Pre-publication draft of avpaper which appeared in User Modelling and User-Adapted Interaction, 5 (2) pp 117-150 1996. Springer.

Feature Based Modelling: A methodology for producing coherent, consistent, dynamically changing models of agents' competencies.

GEOFFREY I. WEBB and MARK KUZMYCZ
School of Computing Mathematics, Deakin University, Geelong Vic 3217, Australia.
E-mail: webb@deakin.edu.au

Abstract. Feature Based Modelling uses attribute value machine learning techniques to model an agent's competency. This is achieved by creating a model describing the relationships between the features of the agent's actions and of the contexts in which those actions are performed. This paper describes techniques that have been developed for creating these models and for extracting key information therefrom. An overview is provided of previous studies that have evaluated the application of Feature Based Modelling in a number of educational contexts including piano keyboard playing, the unification of Prolog terms and elementary subtraction. These studies have demonstrated that the approach is applicable to a wide spectrum of domains. Classroom use has demonstrated the low computational overheads of the technique. A new study of the application of the approach to modelling elementary subtraction skills is presented. The approach demonstrates accuracy in excess of 90% when predicting student solutions. It also demonstrates the ability to identify and model student's buggy arithmetic procedures.

Key words: Student modelling, machine learning, modelling competency

1 INTRODUCTION

Most previous approaches to modeling agent's competencies have sought to develop *process models* - models of the internal processes that underlie those competencies (for example, Anderson, Boyle, Corbett and Lewis, 1990; Brown and Burton, 1978; Clancey, 1987; Corbett and Anderson, 1992; Goldstein, 1979; Ikeda, Kono and Mizoguchi, 1993; Langley, Wogulis and Ohlsson, 1990; London, 1992; Martin and VanLehn, 1993; Sleeman, 1987; Stevens, Collins and Goldin, 1982; VanLehn, 1986; Young and O'Shea, 1981). In contrast, input-output agent modeling models an agent's competencies without seeking to describe the internal processes that produce those competencies. The models produced can be considered to directly map the relationships between the inputs and outputs to the agent.

Feature Based Modeling (FBM), a form of input-output agent modeling based on attribute-value machine learning, has demonstrated that it is possible within reasonable computational constraints to produce models of agents' competencies with high predictive accuracy. FBM models can be updated in computational time spans measured in CPU seconds. It is thus feasible to use FBM in an interactive setting. Evaluation of FBM in the domain of elementary subtraction has demonstrated high accuracy in the prediction of students' precise answers. No previous student modelling system has been tested in the context of making precise predictions of students' answers, let alone demonstrated such levels of predictive accuracy.

A number of previous short papers have examined aspects of FBM (Amato and Tsang, 1990; Kuzmycz and Webb 1991; Webb, 1991; Webb. Cumming, Richards and Yum, 1989). This paper brings together the key findings of this previous work, provides greater detail of the methodology and its implementation, examines the broader implications of the approach and presents the results of a new study evaluating the effectiveness of the approach for modelling competency in elementary subtraction.

2 OVERVIEW

Attempts to model the internal operation of the cognitive system are fraught with difficulties. First, cognitive science is still in its infancy and the precise mechanisms by which the cognitive system operates are not well understood. This alone makes it extremely difficult to produce models of the internal operation of the cognitive system. Additional difficulties are introduced by inability to observe internal cognitive operations. In consequence, any hypothesis about a single internal event must be based on a chain of further hypotheses about other unobserved internal events which eventually leads back to an observed external event. The more detailed the model, the longer such hypothesis chains must stretch and the more tenuous they must become.

Model tracing (Anderson, Boyle and Reiser, 1985) circumvents these problems by modelling the agent at the level of observable operations. The model is a process model in that the elements of the model are intended to map directly onto internal cognitive processes. However, the model is also an input-output agent model in that each element of the model directly relates a situation or input to an action or output.

FBM also circumvents these difficulties by producing a model that is grounded in observable events and which does not attempt to create an explicit representation of unobservable events. This should not be taken as embracing philosophical behaviorism. To use FBM is not to deny that cognitive events occur. Rather, FBM can be viewed as embodying a form of weak methodological behaviorism in that it is based on the assumption that it is possible to obtain useful information about the cognitive system without understanding its internal mechanisms.

The key difference between model tracing and FBM is that the former relies upon a library of situation-action associations (the rules from which a model is constructed) whereas FBM generates these associations as required. This makes FBM more flexible in that it does not require the system designer to anticipate all situation-action associations that may be relevant to modelling a student.

An FBM model can be considered as a description of an agent's competency. It can be used to explain and to predict an agent's capabilities. However, it does not explain how that competency is achieved, nor why alternative competencies are not obtained.

Such a model is a cognitive model in the sense that it is able to model the performance of the cognitive system, albeit, a 'black box' model. However, it is not a cognitive model in the sense of a model of the operation of the cognitive system.

It is undoubtedly true that an accurate model of the precise internal operation of the cognitive system will be more powerful and more useful for many purposes than a model that does not contain this information. However, it is likely that, in at least some contexts, an accurate model of competency will have greater utility than a less accurate process model. A further reason for wishing to develop the less complex form of model is that the computational overheads involved in its construction are lower, increasing its utility, particularly in an interactive environment. While the computational overheads involved in model tracing are lower than those of FBM, as model tracing considers a more circumscribed set of situation-action associations, FBM places lower demands on the system developer. A further consideration is that there is evidence that human tutors only rarely attempt to construct process models (Putnam, 1987; McArthur, Stasz and Zmuidzinas, 1990).

FBM has been developed for purposes of student modelling. However, the technique is applicable to the much broader contexts of user modelling and the modelling of agents in general. This paper concentrates on issues relating to the application of FBM to student modelling. This should be taken as a reflection of the environment in which the methodology has been developed rather than a reflection of the potential scope of its application.

3 CONSTRUCTING THE MODEL

For the purpose of illustration, examples will be drawn from the domain of elementary subtraction skills. This domain will be used for this purpose throughout the rest of this paper. The approach to modelling this domain that has been selected treats the examination of a single column as the elementary unit of analysis.

3.1 The Form of the Model

FBM produces a model of an individual agent on the basis of observations of that agent's performance. This model relates the contexts in which actions are performed to aspects of those actions. Both contexts and actions are described in terms of their relevant features. Context features describe properties of the context in which an action is performed. For the example approach to modelling subtraction competency, the context features describe the column and the context in which that column appears. Such features might include indications that the subtrahend of the current column is smaller than the minuend and that the subtrahend is zero.

Action features describe properties of the agent's actions. Action features in our example approach to modelling elementary subtraction skills represent properties of the student's answer for a single column. These might include features indicating that the result equals the minuend and that the result represents a correct solution. Appendix A presents the complete set of context and action features used in the Subtraction Modeller, an FBM system for modelling elementary subtraction skills that will be described in more detail below.

Features are a very general formalism. They may range from the very high level, such as *The answer is correct,* through intermediate levels, such as *The answer is the minuend minus the subtrahend*, to the very low level, such as *The answer is zero*. Features are currently restricted to categorical values. However, there is no reason in principle to prevent the extension of the approach to ordinal values.

An association $X \to a$ is a relationship between a set of context features, X, and a single action feature, a. An association indicates that action feature a applies to the agent's actions in context X. For example, the association *{the subtrahend is Zero}* $\to$ *the result equals the minuend* indicates that when the subtrahend is zero the agent provides the minuend as the result. An FBM model consists of a set of associations.

For some purposes it is useful to divide associations into those that are appropriate and those that are not. An association is appropriate if it is considered desirable for the agent to embody it. For example, the association *{the subtrahend is zero, the subtrahend to the immediate left is less than the minuend to the immediate left}→ the result equals the minuend* is appropriate if correct arithmetic performance is desired. Associations that are not appropriate are called erroneous associations. Appropriate and inappropriate associations correspond respectively to rules and mal-rules (Sleeman, 1982). However, unlike rules and mal-rules, there is no implication that associations correspond directly to discrete internal cognitive operations. The methods for judging whether an association is appropriate or erroneous must be provided by the instructor creating a lesson, or, in more general terms, the person creating the modelling system.

3.2 Features and the Feature Network

The relationships between features are described by a feature network (Webb, 1988). The central type of element of a feature network is a feature choice. This can be viewed, in machine learning parlance, as an attribute. Features, in machine learning terminology, are categorical attribute values. Every feature must belong to exactly one feature choice. The other features that belong to the same feature choice as a feature *f* are called *f*'s siblings. Siblings are mutually exclusive. That is, no two siblings may apply to the same context or action.

A feature network also specifies dependency relationships between features and feature choices. From these it is possible to derive generalisation and specialisation relationships between features. Feature *a* is a generalisation of feature *b* if and only if *a* must apply to a context or action if *b* applies to that context or action. For example, the context feature *the subtrahend is less than or equal to the minuend* is a generalisation of *the subtrahend equals the minuend*.

It is useful to augment the feature network with inter-feature-set generalisation relationships. For example, *{the result equals the minuend}* is a generalisation of *{the subtrahend is zero, the result equals the minuend minus the subtrahend}*. The former must apply to all contexts to which the latter applies. The feature choices and generalisation relationships can be used to constrain the associations that need be considered by the modelling system. This affects implementation efficiency only. This is the only role of the feature network in FBM.

