besides: it is possible automatically to transform models into alternative forms of representation; and an inefficient model can be converted automatically into an efficient one. These features and the principles that underpin them are discussed below.

## 2.1 Multiple forms of representation

InterModeller provides the following forms of representation:

- Classification trees (see the 'Bikes ctree' window in Figure 1) organise classes or categories hierarchically, with arcs representing subclass relationships. InterModeller uses normal type to identify class names and italic type for class features, which are supposed to provide prototypical descriptions of each class.

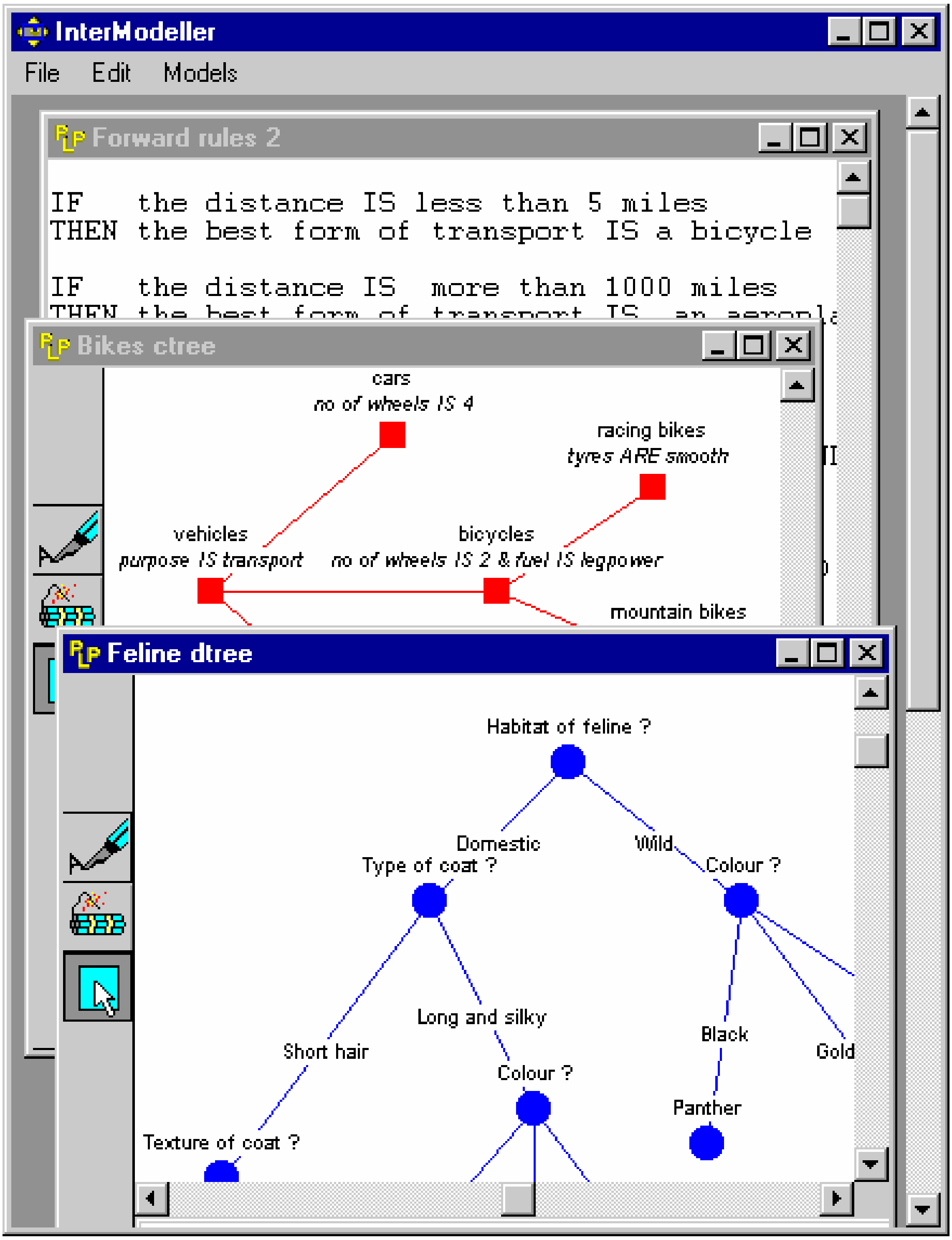


**Figure 1 InterModeller screen shot**

- Decision trees (see the Feline dree window) are flowchart-like representations which use a branching structure of questions and answers to distinguish between categories.

