



# Artificial intelligence for user modeling: Capabilities and limitations

**Geoff Webb**

# Student modeling: The dream

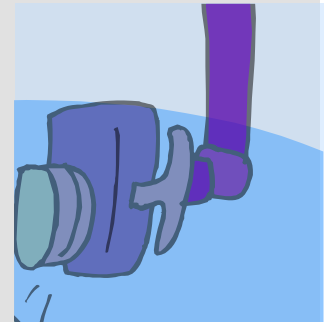
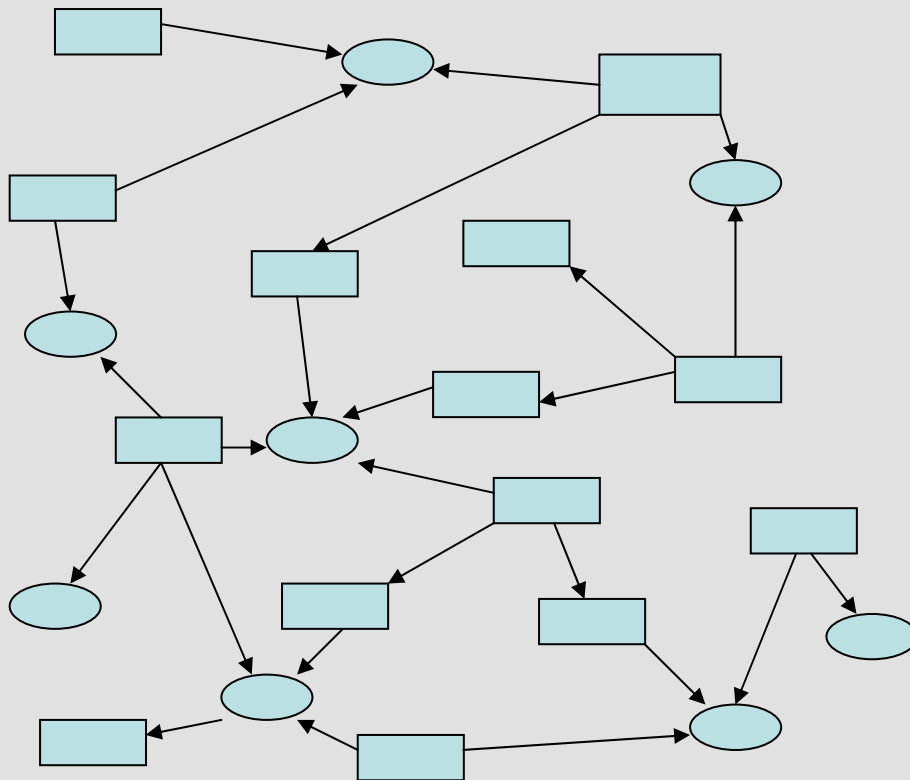
- **Artificial Intelligence systems monitor each student's actions and develop detailed models of their knowledge and skills.**
  - Individualised tailored education for all

# Some notable successes

- **Some very sophisticated and impressive systems based on these core ideas have been developed and deployed**
  - CMU Lisp and Geometry tutors
  - Pittsburgh Newtonian Physics Tutor
- **But after a quarter century of research, why aren't we all using them?**

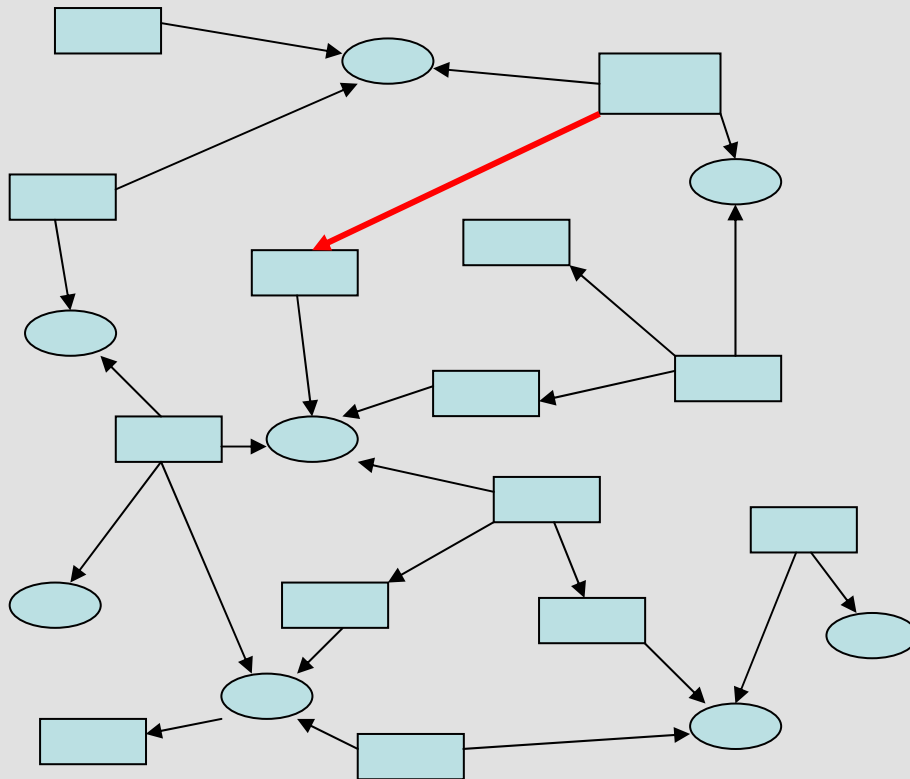
# X-Ray of the mind!