3.3 Constraining the Associations That Need To Be Considered

Two types of constraint are employed. First, it is not possible to have a set of context features that contains a feature *f* and any feature *x* such that a generalisation of *x* is a sibling of *f*. Thus, it is not possible to have a set of context features that contains the features *the subtrahend is less than the minuend* and *the subtrahend is equal to the minuend*, because the latter feature is a specialisation of *the subtrahend is greater than or equal to the minuend*, which is a sibling of the former feature. All such combinations of features can be removed from consideration.

Second, it is possible to further constrain the possible associations by deleting any set of context features that contain a set of features *A* and its generalisation *B*. An example of such a set of context features is *{the subtrahend is less than the minuend, the subtrahend is less than or equal to the minuend}* in which the second feature is a generalisation of the first. Such a set of context features will always apply to exactly the same contexts as the same set with the more general feature deleted. Thus, the descriptive power of the two expressions are identical. The addition of the more general feature adds nothing but syntactic complexity to the model. In consequence, the descriptive power of the modelling system is not affected by removing all such feature sets from consideration.

3.4 Handling Noise

Any practical cognitive modelling system must allow for the existence of noise. Noise may result from inaccuracies in data collection or from slips or lapses in concentration resulting in behavior that is not representative of the agent's general performance.

FBM allows for noise by accepting an association $C \to a$ if and only if

1. $\#(C \to a) \geq \min\_\ evidence$;

2. $\dfrac{\#(C \to a)}{\#(C \to a) + \#(C \to\sim a)} \geq \min\_\ accurancy$; and

3. there is no association between a specialisation of C and a sibling of a.

where

• $\#(C \to a)$ is the number of observed cases in which all features in C and feature a have been present;

• $\#(C \to\sim a)$ is the number of observed cases in which all features in C and a sibling of a have been present; and

• *min_evidence* and *min_accuracy* are implementation dependent parameters.

Most implementations of FBM have used *min_evidence* set to 3 and *min_accuracy* set to 0.8. Although *min_accuracy* of 0.8 ostensibly allows for an association to be accepted when almost 20% of the evidence contradicts it, clause 3 limits the probability of this occurring by suppressing an association if there is a regularity detected in the contrary evidence.

3.5 Handling Concept Change

Another factor of which a practical modelling system must take account is that an agent's approach to a domain may change over time. Indeed, in an educational environment, the aim of the interaction is precisely to achieve such a transformation. FBM can accommodate change by placing less weight on old evidence in contrast to new evidence. This process is called data aging.

When an agent is observed to act in manner a in context C this is initially considered as one unit of confirmation for the association $C \rightarrow a$. Similarly, when an agent is observed to act in manner $\sim a$ (where $\sim a$ is a sibling of a) in context C this is initially considered as one unit of disconfirmation for $C \rightarrow a$. The evidence for and against an association can be described by two numbers, a tally of confirmations and a tally of disconfirmations. When data aging is employed, every time that an action is observed in a context described by C, all tallies relating to C are discounted through multiplication by a set discounting rate. Most implementations of FBM to date have used a discounting rate of 0.9. The tallies are then updated as appropriate in accord with the new evidence. When determining whether an association should be accepted, the aged tallies can be employed in place of $\# C \rightarrow a$ and $\# C \rightarrow \sim a$. As a result, the model will take greater account of recent actions than of earlier actions.

3.6 Selecting Features

It is up to the instructional designer (or other developer of the modelling system) to identify the relevant context and action features for a given modelling task. While the theoretical foundations on which FBM are based suggest that context and action features should be observable properties of the agent's environment and actions, respectively, the methodology is not able to enforce this. The system developer is able to include any properties that he or she wishes. These might include internal cognitive states that the developer believes should or could be generated in a particular context. For example, in the subtraction domain, one context feature might be that the current problem requires carry (where carry is intended to be an internal cognitive operation performed to generate a solution). As can be seen by examination of the features listed in Appendix A, the Subtraction Modeller does not use such features.

In the systems developed to date, context features have exclusively described properties of the task on which a student is currently engaged. It is intended, however, that any aspect of the environment in which an action occurs could be described by a context feature. For example, relevant context features in the domain of elementary subtraction might include details of the answers provided by students seated close to the student whose subtraction competency is being modelled.

It is of course, likely that some students will not attend to all context features that are made available to the modelling system. In this case, those features should not be observed to influence the student's actions and should not appear in the associations that are identified. It is also likely that some students will attend to factors that are not described by the context features available to the modelling system. For example, a student solving subtraction problems might take account of the answers being written by another student when those details are not available to the modelling system. In this case, the system should not be able to detect associations between the available context and action features, or the associations detected will describe a complex system of which the agent is but one part. In the example of the student attending to another student's work, the complex system will be the pair of students.

3.7 Constructing the Model

To reduce computational overheads, before use, all possible sets of context features for a domain can be computed. From these, all invalid and redundant sets of features can be removed. A set of features is invalid if it contains a feature *f* and either a sibling of *f* or a specialisation of a sibling of *f*. A set of features is redundant if it contains a set of features *F* and a set of features that is a specialisation of *F*. Each remaining set of features can be provided a unique numeric identifier that is used as an index into an array that records the entire model. This array is indexed in one dimension by a set of context features and in the other dimension by an action feature. Each element of the array contains the data required to determine whether there is an association between that set of context features and action feature. Thus, the model can be represented by a single large static direct access data structure.

For every most specific set of context features, a list of all generalisations is generated. This enables rapid update of statistics within the model by removing the need to compute and locate valid generalisations during operation.

Current implementations of FBM employ exhaustive analysis when creating and updating a model. It is feasible to construct a model containing all supported associations in a moderately complex domain by constructing a two dimensional array indexed in one dimension by the valid sets of context features and in the other dimension by the action features. Each cell contains both the aged tallies corresponding to $\#(C \rightarrow a)$ and $\#(C \rightarrow\sim a)$ along with a flag indicating whether the association is accepted. When a new action is processed by the modelling system it is only necessary to update each cell indexed by a subset of the set of context features that describe the context of the action. Thus the computational complexity of updating the model is of the order 2^n, where n is the number of context features describing the current action. While this complexity is exponential, the magnitude of n is often sufficiently constrained for this not to be a significant problem. Indeed, for the Unification Tutor, an interactive system employing this approach to updating an FBM model that is described in more detail below, the time taken to update the agent model is generally not noticeable.

The model created by FBM is similar to a version space (Mitchell, 1977) in that it contains all associations that are consistent with the evidence (after allowance for the constraints specified above). However, FBM differs substantially from the version space machine learning algorithm in that it supports disjunction (a model is composed of multiple disjoint rules), allows hypotheses that are inconsistent with the training data (through the allowance for noise) and explicitly considers all hypotheses within the version space. It is further distinguished by the selection of different subsets of the version space for use in different contexts. This selection process is described below.

FBM is distinguished from most previous machine learning techniques by the manner in which it develops all rules that describe the training data from which different subsets are selected for different purposes, as described below. By way of contrast, most machine learning systems select a single highly restricted subset of the rules that describe the training data. In most cases, this restricted subset includes the minimum number of rules necessary to describe all examples.

FBM does not require a sophisticated learning algorithm. All that is required is a straight forward sequential update of a table of tallies of supporting and counter evidence for the associations.

A separate model is maintained of each agent. Every time that an action by that agent is observed the model is updated accordingly. The model may be consulted at any time. The model can be expected to grow in accuracy as the number of observed actions increases.

3.8 The Meaning of an Association

The associations within an FBM model can be considered to be production rules. They differ, however, from the form of production rule commonly formed by modelling systems such as ACM, BUGGY or model tracing, in that the consequents are features of actions (or partial specifications of actions) rather than complete actions. To predict precise actions it is necessary to consult all active associations which will lead to the specification of a set of action features. This set of features may allow the precise identification of a specific action. Alternatively, it might be able to constrain the set of possible actions without identifying one specific action. This latter situation is likely if the student model is not well developed, perhaps because of lack of information, or if the student is not acting in a consistent manner, perhaps due to a change of problem solving strategy or the failure to adopt a consistent approach to the problems being examined.

Associations are flat rules. There is no form of chaining from the consequent of one rule to the antecedents of another. However, it is possible to construct complex models containing considerable relevant internal structure. First, many associations may relate to a single action feature. Thus, it is possible to form disjunctive descriptions of a student's competency. Second, it is possible to have both action and context features of differing levels of generality. This enables the formation and application of the model at differing levels of granularity.

4 USING THE MODEL

Depending upon the intended use for the model, it may be desirable to extract a more concise set of associations than those included in the complete model. That is, when using the model, rather than manipulating every association that is supported by the evidence, it may be useful to consider only a select subset of those associations that capture key aspects of the system's understanding of the agent. For example, if association $C \rightarrow a$ is in the full model then every association $X \rightarrow a$, such that X is a specialisation of C is likely to be in the model, unless there is insufficient or inconsistent evidence with regard to that specialisation. If the modelling system is not prepared to accept some specialisations, $X \rightarrow a$, then it should not be prepared to accept $C \rightarrow a$, as the latter implies the former. If it does accept the more general form then the more specific form is implied and need not be considered when applying or describing the model.

Appropriate strategies for simplifying the model will vary depending upon the intended use. For example, if the modelling system is being used to control interactions with the agent, it may be desirable to consider only the most specific associations that the system can form. These will correspond to the finest grained discrimination between tasks that the system is capable of supporting. However, they will capture very little generalisation beyond the situations that have been observed and will be unnecessarily complex.