- Factor tables (see Figure 2) are more commonly called decision tables. Each category named in the rightmost column is defined by a row which specifies its features as values for a set of attributes named on the header row. [20]
- Rules are statements of if/then relationships. Each category is defined by one or more rules that specifies a set of conditions sufficient to identify the category. Rules in InterModeller may be written in a variety of syntaxes: the examples shown in Figure 1 illustrate the so-called Forward Rules syntax, in which conditions follow an attribute/value structure and conclusions appear at the end of the rule.

These forms of representation were selected because they are well matched to the needs of classification modelling. Specifically it can be noted that the coherence of a classification structure is much more manifest when the structure is expressed in tree or table rather than rule form. Also, our research suggested that with the possible exception of rules, these are all forms of representation to which children already are exposed in a variety of contexts in primary and early secondary schooling.

The provision of multiple forms is partly an acknowledgment that there is no 'best' form of representation for classification. A decision to prefer one form over another may be justified by a range of contextual factors including the structure of the information provided by the source domain, the purpose to which the model will be put, and the previous experience of the model-builder. Since classification 'in the world' may take many forms it seems right that a learning environment should offer practice of a variety of these, rather than imposing a single form.

At runtime a model of whatever kind is compiled into a classification procedure and the system engages in a question-answer dialogue with its user in an attempt to classify an item. If students are to learn how classification representations can be used for problem-solving then it is surely important that this runtime behaviour should be transparently related to a model's source form. Fortunately the relationship is mostly quite straightforward, with the possible exception again of rule models. For both kinds of tree model, the dialogue corresponds to a traversal of some branch of the tree starting from the root. For a table model it is a top-to-bottom search through the rows of the table, eliciting attributes in column order. It would be possible to 'animate' a model at runtime so as to trace the dialogue strategy, but the existing transparency is such that the effort hardly seems justified.

| No. of wheels | Fuel | Vehicle |
|---|---|---|
| 4 | Petrol | Car |
| 4 | Diesel | Van |
| 2 | Leg power | Bicycle |
| 2 | Petrol | Motorbike |

**Figure 2 Example of a factor table**

## 2.2 Transforming representations

In the experimental Primex system mentioned earlier, multiple representations were provided but the selection of a representational form was a commitment by the user that could hardly be reversed. The only way to obtain (say) a table version of an existing decision tree model was to rebuild the model from scratch, copying over by hand into a table all of the information that had become locked into the tree. Thus learners who selected initially an unsuitable form of representation were heavily penalised. This gave too little incentive to experiment with unfamiliar representational forms and to explore alternative representations of a domain.

By contrast, in InterModeller it could hardly be easier to transform a model. A menu command summons the dialogue box shown in Figure 3 and on clicking Ok, a model in the required form of representation is generated automatically. Normally this model is provided in a new window, leaving the original model untouched. Transformation is fast, typically taking perhaps a second or two of time.

Implementation of this capability depends upon a set of methods which can translate InterModeller's representational forms into Prolog-encoded rule representations and vice-versa. In most cases, these methods are straightforward and give good results. An exception affects transformation into classification tree form from a source model that is not inherently hierarchically structured. For example, the table model shown in Figure 2 will transform into a classification tree model containing four unconnected nodes, one per vehicle type.

As mentioned above, automatic transformation is intended to encourage experimentation with representational forms. From the learner's perspective, the implementation methods are black boxes. However, learners who compare the two versions of a model following transformation may be able to induce some of the methods involved.