Cognitive Model



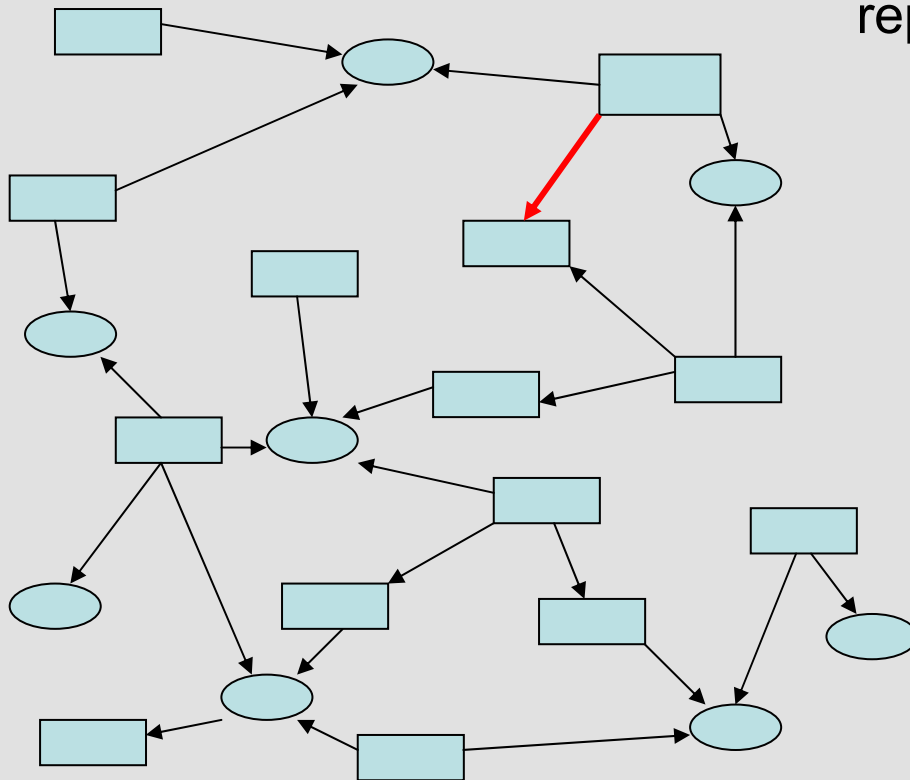
# X-Ray of the mind!

Identify problems



# X-Ray of the mind!

Engage in education activities to repair them



# Burton and Brown (1978) “BUGGY”

- **Modeling subtraction skills**
- **System contains explicit model of correct procedure**
  - step-by-step rules
- **and a library of “bugs”**
  - mal-rules
- **System analyses a student’s solutions to subtraction problems and infers that the student has the bugs that best describe their answers**

# Examples

- **Stops Borrow at Zero**

102	9007	4015	702	2006	10012	8001
<u>39</u>	<u>6880</u>	<u>607</u>	<u>108</u>	<u>42</u>	<u>214</u>	<u>43</u>
73	2227	3408	604	2064	10898	8068

- **Borrow Across Zero**

102	9007	4015	702	2006	10012	8001
<u>39</u>	<u>6880</u>	<u>607</u>	<u>108</u>	<u>42</u>	<u>214</u>	<u>43</u>
3	1227	3408	504	1064	98	6068

(Examples from VanLehn, 1981)

# Errors result from bugs or slips

- **Early work assumed that errors fall into two classes**

- Bugs are systematic faults in the student's skill
- Slips are failures to correctly apply a skill

# Impasses

- **How do bugs arise?**
- **Impasse theory assumes that when a student encounters a situation with which their current procedure cannot cope (*an impasse*) they generate a *repair* on the fly.**
- **Repairs may or may not become established.**

# Bugs are not enough

- **VanLehn (1981) reported on a study of 925 third-grade students who were given a series of subtraction tests.**
- **In his assessment the students can be divided into the following categories in the following proportions**
- **No errors on either test 4%**
- **Stable correct procedure, some slips 50%**
- **Stable bugs, some slips 12%**
- **Stable impasses (plus slips and stable bugs) 12%**
- **Unstable bugs and/or impasses (with slips and stable bugs) 12%**
- **Errors on one or both tests that cannot be analysed 10%**

# Why are so many transcripts inexplicable?

- **Perhaps because the notion of a single correct procedure is ill-founded!**
- **There are many correct procedures for many non-trivial tasks**