4.1 Simplifying the Model for Communication to a User

If aspects of the model are to be communicated to a human, it is desirable to select strategic associations that best summarise the model. The best associations to select will depend upon the particular application. However, if very general associations are presented the system will often appear to be jumping to conclusions far beyond those supported by the available evidence. For example, it would not generally be appropriate for the system, after seeing an agent perform only three simple tasks, all of which are performed appropriately, to announce that it appeared that the agent always performed any task for the domain appropriately.

A useful type of association to work with is the *most specific highest supported association.* An association $C \rightarrow a$ is a *most specific equivalent supported association* if and only if

• it is accepted;

• there is no other accepted association $X \rightarrow b$ such that C is a generalisation of X, a is a generalisation of b and $\#(X \rightarrow b) = \#(C \rightarrow a)$; and

An association $C \rightarrow a$ is a *most specific highest supported association* if and only if

• it is a *most specific equivalent supported association*; and

• there is no other *most specific equivalent supported association* $G \rightarrow b$ such that G is a generalisation of C and $b = a$ or b is a generalisation of a.

Such an association generalises as far as the current evidence warrants but no further. Most specific highest supported associations are particularly relevant when communicating with a human user (in an educational context, either the student or the teacher) as they generalise from the available evidence without overgeneralising to unsustainable conclusions.

Figure 1 shows the most specific highest supported associations that might be obtained from an FBM model of subtraction competency for a student with the SMALLER/FROM/LARGER bug (VanLehn, 1986). As this demonstrates, such a simplified model is able to provide a clear and relatively concise description of the student's approach to the domain. (While the last three associations are all logically equivalent, the current implementation of the system is unable to detect this, resulting in the addition of a small amount of unnecessary syntactic complexity to the model.)

$$\{Minuend > Subtrahend\} \rightarrow$$
$$result = Minuend - Subtrahend$$

$$\{Minuend < Subtrahend\} \rightarrow$$
$$result = Substrahend - Minuend$$

$$\{Minuend = Subtrahend\} \rightarrow$$
$$result = Minuend - Subtrahend$$

$$\{Minuend = Subtrahend\} \rightarrow$$
$$result = Subtrahend - Minuend$$

$$\{Minuend = Subtrahend\} \rightarrow$$
$$result = 0$$

FIG. I. MOST SPECIFIC HIGHEST SUPPORTED ASSOCIATIONS FOR THE SMALLER/FROM/LARGER BUG.

4.2 Using the Model to Select Tasks for a User

It is sometimes desirable to use the model to select tasks for an agent to perform that will provide evidence that will help further refine the model. In this context, it is useful to consider potential associations rather than accepted associations. The investigation of two types of potential association will maximise refinement of the model - *most general insufficient evidence* and *most specific inconsistent evidence associations.*

An association $C \rightarrow a$ is a most general insufficient evidence association if and only if

- $\#(C \rightarrow a) <$ *min_evidence*; and
- there is no other association $X \rightarrow b$ such that *X* is a generalisation of *C*, *b*, is a generalisation of *a* and $\#(X \rightarrow b) <$ *min_evidence*.

If the educational system selects tasks for the student to consider, selecting tasks that will provide further evidence relating to most general insufficient evidence associations will serve to most rapidly expand the set of associations that are accepted into the model (Kuzmycz and Webb, 1991).

An association $C \rightarrow a$ is a most specific inconsistent evidence association if and only if

- $\#(C \rightarrow a) \geq$ *min_evidence*; and
- 1-*min_accuracy*$< \dfrac{\#(C \rightarrow a)}{\#(C \rightarrow a)+\#(C \rightarrow \sim a)} <$ *min_accuracy*; and
- there is no other set of features *X* such that *C* is a generalisation of *X*.

The existence of a most specific inconsistent evidence association indicates one of three things. One possibility is that the available evidence contains noise, in which case more evidence may be required in order to enable the system to filter out that noise. Another possibility is that the available features do not permit the construction of an accurate model of the agent. The final possibility is that the agent has changed approach to the domain and is now performing in a different manner with regard to the contexts in question.

A hypothesis that noise is the cause of a most specific inconsistent evidence association is plausible if $\#(C \rightarrow \sim a)$ is small. Alternatively, it is credible that such an association arises from an insufficient model if the contrary evidence is widely distributed across time. The third alternative, that the agent's approach to the domain is changing, is plausible if the contrary evidence (support for $C \rightarrow \sim a$) is concentrated in an identifiable time period.

4.3 Applying the Model

Associations are equivalent to production rules. In consequence, it is possible to use a constructed model to solve problems. Depending upon the specificity of the action features that appear in the associations, execution of the model may result in either full or partial solutions. The ability to construct partial solutions means that the modelling system is able to predict aspects of an agent's actions even when there is insufficient evidence available to predict the precise actions.

An FBM model can be used to describe the agent's competency in the domain, predict the agent's future actions and/or assist a computer-based system to manage interactions with the agent. Examples of each of these uses of an FBM model are provided below.

5 SUMMARY OF THE MODELLING PROCESS

To summarise, the process of constructing an FBM model, as it is currently implemented, is as follows.

1. The system developer identifies suitable context and action features along with an augmented feature network that describes the relationships between those features.
2. The system developer creates mechanisms to allow the system to identify the context and action features that characterize agents' actions.
3. The augmented feature network is compiled to generate a two dimensional static array that will contain the system's model of an agent. This array is indexed in one dimension by sets of context features and in the other by individual action features. Each entry is a tally of confirmations and disconfirmations of associations between the respective set of context features and the action feature.
4. As agent actions are observed the model is updated incrementally. Each update alters the tallies for every entry for a subset of the set of context features that describe the context of the current action as follows:

   a. If data aging is employed, the tallies are aged.
   b. The confirmation tallies are incremented for all entries indexed by action features that are known to be present.

c. The disconfirmation tallies are incremented for all entries indexed by action features that are known not to be present.

5. The model may be consulted at any time. When consulted, suitable simplified summaries of the model may be derived.

6 AN EXAMPLE

The following is a simplified example of the application of FBM. In keeping with the examples presented so far, a model is to be constructed of competency in three column subtraction. Each column of a subtraction problem is treated as a separate task for the purposes of constructing the model. The following context features are employed.

- The minuend is less than the subtrahend (M < S)
- The minuend equals the subtrahend (M = S)
- The minuend is greater than the subtrahend (M > S)

Problem		256 -162 114			834 -184 750			515 -468 153	
Context	$M > S$ $MR < SR$	$M < S$ $MR \geq SR$	$M > S$	$M > S$ $MR < SR$	$M < S$ $MR \geq SR$	$M = S$	$M > S$ $MR < SR$	$M < S$ $MR < SR$	$M < S$
Action	$R = M - S$ $R \neq S - M$	$R \neq M - S$ $R = S - M$	$R = M - S$ $R \neq S - M$	$R = M - S$ $R \neq S - M$	$R \neq M - S$ $R = S - M$	$R = M - S$ $R = S - M$	$R = M - S$ $R \neq S - M$	$R \neq M - S$ $R = S - M$	$R \neq M - S$ $R = S - M$

Problem		246 -124 122			527 -449 122	
Context	$M > S$ $MR \geq SR$	$M > S$ $MR \geq SR$	$M > S$	$M > S$ $MR < SR$	$M < S$ $MR < SR$	$M < S$
Action	$R = M - S$ $R \neq S - M$	$R = M - S$ $R \neq S - M$	$R = M - S$ $R \neq S - M$	$R = M - S$ $R \neq S - M$	$R \neq M - S$ $R = S - M$	$R \neq M - S$ $R = S - M$

Note: the action features $R \neq M + 10 - S$ and $R \neq M - S - 1$ apply to all columns and have not been listed above.

FIG. 2. FIVE EXAMPLE SUBTRACTION PROBLEM SOLUTIONS AND ASSOCIATED FEATURES.

- In the column to the immediate right, the minuend is greater than or equal to the subtrahend ($MR \geq SR$)
- In the column to the immediate right, the minuend is less than the subtrahend ($MR < SR$)

The following action features are employed.

- Result equals minuend minus subtrahend (*R*=*M*-*S*)
- Result equals subtrahend minus minuend (*R*=*S*-*M*)
- Result equals minuend plus 10 minus subtrahend (*R*=*M*+*10*-*S*)
- Result equals minuend minus subtrahend (*R*=*M*-*S*)

Note that there are siblings to each of these action features ($R \neq M - S, R \neq S - M, R \neq M + 10 - S$ and $R \neq M - S$) but that in this case the siblings are always present whenever the originals are not and vice versa. These siblings are not presented as they add little to the understanding of the approach but do much to clutter its presentation.

This set of features is not sufficient to represent correct subtraction skills or a large range of incorrect approaches to subtraction. It serves, however, to illustrate the basic principles of FBM.

For the sake of simplicity, we will not employ data aging in this example.

We will illustrate the generation of a model with respect to the set of three column subtraction problems and answers presented in Figure 2. In this figure, immediately below each subtraction problem is a summary of the features for each of the columns of the problem. The left-most, centre and right-most columns of this summary correspond respectively to the left-most, centre and right-most columns of the problem.