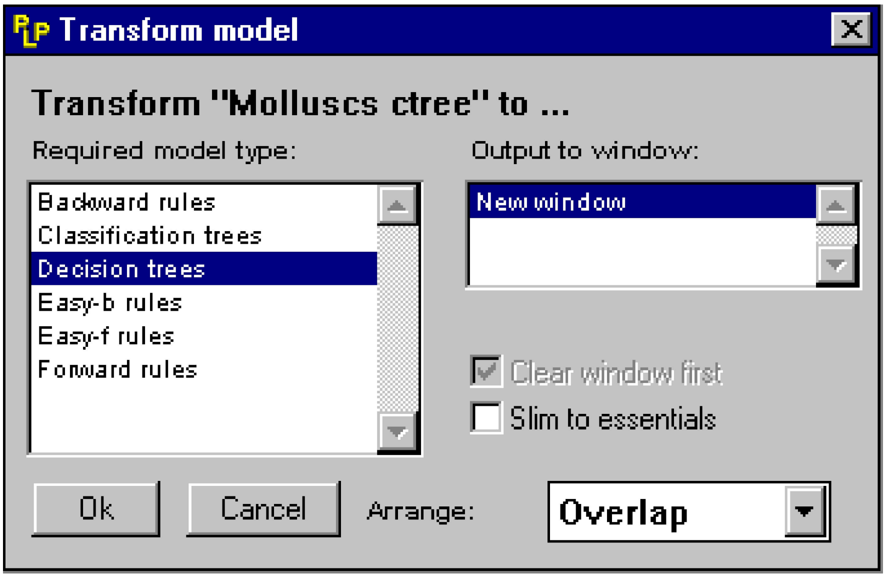


**Figure 3 The Transform dialogue**

## 2.3 Converting for efficiency

A person skilled in classification knows what questions to ask in order to arrive most rapidly at a conclusion. A simple example is again provided by Figure 2. A skilled classifier seeking to use this factor table to identify a vehicle will ask about `Fuel` first, since the answer to that question may be all that is needed to reach a conclusion (making use of the 'closed world' assumption). This insight could be encoded as Figure 4 which re-sequences the columns and marks the 'don't care' values by asterisks or wildcards. Using InterModeller's method of interpretation of factor tables, in which attributes are elicited in column order and wildcard values cause questions to be skipped, Figure 4 will run in a manner that exactly simulates the efficient procedure of the skilled classifier.

| Fuel | No. of wheels | Vehicle |
|---|---|---|
| Petrol | 4 | Car |
| Diesel | * | Van |
| Leg power | * | Bicycle |
| Petrol | 2 | Motorbike |

**Figure 4 Revised table model**

In fact, InterModeller can make such efficiency-improving model transformations automatically. All that is required by the user is to select the 'Slim to essentials' box in the transformation dialogue box (see Figure 3). Note that the efficiency benefit can be combined with a switch of representational form. For example, it might be thought preferable to slim Figure 2 to decision tree form since this representation makes the improved questioning strategy very obvious.

The implementation of this capability makes use of the ACLS induction technique [21]. In educational terms, three benefits are hypothesised. First, it provides helpful scaffolding: learners can concentrate on obtaining a correct declarative representation of the domain, without concern for procedural efficiency since that can be supplied later by an easy transformation. Second, InterModeller demonstrates a skill — that of generating an efficient classification procedure from declarative knowledge — which learners should appreciate and hopefully, ultimately acquire themselves.

A third benefit is that the process can sometimes expose interesting features of a domain. An example is a table model [27] which classifies 18 cases of coastal farmland erosion in terms of altitude, soil type, rainfall and grazing factors. After slimming the same classification is achieved using only 8 cases and the altitude factor is eliminated. On the evidence available, therefore, it would be reasonable to conclude that altitude is not relevant to erosion — something that was far from evident from the model prior to transformation.

# 3. Model-building methods

Classroom teachers sometimes respond to InterModeller by expressing uncertainty about how the technology can be incorporated within their practice. A common question is: How can children be taught to build models? Recent classroom-based action research with InterModeller has addressed this question. In what follows, some of the conclusions of that research are presented.

## 3.1 Knowledge engineering methodologies

Children building InterModeller models are doing a form of knowledge engineering and so the knowledge engineering literature is an obvious source of ideas for classroom methodology. The de facto standard KADS approach [15] is clearly over-elaborate for classroom purposes but less formal methods, generally based upon versions of evolutionary prototyping, are well documented. One such approach recommended by Durkin [22] contains the following stages:

1. Assessment of requirements
2. Knowledge acquisition
3. Design [and implement]
4. Testing

5. Documentation
6. Maintenance

Each stage is defined by tasks. For the design stage there are six, as follows:

3-1. Select knowledge representation technique
3-2. Select control technique
3-3. Select expert system development software
3-4. Develop the prototype
3-5. Develop the interface
3-6. Develop the product

These tasks too are iterative. Assuming no severe backtracking is required, the result should be a succession of working models which gradually evolve into the final system.