# Subtraction Procedure 1: Borrowing

$$\begin{array}{r} \overset{1}{\cancel{2}}0 \\ -15 \\ \hline \end{array}$$

# Subtraction Procedure 2: Adding

$$\begin{array}{r} 20 \\ -15 \\ \hline \end{array}$$

# Subtraction Procedure 3: Left to Right

$$\begin{array}{r} 20 \\ -15 \\ \hline 0 \\ \cancel{1}5 \end{array}$$

# Restrictiveness

- **Cognitive modeling requires that a system either**
  - impose a single approach to a domain and force students to work within it, or
  - encode multiple approaches and be able to identify which approach a student is (incorrectly) using at any time

# Granularity

- **How do you ensure that your models capture the level of detail that is relevant to analysing a student's mastery?**

# Comprehensiveness

- **The domain model for each approach must include**
  - explicit descriptions of every skill or procedure at the relevant level(s) of granularity
  - explicit descriptions of every possible bug

# A catch-22

## McArthur, Lewis, and Bishay (1994)

- ITS require *relatively complete mastery of the subject area they are to tutor, including an understanding of likely student misconceptions.*
- The automation of a competency ipso facto reduces the value of teaching it!

# Interactive Learning Environments

- **Rather than teaching the low level mechanical competencies to which cognitive modeling is suited, ILEs provide rich exploratory environments.**



# Summary of issues for cognitive modeling

- **Multiple valid skill and knowledge sets**
- **Restrictiveness**
- **Granularity**
- **Comprehensiveness**
- **Better suited to tutoring mechanical skills than higher level and exploratory activities**

# An alternative to cognitive modeling

- **Feature Based Modeling: Develop model of competencies at the level of observables rather than low level internal cognitions**
- **Models characterise**
  - the contexts in which a student acts, and
  - the actions that are taken
- **Example**

Minuend to immediate right is zero AND  
Minuend two cols to right < subtrahend  
=> result in current column = correct - 1  
vs “Borrow-over-zero”

# Advantages

- **No inference of unobservable internal cognitions**
- **No assumptions made about approach student has adopted**
- **Can predict student actions in novel contexts**
- **Can characterise contexts in which student's actions will have specific features (eg incorrect)**
- **Need not be prescriptive**
- **Inspectable**
  - as models relate only to observable features of contexts and actions, they can be readily described to a student or teacher
- **Easier to create than bug libraries**
- **Extensible**
  - can add features as time progresses

# Limitations

- **Descriptions can be less succinct than a cognitive model**
- **It can be less straightforward to develop educational interventions**

# Using FBM for tutoring

- **Use model to select tasks at boundary of student's competency**
- **Generate immediate feedback that specifies inappropriate patterns that have been observed in the student's actions**
  - help the student to identify and repair the deficiencies in their knowledge and skills

# The Unification Tutor

Enter a most general unifier for the following terms or type none, ? or exit.

`second(value(z), u)`

`second(E, E)`

`=>{E=value(z) ,E=u}`

It appears to me that when two terms have a variable appearing more than once opposite terms that are different you create two substitution pairs with the same variable on the left of each.

You should never create two substitution pairs with the same variable on the left of each.

Perhaps you should reconsider how you tackle such problems.

My answer is none.

Press space to continue.

# Experience with the Unification Tutor

- **Formal evaluation of the unification tutor showed**
  - Students were extremely positive about it
    - > 'using the Tutor improved my understanding of unification' mean = 4.88 (scale 1-6)
    - > 'the Tutor was easy to use' mean = 5.00 (scale 1-6)
  - Students rated comments generated from the model (mean 3.88 on scale of 1-6) higher than those generated without the model (mean 3.45) although this difference was not significant ( $t(df=8)=.44$ ,  $P=0.27$ )

# Subtraction modeller

- **Seek to evaluate accuracy of models that can be formed**
- **Give 73 grade-three students 40 x 3 digit subtraction tasks over successive weeks**
- **Seek to predict performance**
  - in context of changing competency!

# Evaluation on second test

- **8334 digits predicted (73 students, 40 questions each, 3 digit answers, less questions with no answer)**
- **prediction made for 80% of all digits.**
- **92% of predictions were correct.**
- **For 5% of answers the model predicted the student would provide an incorrect response.**
  - 54% of these predictions were accurate.
- **Of the 665 (8%) digits for which the students gave incorrect values, the system predicted a value in 62%. (chisq,  $p < 0.001$ )**
- **Of these, 30% were accurate predictions.**
  - random prediction 10%, model of correct subtraction 0%, so  $\geq 20\%$  due to non-trivial model of students' buggy subtraction strategies.
  - for half of students, questions with errors from first test repeated on second. Of the answers provided to these in the second test,
    - > 49% were correct,
    - > 26% were erroneous but different from the first test
    - > only 25% of answers identical to those provided in the first test.

# Conclusions

- **Cognitive modeling is fraught with difficulties**
- **Feature Based Modeling produces student model at level of observable features of actions and contexts**
- **FBM can be used to**
  - produce useful models from few examples
  - make predictions,
  - provide feedback, and
  - guide selection of activities
- **FBM is applicable to interactive learning environments as well as structured tutoring**