TABLE 1. TABLE OF EVIDENCE IN SUPPORT OF EACH ASSOCIATION

	R=M-S		**R=S-M**		**R=M+10-S**		**R=M-S-1**	
	$\#(C \to a)$	$\#(C \to \sim a)$	$\#(C \to a)$	$\#(C \to \sim a)$	$\#(C \to a)$	$\#(C \to \sim a)$	$\#(C \to a)$	$\#(C \to \sim a)$
$\{M < S\}$	0	6	6	0	0	6	0	6
$\{M = S\}$	1	0	1	0	0	1	0	1
$\{M > S\}$	8	0	0	8	0	8	0	8
$\{MR \geq SR\}$	2	2	2	2	0	4	0	4
$\{MR < SR\}$	4	2	2	4	0	6	0	6
$\{M < S, MR \geq SR\}$	0	2	2	0	0	2	0	2
$\{M = S, MR \geq SR\}$	0	0	0	0	0	0	0	0
$\{M > S, MR \geq SR\}$	2	0	0	2	0	2	0	2
$\{M < S, MR < SR\}$	0	2	2	0	0	2	0	2
$\{M = S, MR < SR\}$	0	0	0	0	0	0	0	0
$\{M >= S, MR < SR\}$	4	0	0	4	0	4	0	4

A table of evidence supporting all possible associations is formed as per Table 1. In this table, there is a row corresponding to each possible combination of context features. Note that some combinations are disallowed (such as {$M<S$, $M=S$}). The valid combinations are determined by the feature network and associated generalisation relationships (not presented here). There are two columns for each action feature. The columns labeled $\#(C \rightarrow a)$ contain counts of the numbers of times that the action features have appeared in conjunction with the relevant set of context features. The columns labeled $\#(C \rightarrow\sim a)$ contain counts of the numbers of times that the siblings of the action features have appeared in conjunction with the relevant set of context features.

For each of the fifteen columns, the relevant context and action features are determined. For the left-most column of the left-most problem these are $M > S$, $MR < SR$, $R = M - S$, $R \neq S - M$, $R \neq M+10 - S$ *and* $R \neq M - S - 1$. The table of evidence is then updated. In this case, the $\#(C \rightarrow a)$ values for $R=M - S$ get incremented for sets of features $\{M > S\}$, $\{MR < SR\}$ and $\{M > S, MR < SR\}$ (the action features $R \neq S\text{-}M$, $R \neq M\text{+}l0\text{-}S$ and $R \neq M\text{-}S\text{-}1$ are not maintained in this example). The $\#(C \rightarrow\sim a)$ values get incremented for all other action features in conjunction with these three context feature sets.

Table 1 represents the evidence after all fifteen columns have been so examined. Using a minimum evidence criterion of 3 and a consistency criterion of 0.8, an association is accepted if $(\#(C \rightarrow a) + \#(C \rightarrow\sim a)) \geq 3$ and $\#(C \rightarrow a)/(\#(C \rightarrow a) + \#(C \rightarrow\sim a)) \geq 0.8$.

The following associations satisfy these conditions:

- $\{M < S\} \rightarrow R = S - M$
- $\{M > S\} \rightarrow R = M - S$
- $\{M > S, MR < SR\} \rightarrow R = M - S$

Note that this model is redundant. The last association is a specialisation of the second. If the second is correct then the third must be correct. Note also that this model is incomplete. It cannot be used to predict the answers for columns in which the subtrahend equals the minuend.

All of these associations are most specific equivalent supported associations. Only the first two are most specific highest supported associations as the second is a generalisation of the third and the former has higher support than the latter.

The associations relating to the feature sets {M = S}, {M > S, MR ≥ SR}, {M < S, MR ≥ SR} and {M < S, MR < SR} are most general insufficient evidence associations. The examination of problems with these combinations of features has greatest potential to strengthen the model.

There are no most specific inconsistent evidence associations. While the associations $\{MR \geq SR\} \rightarrow R = M - S$; $\{MR \geq SR\} \rightarrow R = S - M$; $\{MR < SR\} \rightarrow R = M - S$; and $\{MR < SR\} \rightarrow R= S - M$ all have inconsistent evidence (the first two of the three criteria for most specific inconsistent evidence are satisfied) in each case there exist specialisations of the feature set ($\{M < S, MR \geq SR\}$, $\{M > S, MR \geq SR\}$, $\{M < S, MR < SR\}$ and $\{M > S, MR < SR\}$). In consequence, the third criterion is not satisfied. The absence of most specific inconsistent evidence associations suggests that the available features are adequate for constructing a model of the student's competency in the domain.

7 IMPLEMENTATIONS

FBM has, been implemented in four major test-bed systems. The first implementation was as part of an experimental generic shell for supporting lessons using feature networks as a knowledge representation formalism (Webb, 1988). The major system developed using this shell provided tuition in English word classes. This implementation of FBM served to demonstrate that the approach could develop plausible models and provided motivation for further research and evaluation.

Amato and Tsang (1990) developed a system for tutoring the playing of scales on a piano. In this system, context features described the appropriate tonic, hand motion, number of octaves, touch and tone of a scale. Action features described the observed tonic, hand motion, number of octaves, touch and tone of the student's attempt to play the scale. The model was used both to generate advice to the student and to select appropriate scales for the student to practice. This system has demonstrated the applicability of FBM to the training of motor skills.

The two most extensively studied implementations of FBM, however, are the Unification Tutor and the Subtraction Modeller. It is instructive to examine these systems in some detail,

7.1 The Unification Tutor

The Unification Tutor was developed to explore the application of FBM to non trivial problem solving skills (Webb, 1991). This system tutors students in the unification of terms from the Prolog programming language. It has been used over a period of four years by third year Computer Science students at both La Trobe and Deakin Universities.

A task in this domain consists of two Prolog terms to be unified. Context features describe the two terms and the relationships between them. Action features describe the solution that the student proposes.

For use in tutoring, the Unification Tutor needs to identify which associations are appropriate and which are inappropriate. It does this by observing the tasks that a student tackles. For each task the system is able to identify the appropriate action features (the features of a correct solution to the unification problem). An association is considered inappropriate if and only if it identifies an inappropriate action feature for one of the tasks that the student has examined. This scheme has two desirable consequences. First, it prevents the system from acting upon an inappropriate association before the agent has had a chance to demonstrate that he or she will not apply it in inappropriate circumstances. Second, it circumvents the need for the system designer to provide a alternative mechanism by which to identify which associations are appropriate and which are not.

The student model is used both to select problems for the student to examine and to provide feedback and advice. A refined model containing only inappropriate most specific highest supported associations is derived from the full student model.

Associations from the refined model are described to the student at opportune occasions. It is deemed to be an opportune occasion to describe an association $C \rightarrow a$ when

- the context features C apply to the most recent task;
- the action feature a applies to the agent's response to the task; and
- a was inappropriate for the task.

These conditions ensure that

- only inappropriate associations (associations that the tutor wishes to prevent the student from exhibiting) are examined;
- the student has the opportunity to observe the accuracy of the association in describing her approach to the domain;
- the student has the opportunity to observe that the association is inappropriate; and
- a concrete context is provided in which the association may be examined.

The explicit description of associations serves to highlight to the students the forms of error that the system is observing. It is up to the students to determine how their problem solving strategies lead to these forms of error and how those strategies may be repaired.

When it is not possible to provide such feedback, the system provides simple feedback indicating whether the answer is correct and, if the answer is incorrect, indicating how it is incorrect and supplying a correct answer generated by the machine. In early implementations of the system (including that examined by Webb, Cumming, Richards and Yum, 1990), this feedback was always generated and displayed while the system updated its model, thus minimising the apparent delays caused by the computation of the new model. However, the current implementation of the modelling system has sufficient computational efficiency that this is not necessary. The current system is able to update the student model in a matter of CPU seconds. This time delay is barely perceptible to the user. In consequence, it is able to determine whether an appropriate association can be discussed with the student and only resorts to a simple response if no such association exists.

A unification task is only presented to the student if the model does not indicate that it will be solved correctly and if the model indicates that the student has acquired sufficient sub-skills to be able to tackle it. The model will not indicate that a task can be solved correctly if either there is insufficient evidence to create appropriate associations that cover the task or if there are inappropriate associations that cover the task.

An ambitious study was undertaken to evaluate the educational utility of FBM within an early implementation of the Unification Tutor (Webb, Cumming, Richards and Yum, 1990). In this study two versions of the tutor were created. One version employed FBM and the other did not. The aim was to compare two very similar systems for which it was reasonable to attribute any difference in performance directly to the use of or failure to use FBM. Three forms of evaluation were performed. Students were asked to provide interactive evaluation of the system during its operation, they were asked to answer a questionnaire after its use and their performance on an end of year examination was also monitored.

Although the results in general favoured the use of FBM, few of the differences were statistically significant. In particular, although the average examination results of the students that used the version of the system with FBM were higher than those that did not, this difference was not statistically significant.

One of the key lessons to be learned from this study is that it is extremely difficult to evaluate a student modelling system. The experimental design did not directly test whether the modelling system constructed useful models. Rather, it tested whether the manner in which the models were utilised provided benefit. In such an experimental design a system that developed poor models could well fare better than a system that developed good models solely due to the manner in which those models were employed for tutoring.

A further issue highlighted by this study is that it is exceedingly difficult to construct an accurate model in an active educational context as the primary aim of an educational interaction is to transform an agent's approach to a domain. Thus, the modelling system is always trying to strike a moving target.

7.2 The Subtraction Modeller

Due to the difficulties experienced when evaluating FBM in the context of the Unification Tutor, it was decided that the next form of evaluation should concentrate on the predictive accuracy of FBM in a non-tutoring context.