## 3.2 Webb's methodology

There are few published classroom-based studies of children's model building methods. An exception is that by Webb [23] who, following classroom observation of primary children using the Expert Builder shell, recommended an approach based upon the following stages:

1. Identify the area of interest
2. Define the problem
3. Decide on the scope, boundaries and purpose of the model
4. Build the model
5. Test the model
6. Evaluate the model

Note the resemblance to Durkin's methodology. However, children needed help and Webb reported that steps 1 and 2 were commonly done by the teacher. Teachers intervened at three levels: manipulating the software, selecting knowledge and structuring the model. A good place to start was found to be interacting with, and making modest extensions to, a preconstructed model. Webb noted that providing the subject matter was familiar to them, children as young as eight could be helped to build successful models.

## 3.3 InterModeller's 'Seven Steps'

During the development of InterModeller the methodologies of Durkin and Webb were discussed extensively with classroom triallists of the software. Durkin's approach was generally viewed as over-complex. Webb's approach on the other hand, while regarded as useful, was considered to be lacking in detail. Finer-grained guidance was thought to be needed especially in support of teachers and children who are new to this kind of model building.

The 'Seven Steps' methodology which evolved from these discussions refines Webb's approach in two main ways. First, where Webb's Expert Builder offered no choice of representational form, InterModeller provides a rich set of alternatives. Selecting a suitable form of representation becomes a major step in the model-building process. Second, InterModeller's focus on classification provides constraints that are helpfully exploited to guide the development process.

The methodology comprises the following seven steps:

1. Decide on the purpose of the model

2. Identify decision factors
3. Select a form of representation
4. Review the design
5. Start the model
6. Develop the model
7. Reflect and evaluate

These are described briefly below. For teachers and learners the steps are elaborated at much greater length in the form of a booklet titled 'Seven Steps to Modelling' that comprises some 16 pages and which is intended to scaffold the model-building process until learners can be relied upon to apply the steps independently.

### Step 1: Decide on the purpose of the model

In a classification domain, the purpose of a model is always to categorise things. The key questions to ask of learners are the obvious ones: What are the categories? What are some examples of things that belong to each category?

### Step 2: Identify decision factors

Decision factors are attributes (information types) that can be used to distinguish different categories. Learners are invited to identify decision factors that distinguish the examples of things which they named previously and also to identify the values of these factors.

### Step 3: Select a form of representation

Examples of classification trees, decision trees, factor tables and rules are reviewed and guidelines are offered on how to select between these forms. Learners are invited to consider these in relation to their own intended model, using a checklist of basic criteria.

### Step 4: Review the design

At this step learners itemise their decisions to date, including the purpose of the model, the names and values of decision factors, and the identity of the selected representational form.

### Step 5: Start the model

The advice that is given here varies according to the selected representational form. For instance, if a classification tree has been selected then the advice is: Create a new tree with a single node and label the node to identify the kinds of thing that it is the model's purpose to describe.

### Step 6: Develop the model

Learners follow a flowchart for which the main loop contains the steps (i) add information (ii) test (iii) revise. The 'add information' step is supported by advice about how the selected representational form may typically 'grow'. For instance, in the case of classification trees, the advice refers to the kind of questions that would be asked in a laddered grid interview [24] such as those which elicit the distinguishing features of sibling classes. Each representational form has its own associated advice; essentially, the properties of the selected form are exploited to drive forward the acquisition of knowledge. The 'revise' step may include changing information that was provided earlier or switching to a different representational form. A final step may be 'slimming' the model to create an optimally efficient classification procedure.

### Step 7: Reflect and evaluate

In this final phase, learners are encouraged to demonstrate and discuss their models with others. They also use simple check lists to self-evaluate their performance in the previous steps and to summarise their own learning.

## 4. Evaluation and discussion

An experimental study which has been fully reported by the present author elsewhere [8,14] confirms the worth of the non-rule forms of representation provided by InterModeller. In that study, 82 children each aged 15 years undertook a modelling course occupying roughly eight hours of class time. Analysis of the 632 models that resulted showed that rule models were almost never as high in quality (measured by indices of correctness, efficiency, and conciseness) as those built using the alternative factor table, decision tree and classification tree representations. Questionnaire responses indicated that children least enjoyed working with rules. That study also reported that children's ability to construct representations of classification improved significantly as a result of the modelling course.

Several reasons can be suggested for the superiority of the non-rule forms of representation. First, children encounter them in the curriculum more commonly than they do rules. Second, the non-rule forms give rise to models with a manifestly coherent structure; the coherence of a rule model is much less obvious. Third, and related to this, is the fact that as a non-rule model develops its evident incompleteness provides clues about what knowledge still needs to be acquired and incorporated. Fourth, rule models are probably the most difficult type to validate mentally. They are also the least forgiving of error. Minor typographical slips for example are hard to detect in a collection of rules but they can have a severe effect on how the rules behave at runtime.

Two further studies confirm that skills in classification modelling are both complex and educationally significant. The first was a small-scale observational study of 10-year old children using prototypes of the table and tree editing tools that were eventually built into InterModeller [8]. Children were adept at tool use but they experienced real conceptual difficulties in understanding a domain, selecting a representational form, and building coherent information structures. The second was a study in which pupils aged 16 were given paper-and-pencil tasks which tested their ability to make use of InterModeller-type representations [25]. In general pupils were quite good at interpreting given representations but they were less proficient at creating them from source material. Skill in creating trees and tables was found to correlate significantly with pupils' subsequent attainment in Mathematics at the Scottish Standard Grade.

A different kind of measure of InterModeller is its acceptance by schools. At the time of writing, one year after the launch of InterModeller for Macintosh but only a month after the release of the Windows version, approximately 70 of Scotland's 460 secondary schools have purchased licences for the software.

## 5. Conclusion and future work

InterModeller shows one way (almost certainly it is not the only way) in which to design a cognitive tool for classification learning. It provides an environment that has made classification model building a realistic proposition for a wide range of Primary and Secondary school children. There is evidence that children learn from their time spent with InterModeller. Further, there are signs that the software is acceptable to teachers. It may become quite widely adopted by schools.

A key design decision has been the provision of a variety of forms of representation and this has been justified by our evaluations of learners' experiences. Multiple forms of representation that are carefully selected and well integrated can enrich the model-building process. That design decision strongly influenced the work that followed. For once learners have a choice of representation, they may need advice on how that choice should be exercised and they may need to be helped when their chosen form proves to be unsuitable. Accordingly InterModeller provides automatic transformation between representational forms and the 'seven steps' approach was devised as a classroom knowledge engineering methodology. These aspects however require further evaluation.

Future work could take a number of directions including the following.

- A version of the 'seven steps' methodology might be incorporated into InterModeller in a variety of ways, for example as an online help assistant.
- Previous work on automated model analysers showed the potential of what might be called meta-tools to critique a learner's classification model, both for the purpose of teacher assessment and also to stimulate learners into refining their ideas [26].
- Knowledge acquisition techniques and tools could be developed to help learners to conceptualise a domain. Some work along these lines has been undertaken already and found promising [8,13].

It is tempting to add ever more technology to the software but ultimately this could defeat the main purpose. Perhaps one day a program will be developed that can build automatically a classification model from some scanned natural language text. Such a program would be a marvel of artificial intelligence but as a cognitive tool, one that left the learners in charge whilst engaging them in constructive thinking, it would be totally useless.

## 6. References


[1] InterModeller is available for Windows and Macintosh computers and is distributed by Parallel Logic Programming Ltd, UK. Evaluation copies may be downloaded from the PLP web site at www.parlog.com.

[2] Bruner, J., Goodnow, J. & Austin, G. (1956). *A Study of Thinking.* New York: John Wiley.

[3] Klausmeier, H., Ghatala, E. & Frayer, D. (1974). *Conceptual Learning and Development: a Cognitive View.* New York: Academic Press.

[4] Homa, D. (1984). *On the nature of categories.* In: Bower, G. (Ed). *The Psychology of Learning and Motivation.* New York: Academic Press.

[5] Inhelder, B. & Piaget, J. (1964). *The early growth of logic in the child.* New York: Norton.