Most previous evaluation of the power of cognitive modelling systems has taken the form of constructing a model from a set of examples and then evaluating that model's capacity to explain those examples (for example, Brown and Burton, 1978). Such evaluation provides no indication of the accuracy of the model. It evaluates only the ability of the modelling formalism to create a model that is consistent with the given data. By contrast, to evaluate predictive accuracy, models must be constructed from one set of examples and then evaluated in terms of their ability to predict new, previously unsighted, cases. The latter provides a far more rigorous form of evaluation than the former.

Evaluation of predictive accuracy should also be distinguished from the form of evaluation conducted by Corbett, Anderson and O'Brian (1993) in which the modelling system predicts the rate of error and evaluation compares the predicted and observed rates of error. Evaluation of predictive accuracy involves predicting the precise actions to be performed and comparing the predicted and observed actions.

To this end, a modelling system was developed for the domain of elementary subtraction. This is one of the 'classic' problems in student modelling (Attisha and Yazdani, 1984; Brown and Burton, 1978; Langley, Wogulis and Ohlsson, 1990; VanLehn, 1982; Young and O'Shea, 1981).

This system was distinguished from previous FBM modelling systems in that it was not embedded in a computer-based tutoring environment. Rather than providing tuition, it concentrated solely on the development and evaluation of models from tests. In consequence, it was expected that there would be less difficulty in evaluating the modelling system's performance as there would be less likelihood of student's approaches to the domain changing over the period of observation. The tests were printed and the students worked with pen and paper. Test preparation was decoupled from the modelling system so that different techniques for generating test problems could be employed. The features presented in Appendix A were employed by the system.

An initial study demonstrated average predictive accuracy of 92% when models constructed by the system were used to predict students' answers to subsequent problems (Kuzmycz and Webb, 1992).

This initial study raised a number of issues. Five tests containing 32 questions each were administered to the students with gaps between tests of one or two weeks. The students received ongoing tuition in the subject matter, and, in particular received tuition explicitly addressing problems with the subtraction skills revealed by the tests that they were receiving. There was some evidence that this tuition was lowering the system's predicative accuracy.

Another issue was how consistent were the student's in their subtraction performance. When they made mistakes, were these essentially random (slips); consistent in the contexts in which they occurred, but inconsistent in terms of responses (tinkering); or the result of a consistent buggy subtraction procedure?

A final issue was that the tests were generated individually for each student using a procedure that produced tests designed to improve the student model. In consequence, the tests contained problems for which the modelling system believed there was reason to suspect that the predictions might be poor. It was impossible to determine what effect this might have had on the system's predictive accuracy.

8 FURTHER EVALUATION OF THE SUBTRACTION MODELLER

A new study was conducted to explore these issues. The number of subtraction problems presented per test was increased to 40. This increased the number of consecutive solutions between which no tuition or extended time for procedure change was available.

Checks were added for consistency of errors. As these checks could affect student or system performance, two treatment groups were formed. Students in the random treatment received two randomly generated tests. Different tests were administered to each student. For each student in the Error Repeat treatment, each question for which a student produced an error on the first test was repeated in the second test. Randomly generated subtraction problems were then added to make a total of forty questions.

73 nine to ten year old students from three schools were tested. These were assigned to treatments by the following process. First, all subjects were sorted by school and then within each school, into alphabetical order by surname. The first subject was placed in the random treatment and proceeding down the list subsequent subjects were placed in alternate treatments.

The second test was administered one week after the first.

The random subtraction problems were generated as follows:

minuend = (random() modulo 900) + 100

subtrahend = random() modulo (minuend + 1)

where random () is a pseudo random number generator that generates thirty-two bit unsigned integers. This resulted in random three digit positive integer subtraction problems such that the minuend contained three digits and the correct result was positive.

Normal tuition proceeded between tests. Thus, students' approaches to the domain could be expected to alter between sessions.

A student model was constructed for each student from analysis of the first test. Performance of the system was assessed by using this model to predict the exact digit that the student would provide for each column of each answer in the second test (as per Kuzmycz and Webb, 1992).

Of the 8334 digits (forty answers each comprised of three digits provided by each of the 73 students for the second test, less questions for which no answer was provided) the system made a prediction for 80% [Random: 80%, Error Repeat: 80%; one-tailed z-test: $z = 0.003$] of all digits. Of these predictions, 92% [Random: 93%, Error Repeat: 92%; one-tailed z-test: $z = 1.0882$] were correct. However, the accuracy was considerably lower when only those 5% [Random: 5%, Error Repeat: 5%; one-tailed z-test: $z = 1.4025$] of answers for which the model predicted the student would provide an incorrect response are considered. Of these, 54% [Random: 52%, Error Repeat: 56%; one-tailed z-test: $z = -0.6002$] were correct. Of the 665 digits for which the students gave incorrect values, the system predicted a value in 62% [Random: 62%, Error Repeat: 61%; one-tailed z-test: z = 0.3238] of cases. Of these predictions 30% [Random: 34%, Error Repeat: 27%; one-tailed z-test: $z = 1.6463$] were correct. A chart of these results is presented in Figure 3. It is interesting to note that none of the differences in performance between the two groups are statistically significant at the 0.05 level.

It is difficult to evaluate the quality of the predictions presented above without further information about the subtraction performance of the students involved. At first hand it might seem extremely poor that the system's predictions were correct in only 30% of the cases where a student made an error on the second round. However, several points should be kept in mind when considering these results. First, the system was predicting exactly which of ten possible digits the student would provide for a specific column. Thus, random performance would result in 10% predictive accuracy. A model of correct subtraction skills would result in 0% accuracy (as all the answers were, by definition, not the answers provided by a correct model). Thus, the difference between 10% accuracy and the observed accuracy can only be accounted for by a non-trivial model of the students' buggy subtraction strategies.

Further insight into the quality of the model is offered when one considers the performance of the students in the Error Repeat treatment on the subtraction problems for which they provided an incorrect answer in the first test. Of the answers provided to these problems in the second test. 49% of the answers were correct, 26% were erroneous but different from the answer provided in the first test and only 25% of answers were identical to those provided in the first test. These results are charted in Figure 4.

Each of these categories might be taken as an approximate indication of the relative proportions of slips, tinkering and consistent bugs. Errors that were corrected might be largely slips. Those that were changed might be tinkering associated with unresolved impasses (Brown and VanLehn, 1980). Those that were repeated might result from consistent buggy subtraction strategies. However, these indicative assumptions can only be considered very approximate. 10% of tinkering for which a random response was provided would result in each of corrected and repeated answers. Corrected answers could also indicate bugs or impasses that have been repaired. Further, a student could make slips when applying an erroneous strategy.

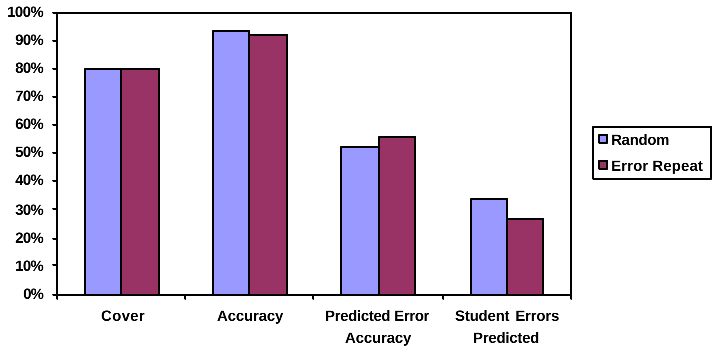
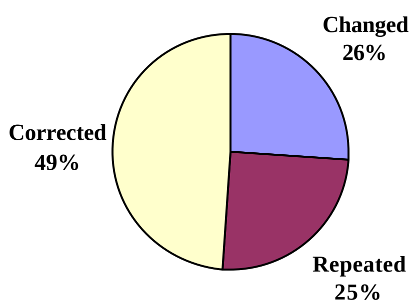


FIG. 3. SUMMARY OF SYSTEM PREDICTIONS.



•

FIG. 4. ANALYSIS OF ANSWERS PROVIDED HI THE SECOND TEST TO SUBTRACTION PROBLEMS FOR WHICH ERRORS WERE MADE IN THE FIST TEST.

These results place a new light on the 34% accuracy in predicting student errors obtained for the Random treatment. If only 25% of errors are consistent across tests and approximately 50% are corrected by the next test, this suggests a theoretical upper limit on the proportion of student errors that could be accurately predicted of approximately 33%. This would be obtained if the system made no predictions for the inconsistent errors, predicted that the student would provide correct answers for the once off errors and identified the consistent strategy for the consistent errors. The system's predictions with respect to the once off errors would be incorrect (approximately 50% of the student's answers), the predictions for the consistent errors would be correct (approximately 25% of answers) and no prediction would be made for the inconsistent errors (approximately 25% of answers). Thus, of the 75% of the student's erroneous answers for which the system made a prediction, 33% would be correct (with respect to Figure 4, *repeated* as a proportion of *repeated* + *corrected*). An accuracy of 34% is remarkably close to this approximate theoretical upper limit.

The analysis of student error consistency also throws a different light on the accuracy of the system's predictions that the student would make an error. It might at first seem surprising that the accuracy of these predictions is lower for the Error Repeat condition than for the Random condition. However, this result suggests the following tendencies. When a problem is encountered for which an error was previously observed, the system will tend to predict a repetition of that error. In contrast, when making predictions for problems that are similar but distinct from problems for which the student has previously made an error, a repetition of the error will only be predicted when there is a consistent pattern underlying the student's answers (a consistent bug). To re-express this, the highly specific rules in the model tend to closely map the student's exact observed actions. In contrast, the more general rules tend to capture regularities in the student's performance. If this is correct, the model can be expected to improve as more observations are made allowing the generation of ever more general rules.

The consistency results suggest that a modelling system that was not distinguishing the consistent from the once-off and inconsistent errors would only obtain an accuracy of 25% for its predictions that the student would make an error. The system has clearly demonstrated that it is successfully abstracting consistent errors from the others by obtaining an accuracy of 54% in its predictions that the student would make an error.

It is not clear to what extent the above results are influenced by the passage of time between the two tests. Class room tuition proceeded between tests. The student's approaches to subtraction may also have evolved under any number of other influences over this period of time.

8.1 Study 2

To isolate these influences, a further study was conducted in which 16 nine to ten year old students were given two pre-generated tests of forty problems each with an interval between tests of no more than thirty minutes. It was not possible to perform consistency checking in this study as it was not feasible within this time frame to generate new tests based on the answers from previous tests.

Like the first study, models were formed for individual students by analysis of their performance on the first test. For each student, the model constructed for that student from the first test was then used to predict the precise answers that would be provided for the problems in the second test.

Of the 1917 digits (forty answers each comprised of three digits provided by each of the 16 students, less one problem for which no answer was provided) the system made a prediction for 84% of all digits. Of these predictions, 93% were correct. The system predicted an erroneous response for 5% of answers. Of these, 83% were correct (a considerable increase over the results for the first study). Of the 293 digits for which the students gave incorrect values, the system predicted a value in 168 cases. Of these predictions, 44% (74) were correct, another large increase in accuracy. When these results are considered in the context of the observed consistency rates from the previous study, the accuracy rates are very high. As the system exceeds the 33% plausible upper limit suggested by the consistency results from Study 1, it seems likely that there is greater consistency in performance over a thirty minute interval than over a one week interval of time. Figure 5 presents a comparison of these results with those obtained for the Random treatment in Study 1.

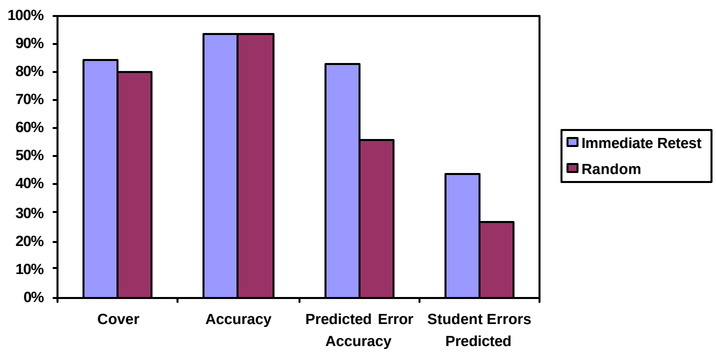


FIG. 5. COMPARISON OF SYSTEM PREDICTIONS, IMMEDIATE RETEST VS. RANDOM TREATMENTFROM ONE WEEK RETEST STUDY.

The results of the second study show that when the system is not hampered by the concept shifts that accompany a sizeable temporal gap between test applications, it is quite accurate in its ability to identify the students' consistent erroneous solution patterns. It is also able to identify at least some parts of the problem space for which the student does not have a consistent solution pattern and to refrain from making predictions in this part of the solution space.

9 GENERAL ISSUES

9.1 Evidence of the Existence of Consistent Bugs

The studies provide strong evidence that, while the majority of errors cannot be attributed to a buggy subtraction procedure, a non-trivial proportion can. If a student provided random answers in contexts in which he or she did not have a correct procedure, only 10% of errors would be repeated. A one-tailed *z*-test demonstrates that the observed repetition rate (25%) was significantly greater than 10% (z = l0.05; $p < 0.005$). The students' erroneous answers are not random.

Further, the identification by the system of consistent bugs provides the only plausible explanation of the accuracy with which the system was able to both predict that the student would provide an erroneous answer and predict the answer provided when it was erroneous. The accuracy of error predictions (54%) and the proportion of errors for which a prediction was made for which that prediction was correct (30%) were both significantly higher than the observed error repetition rate (25%) (one-tailed z-test ; $z = 12.09$, $p < 0.005$ and $z = 2.92$, $p < 0.005$, respectively).

The higher accuracy of the system in predicting student errors in the second study suggests that more consistent bugs were still present after thirty minutes than survived the week between tests in the first study. This suggests that, at least in a class room context, many buggy procedures have a short life span. Possible explanations of this effect include short-term memory of procedures developed by tinkering; self reflection leading to revision of erroneous procedures; and procedure change resulting from educational interactions.

9.2 Computational Issues

Aside from the predictive accuracy that FBM can provide, it also promises low computational overheads in comparison to process modelling. The most recent version of the FBM modelling system has been designed to maximise computational efficiency. This software was developed during 1990 and used with the Unification Tutor and Subtraction Modeller in 1991 and 1992.

The Unification Tutor uses thirty-seven context features, and eighteen action features. These have been described in detail by Webb, Cumming, Richards and Yum (1989). From thirty-seven context features it is possible to derive 2^{37} potential sets of context features. Through elimination of invalid and redundant sets of features this was reduced to just 1531 context feature sets. Each of these can be associated with any of the eighteen action features, providing the system with a total of 27,558 potential associations. As an agent model consists of a set of associations, the number of models that can be expressed is 2^{27558}.

As explained above, the computational complexity of updating the model is of the order 2^n, where n is the number of context features that describe the action for which the model is being updated. (While the elimination of redundant feature sets reduces the total number of updates, it does not reduce them sufficiently to reduce the order of magnitude of the update task).

Despite the large number of models that the system has the potential to form, efficient implementation provided more than adequate performance for interactive use. During 1993 access to the Unification Tutor was provided to third year Computer Science students on a Solbourne series 5/602 computer. After each action that the student performed the model was updated and then used to control the system's responses and subsequent actions. The average real time taken to upgrade the model was just 1.4 seconds. The average CPU time was 1.2 seconds. Such response times are clearly rapid enough to support interactive application.

9.3 Relationship to Constraint-Based Modelling

There are some similarities between FBM and Constraint-Based Modelling (Ohlsson, 1992). Constraint-Based Modelling monitors for constraints on an application domain that are violated by the agent. Like FBM, Constraint-Based Modelling restricts itself to consideration of observable aspects of the agent's problem solving and does not form a process model of the student.

However, the approaches differ in a number of significant respects. First, whereas Constraint Based Modelling detects only aspects of an agent's erroneous performance (the constraints that the agent violates) FBM also models mastery (both what the agent does correctly and incorrectly). Further, unlike Constraint Based models, Feature Based models are executable. They can be used to predict future performance.

It is possible to combine both FBM and Constraint Based modelling. If constraints are cast as action features (a feature that is present when the constraint is violated), FBM will form a model that indicates not only which constraints are violated, but also the circumstances in which they are violated. Such a model provides an executable form of Constraint Based model. Many of the action features employed in the unification tutor can be thought of in this way (Webb, 1991).

9.4 Viewpoint Independence

Viewpoints raise a serious issue for cognitive modelling systems. Wenger (1987) defines a viewpoint as an interpretive context. A viewpoint can be defined in many dimensions including *problem solving strategies, solution methods* and *conceptual* frameworks. In many educational contexts, different students will have different viewpoints of the subject matter being examined. For example, one widely documented difference between viewpoints in subtraction relates to the strategy for handling cases where the subtrahend in one column is greater than the minuend (Fawcett and Cummins, 1970). One strategy adds one to the subtrahend in the column to the left. Another subtracts one from the minuend.

If a modelling system assumes one viewpoint and the agent being modelled adopts another, it will not be possible to construct an accurate model. Further, if the content of a model based on an incorrect viewpoint is discussed with the agent (for example, if a tutoring system describes a bug that it believes a student exhibits), the outcome is likely to be undesirable (for example, the student will lose faith in the tutor).

A system that seeks to model the internal operation of the cognitive system is forced to assume a viewpoint. Those systems that address the issue of multiple viewpoints do so by supporting a set of alternative viewpoints. When constructing a model, such a system must select one of the available viewpoints for that model (see, for example, Burton and Brown. 1982; Langley, Ohlsson and Sage, 1984). Selection between multiple viewpoints adds greatly to the computational complexity of modelling task. The need to adopt a viewpoint adds an additional factor that may introduce error into a model.

In contrast, FBM does not require the adoption of a viewpoint because the model relates contexts directly to actions without attempting to reconstruct the intervening cognitive processes. For example, the Subtraction Modeller does not make any assumptions about whether a student borrows from the minuend or carries to the subtrahend.

This is not to say that viewpoints can never cause difficulties for an FBM modelling system. Different viewpoints may take account of different context features. If the designer of a modelling system fails to anticipate the features relevant to a particular viewpoint, then the system will not be able to produce accurate models for agents adopting that viewpoint. For example, although most common approaches to subtraction consider columns from right-to-left, it is also possible, and in some respects less complex, to solve subtraction problems working from left-to-right. For a student that solves subtraction problems from left-to-right, features of the columns to the left of the current column might be relevant to modelling performance. The current implementation of the Subtraction Modeller considers only features of columns to the right of the current column, and thus is unable to adequately model a student with a left-to-right solution strategy.

While the existence of multiple viewpoints can cause difficulties for an FBM system, the magnitude of the problem is much smaller than for process modelling techniques. A single set of context features may be adequate to describe competency in multiple viewpoints. An FBM system does not need to select a viewpoint in order to produce a model.