[6] Greene, T. (1989). *Children's Understanding of Class Inclusion Hierarchies: The relationship between External Representation and Task Performance.* Journal of Experimental Child Psychology 48, 62-89.

[7] Keating, D. (1980). Thinking processes in adolescence. In J. Adelson, (Ed) Handbook of adolescent psychology pp211-246. New York: Wiley. Cited in Bee, H. (1992) *The Developing Child,* p270. New York: Harper Collins.

[8] Conlon, T. (1997). *Beyond Rules: The Development and Evaluation of Knowledge Acquisition Systems for Educational Knowledge-based Modelling.* PhD thesis, Department of Artificial Intelligence, University of Edinburgh.

[9] Jonassen, D. (1991). What are cognitive tools? In Kommers, M., Jonassen, D, & Mayes, J. (Eds) *Cognitive tools for learning computers and system sciences* (vol 81). Berlin: Springer-Verlag.

[10] Salomon, G. (1993). *On the Nature of Pedagogic Computer Tools: The Case of the Writing Partner.* In: Lajoie, S. & Derry, S. (Eds) *Computers as Cognitive Tools.* Hillsdale NJ: Erlbaum.

[11] Collins, A., Brown, J. & Newman, S. (1989). Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing, and Mathematics. In: Resnick, L. (Ed) (1989). *Knowing, Learning and Instructing: Essays in honour of Robert Glaser.* Lawrence Erlbaum.

[12] Conlon, T. & Bowman, N. (1995). *Expert Systems, Shells, and Schools: Present Practice, Future Prospects.* Instructional Science Vol. 23 Nos 1-3 pp111-131.
[13] Conlon, T. (1997). *Towards diversity: advancing knowledge-based modelling with knowledge acquisition.* In Brna P., & Dicheva, D. Proceedings of PEG97, University of Sofia, Bulgaria.
[14] Conlon, T. (1999). *Alternatives to Rules for Knowledge-based Modelling.* Instructional Science Vol 27 No 6 pp403-430.
[15] Tansley, D., & Hayball, C. (1993). *Knowledge-Based Systems Analysis and Design: A KADS Developer's Handbook.* Hemel Hempstead:Prentice-Hall.
[16] Clancey, W. (1985). Heuristic Classification. *Artificial Intelligence,* 27, 289-350.
[17] Conlon, T. & Pain, H. (1996). *Persistent Collaboration: A Methodology for Applied AIED.* Journal of Artificial Intelligence in Education, Vol 7 No. 3/4 219-252.
[18] Moss, C. (1994). *Prolog++: The Power of Object-Oriented and Logic Programming.* Wokingham: Addison-Wesley.
[19] LPA (1995). *Prolog++ Technical Reference.* Logic Programming Associates, London.
[20] At the time of writing, factor tables are provided only by the Macintosh version of InterModeller.
[21] Paterson, A. & Niblett, T. (1982). *ACLS User Manual.* Glasgow: Intelligent Terminals Ltd.
[22] Durkin, J. (1994) *Expert Systems Design and Development.* Macmillan.
[23] Webb, M. (1992). *Learning by Building Rule-based Models.* Computers Educ. Vol 18 No 1-3 pp89-100.
[24] Burton, A., Shadbolt, N., Rugg, G. & Hedgecock, A. (1988). *Knowledge Elicitation Techniques in Classification Domains.* In Kodratoff, Y. (Ed). *Proceedings of the 8th European Conference on AI.* London: Pitman.
[25] Conlon, T. Cox, R., Lee, J., McKendree, J. & Stenning, K. (1999). *Investigating representational competence in secondary school students.* Proceedings of the 1999 International Conference on Artificial Intelligence in Education (AIED'99), Le Mans.
[26] Conlon, T. (1995). *Automated Analysis for Knowledge-based Modelling.* In: Vanneste, P., Bertels, K., & De Decker, B. (Eds) *Proceedings of the Workshop on Automated (Novice) Program Analysis.* AIED 95: World Conference on Artificial Intelligence in Education, Washington DC. pp31-37. Also available as *DAI Research Paper No. 757*, Department of Artificial Intelligence, University of Edinburgh.
[27] This model is provided as the file 'Erosion' in the Examples folder with the InterModeller distribution software.