9.5 Multiple Strategies

There is considerable evidence that students, in at least some problem domains, employ many different problem solving strategies, selecting between strategies on a task-by-task basis (Ohlsson and Bee, 1991; Payne and Squibb, 1990; Siegler, 1989; VanLehn, 1982). This appears to present a major difficulty to the construction of process models, which often rely on the assumption that a single strategy can capture the student's competency.

Just as FBM is not reliant upon identifying a single viewpoint for a student, it is not reliant upon identifying a single problem solution strategy. Rather, an FBM model, as a model of competency, can as adequately model competency arising from the consistent use of a set of strategies as it can model competency arising from the consistent use of a single strategy.

9.6 Defensibility

Martin and VanLehn (1993) raise the issue of the potential need to defend the models produced by a modelling system. For example, defensibility could become very important if the models are to be used for educational assessment. Martin and VanLehn even raise the spectre of the validity of a model being the subject of legal action.

To Martin and VanLehn, the essential requirement for a model to be defensible appears to be that the process by which it is generated is deterministic and statistically sound. This rules out the use of heuristics in the construction of the model.

RBM satisfies this criterion. The construction of a model is not heuristic, is based on simple statistics and thus is readily defensible.

9.7 Interpretability

Another dimension along which modelling systems may be evaluated is that of ease of interpretation. Such a criterion relates to the ease with which users of the system (in an educational context, both teachers and students) can comprehend the models that the system constructs.

FBM models are relatively straight forward to interpret. They can be considered as a set of independent production rules (each association is equivalent to a single production rule). Each association can be considered in isolation. The meaning of each association is straight forward. An association means that, in general, when the nominated context features are present the agent acts in the manner indicated by the action feature.

Experience with the Unification Tutor has demonstrated that it is straight forward to generate natural language descriptions of associations and that students have little difficulty in interpreting those descriptions.

10 LIMITATIONS AND FURTHER RESEARCH

FBM ignores internal cognitive states. In the real world, identical contexts may produce different actions if the cognitive system is in a different state. For example, when solving subtraction problems, if an agent is working from right to left, the solution of one column will depend upon internal cognitive states (such as remembering carry) resulting from the analysis of previous columns. FBEM is forced to assume that the only relevant cognitive states pertain to the agent's understanding of the subject domain and to the features of the context with which the system is presented. It is important that FBM be applied at a level of granularity at which this assumption is realistic. Thus, for example, it is essential that a modelling system for subtraction that examines individual columns should consider other columns of the current problem as part of the context for each action.

The use of categorical attribute-value machine learning techniques is restrictive. For example, it is cumbersome to describe with categorical attributes whether borrow from the previous column is appropriate for the current column (unless a new high level attribute is defined for this purpose). There is scope for the application of more powerful machine learning paradigms, such as the induction of logic programs (Muggleton and Feng, 1990; Quinlan. 1991), to the induction of models that describe the relationships between the inputs and outputs to the cognitive system.

Within the attribute-value machine learning paradigm, Kuzmycz (in preparation) is seeking to extend the power of FBM by supporting constructive induction (Bloedorn and Michalski, 1991) whereby new features are developed and employed as needed.

FBM employs a number of user defined parameters, namely, *min_evidence*, *min_accuracy* and the data discounting rate. Values of 3, 0.8 and 0.9, respectively, have been employed for these parameters throughout most of the evolution of FBM. These values were derived through intuition and limited informal experimentation. It would be valuable to perform formal evaluation of the effects of different values for these parameters in various contexts. With a view to establishing guidelines for selecting appropriate values for specific applications it would be even better if a sound theoretical basis could be developed for selecting appropriate values for these parameters.

There is a need for further consideration and evaluation of the potential educational applications for FBM. While Kuzmycz and Webb (1992) have demonstrated that FBM is able to construct models that capture aspects of an agent's approach to a domain and Webb, Cumming, Richards and Yum (1990) have provided informal evidence of educational benefit arising from the use of FBM, it is yet to be formally demonstrated that such a model can be harnessed for educational benefit. The experience of Webb, Cumming, Richards and Yum (1990) illustrates the difficulties inherent in evaluating the educational benefit of a modelling system. However, this in no way diminishes the importance of performing such experimentation.

11 SUMMARY

FBM employs attribute-value machine learning principles to produce a model of the cognitive system that captures the relationships between the inputs and outputs to that system. Such a model may be considered to be a model of an agent's competency but not as a model of the mechanisms that produce that competency. This approach to cognitive modelling prevents the modelling system from having to reason about chains of hypothesised and unobservable events that are internal to the cognitive system. This greatly reduces the complexity of the modelling task. It minimises the amount of empirical research into student errors that is required before a modelling system can be produced. It also reduces the degree to which the problems of viewpoints and the use of multiple strategies affect the validity of the models produced.

FBM has been implemented in a wide variety of domains encompassing simple classification (word classes), manual skills (keyboard scale tuition) and problem solving (unification and subtraction).

Classroom use of the Unification Tutor has demonstrated that the methodology is capable of producing complex models within a time frame that supports interactive application. Formal evaluation of the Subtraction Modeller has demonstrated that the approach is able to develop models capable of predicting in excess of 90% of an agent's actions in a real world classroom environment despite the presence of noise and of concept shifts resulting from ongoing tuition. Further, the system demonstrated the ability to identify bugs-consistent erroneous approaches to subtraction. In short, FBM is an approach to modelling an agents' competencies that has demonstrated the capacity to produce coherent, consistent models with high predictive accuracy.

Acknowledgements

This research has been supported by the Australian Research Council. We are grateful for numerous helpful suggestions and comments from Phil Smith, Jon Oliver and anonymous reviewers. We are indebted to Tom Richards for providing the initial stimulus for this research and for invaluable support and assistance throughout.

References

Amato, N. H. and C.P. Tsang: 1990, 'Student modelling in a keyboard scale tutoring system.' In C.J. Barter and M. J. Brooks (Eds.,) *Proceedings of the Second Australian Joint Conference on Artificial Intelligence*, Springer-Verlag. Berlin. pp. 225-239.

Anderson, J. R., C. F. Boyle, A. T. Corbett, and M. W. Lewis: 1990, 'Cognitive modelling and intelligent tutoring.' *Artificial Intelligence* **42**, 7-49.

Anderson, J. R., C. F. Boyle and B. I. Reiser: 1985, 'Intelligent tutoring systems.' *Science* **228**, 456-462.

Attisha, M. and M. Yazdani: 1984, A micro-computer based tutor for teaching arithmetic skills.' *Instructional Science* **12**, 333—342.

Bloedorn, E. and R. S. Michalski: 1991, 'Data-driven constructive induction in AQ17-PRE: A method and experiments.' In *Proceedings of the 1991 IEEE International Conference on Tools for Artificial Intelligence*, San Jose, CA, pp. 30-37.

Brown, J. S. and R.R., Burton: 1978, ‘Diagnostic models for procedural bugs in basic mathematical skills,’ *Cognitive Science* **2**, 155-192.

Brown, J.S. and K. VanLehn: 1980, ‘Repair theory: A generative theory of bugs in procedural skills.’ *Cognitive Science* **4**, 379-426.

Burton, R.R. and J S. Brown: 1982, ‘An investigation of computer coaching for informal learning activities.’ In Sleeman, D. H. and Brown, J. S. (Eds.) *Intelligent Tutoring Systems*. Academic Press, London, pp. 79-98,

Clancey, W. J.: 1987, *Knowledge-Based Tutoring: The GUIDON Program*. MIT Press, Cambridge, Mass.

Corbett, A. T. and J.R. Anderson: 1992, ‘Student modelling and mastery learning in a computer-based programming tutor’ In Frasson, C., Gauthier, G. and McCalla, G. I. (Eds.) *Intelligent Tutoring Systems*, Springer-Verlag. Berlin, pp. 413-420.

Corbett, A. T, J.R. Anderson, and A. T. O’Brian: 1993, ‘The predictive validity of student modelling in the ACT programming tutor.’ In *Proceedings of AI-Ed 93*, Edinburgh, pp. 457-464.

Fawcett, H. P. and K. B. Cummins: 1970. *The Teaching of Mathematics from Counting to Calculus*: Merrill, Columbus, Ohio.

Goldstein, I P. 1979, ‘The genetic-graph: A representation for the evolution of procedural knowledge.’ *International J. Man-Machine Studies* **11**, 51-77.

Ikeda, M. Y. Kono, and R. Mizoguchi: 1993, ‘Monmonotonic model inference: A formalization of student modeling,’ In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. Chambery, pp. 467-473.

Kuzmycz, M. (in preparation), ‘Praxis Modelling: An Evaluation of and Extension to Feature Based Modelling.’

Kuzmycz, M. and G. I. Webb: 1991, ‘Modelling elementary subtraction: Intelligent warfare against bugs.’ In *Proceedings of the Fourth Australian Society for Computers in Learning in Tertiary Education Conference*, Launceston, pp. 367-376.

Kuzmycz, M. and G. I. Webb: 1992, ‘Evaluation of Feature Based Modelling in subtraction.’ In Frasson, C., Gauthier, G. and McCalla, G. I. (Eds.) *Intelligent Tutoring Systems*. Springer-Verlag, Berlin, pp. 413-420

Langley, P., S. Ohlsson, and S. Sage: 1984, *A Machine Learning Approach to Student Modeling*. The Robotics Institute, Carnegie-Mellon University, Technical Report CMU-RI-TR-84-7.

Langley, P., J. Wogulis and S. Ohlsson: 1990, ‘Rules and principles in cognitive diagnosis.’ In Frederiksen, N. Glaser, R., Lesgold. A. and Shafto, M. G. (Eds.) *Diagnostic Monitoring of Skill and Knowledge Acquisition*. Lawrence Erlbaum, Hillsdale, NJ, pp. 217-250.

London, R. V. 1992, ‘Student modeling to support multiple instructional approaches.’ *User Modeling and User Adapted Interaction* **2**, 117-154.

Martin, J. D. and K. VanLehn 1993, ‘OLA E: Progress toward a multi-activity, Bayesian student modeller.’ In *Proceedings of AI-Ed 93*, Edinburgh, pp. 410-417.

McArthur, D., C. Stasz and M. Zmu idzinas. 1990, ‘Tutoring techniques in algebra,’ *Cognition and Instruction*. **7**, 197-244.

Mitchell, T M.: 1977, ‘Version spaces: A candidate elimination approach to rule learning,’ *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pp. 305-310.

Muggleton, S. and S. Feng: 1990, ‘Efficient induction of logic programs.’ In *Proceedings of the First Conference on Algorithmic Learning Theory*, Tokyo.

Ohlsson, S.: 1992, ‘Constraint-based student modelling.’ *Artificial Intelligence in Education* **3**. 429-447

Ohlsson, S. and N. Bee: 1991, ‘Strategy variability. A challenge to models of procedural learning.’ *Proceedings of the International Conference of the Learning Sciences*, Charlottesville, VA., pp. 351-356,

Payne, S.J. and H. R. Squibb: 1990. ‘Algebra mal-rules and cognitive accounts of errors.’ *Cognitive Science* **14**, 445-481.

Putnam, R. T. 1987, ‘Structuring and adjusting content for students: A study of live and simulated tutoring of addition.’ *American Educational Research Journal* **24**, 13-48.

Quinlan, J. R. 1991, ‘Determinate Literals in Inductive Logic Programming,’ *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*. Morgan Kauffman, Los Altos, pp. 746-750.

Siegler, R. 1987, ‘Hazards of mental chronometry: An example from children’s subtraction.’ *Journal of Educational Psychology* **81**, 497-506.

Sleeman, D. H. 1982, ‘Assessing aspects of competence in basic algebra’. In Sleeman, D. H. and Brown, J. S. (Eds.) *Intelligent Tutoring Systems*. Academic Press, London. pp. 185-199.

Sleeman. D. H. 1987, ‘PIXIE: A shell for developing intelligent tutoring systems.’ In Lawler, R. W. and Yazdani, M. (Eds.) *Artificial Intelligence and Education*. Ablex, Norwood, NJ, pp. 139-263.

Sleeman, D. H., R. D. Ward, E. Kelly, R. Martinak and J. Moore. 1991. ‘An overview of recent studies with Pixie.’ In P. Goodyear (Ed) *Teaching Knowledge and Intelligent Tutoring*, Ablex, pp 173-185.

Stevens. A. L., A. Collins and S.E. Goldin. 1982, ‘Misconceptions in students’ understanding.’ In D. H. Sleeman and J. S. Brown (Eds.) *Intelligent Tutoring Systems*. Academic Press. London, pp. l3-24

VanLehn, K. 1982, ‘Bugs are not enough: Empirical studies of bugs, impasses, and repairs in procedural skills,’ *Journal of Mathematical Behavior* **3**, 3-72.

VanLehn, K. 1986. ‘Arithmetic procedures are induced from examples,’ In Hiebert, J. (Ed.) *Conceptual and Procedural Knowledge*: *The Case of Mathematics*, Erlbaum, Hillsdale, NJ. pp 133-179.

Webb, G. I. 1988, A knowledge based approach to computer-aided learning. *International J. Man-Machine Studies* **29**. 257-285.

Webb, G. I. 1991, lnside the Unification tutor: The architecture of an intelligent educational system,’ In *Proceedings of the Fourth Australian Society for Computers in Learning in Tertiary Education Conference*. Launceston, pp. 677-684

Webb, G. I., G. Cumming, T, Richards and K-K. Yum. 1989, ‘The Unification Tutor: An intelligent educational system in the classroom.’ *Proceedings of the Third Australian Society for Computers in Learning in Tertiary Education Conference*, Bond University, pp. 408-420.

Webb, G. I., G. Cumming, T, Richards and K-K. Yum 1990, ‘Educational evaluation of feature based modelling in a problem solving domain.’ In R. Lewis and S. Otsuki (Eds.) *Advanced Research on Computers in Education*, Amsterdam, Elsevier, pp 101-108.

Wenger, B. 1987, *Artificial Intelligence and Tutoring Systems*: Morgan Kaufmann, Los Altos.

Young, R. M. and T, O’Shea 1981, ‘Errors in children’s subtraction.’ *Cognitive Science* **5**, 153-177.

Authors’ Vitae

Dr. G Webb

Deakin University, School of Computing and Mathematics, Geelong, Vic., Australia, 3217.
Dr. Geoff Webb is a Reader in Computing at Deakin University. He received his B.A. and Ph.D. degrees in Computer Science from La Trobe University. Dr. Webb is active in several areas of artificial intelligence research, including machine learning, knowledge acquisition and artificial intelligence in education. He has authored more than fifty technical papers spanning the above areas of research. This paper summarises the better part of a decade’s research into user modelling.

M. Kuzmycz

Deakin University, School of Computing and Mathematics, Geelong, Vic., Australia, 3217.
Mark Kuzmycz is a Ph.D. candidate in Computer Science at Deakin University. He received his B.Sc. in Computer Science from Deakin University in 1990. His primary interests lie in the areas of student modelling and artificial intelligence in education. He has worked extensively on the Feature Based Modelling research reported herein. His thesis work extends Feature Based Modelling through dynamic manipulation of the modelling system’s vocabulary of terms.

Appendix

FEATURES EMPLOYED BY THE SUBTRACTION MODELLER

The following is a complete list of the features used by the Subtraction Modeller in the studies presented above. An example illustration is provided for each feature. In this illustration, the column to which the feature refers is outlined by a box. Digits for which the value is not relevant are represented by a question mark. The context features are grouped by feature choice. All action features belong to binary feature choices. Only one value is presented for each of these feature choices. Thus, for each listed action feature there is a corresponding negative action feature.

CONTEXT PEATURES

Minuend>Subtrahend	? **2** ? - ? **1** ?
Minuend<Subtrahend	? **1** ? - ? **2** ?
Minuend=Subtrahend	? **2** ? - ? **2** ?
Minuend>Subtrahend in the column to the right	? ? **2** - ? ? **1**
Minuend<Subtrahend in the column to the right	? ? **1** - ? ? **2**
Minuend=Subtrahend in the column to the right	? ? **2** - ? ? **2**
Minuend>Subtrahend two column to the right	? ? **2** - ? ? **1**
Minuend<Subtrahend two column to the right	? ? **1** - ? ? **2**
Minuend=Subtrahend two column to the right	? ? **2** - ? ? **2**
Minuend is zero	? **0** ? - ? ? ?
Minuend is not zero	? **1** ? - ? ? ?
Minuend is zero in the column to the left	**0** ? ? - ? ? ?
Minuend is not zero in the column to the left	**2** ? ? - ? ? ?
Minuend is zero in the column to the right	? ? **0** - ? ? ?
Minuend is not zero in the column to the right	? ? **2** - ? ? ?

Minuend is one in the column to the left	1 ? ? - ? ? ?
Minuend is not one in the column to the left	2 ? ? - ? ? ?
Subtrahend is zero	? ? ? - ? 0 ?
Subtrahend is not zero	? ? ? - ? 2 ?
Subtrahend is nine	? ? ? - ? 9 ?
Subtrahend is not nine	? ? ? - ? 2 ?
Subtrahend is nine in the column to the right	? ? ? - ? ? 9
Subtrahend is not nine in the column to the right	? ? ? - ? ? 2
Subtrahend is blank	? ? ? - ? ?
Subtrahend is not blank	? ? ? - ? 2 ?
This column is right-most	? ? ? - ? ? ?
This column is left-most	? ? ? - ? ? ?
This column is neither left nor right-most	? ? ? - ? ? ?

ACTION FEATURES

Result = Minuend - Subtrahend	? 2 ? - ? 1 ? = ? 1 ?
Result = Minuend - Subtrahend - 1	? 2 ? - ? 1 ? = ? 0 ?
Result = Minuend - Subtrahend + 10	? 2 ? - ? 3 ? = ? 9 ?
Result = Minuend - Subtrahend + 9	? 2 ? - ? 3 ? = ? 8 ?
Result = Minuend	? 2 ? - ? ? ? = ? 2 ?

Result = Subtrahend	? ? ? - ? **1** ? = ? **1** ?
Result = Zero	? ? ? - ? ? ? = ? **0** ?
Result = Minuend - Subtrahend - 2	? **4** ? - ? **1** ? = ? **1** ?
Result = Minuend - Subtrahend + 8	? **2** ? - ? **1** ? = ? **9** ?
Result = Subtrahend - Minuend	? **1** ? - ? **2** ? = ? **1** ?
Result is correct	? **2** ? - ? **1** ? = ? **1** ?
Result is incorrect	? **2** ? - ? **1** ? = ? **2** ?