# Evaluation of a meta-tutor for constructing models of dynamic systems

Lishan Zhang, Winslow Burleson, Maria Elena Chavez-Echeagaray, Sylvie Girard, Javier Gonzalez-Sanchez, Yoalli Hidalgo-Pontet, Kurt VanLehn

Arizona State University, Computing, Informatics, and Decision Systems Engineering, Tempe, AZ, 85281, U.S.A.

{lishan.zhang, winslow.burleson, mchaveze, sylvie.girard, javiergs, yhidalgo, kurt.vanlehn}@asu.edu

**Abstract.** While modeling dynamic systems in an efficient manner is an important skill to acquire for a scientist, it is a difficult skill to acquire. A simple step-based tutoring system, called AMT, was designed to help students learn how to construct models of dynamic systems using deep modeling practices. In order to increase the frequency of deep modeling and reduce the amount of guessing/gaming, a meta-tutor coaching students to follow a deep modeling strategy was added to the original modeling tool. This paper presents the results of two experiments investigating the effectiveness of the meta-tutor when compared to the original software. The results indicate that students who studied with the meta-tutor did indeed engage more in deep modeling practices.

**Keywords:** meta-tutor , intelligent tutoring systems, empirical evaluation

## 1 Introduction

Modeling is both an important cognitive skill [1] and a potentially powerful means of learning many topics [5]. The AMT system teaches students how to construct system dynamics models. Such models are widely used in professions, often taught in universities and sometimes taught in high schools.

### 1.1 The modeling language, development tool and tutoring system

In our modeling language, a model is a directed graph with one type of link. Each node represents both a variable and the computation that determines the variable's value. Links represent inputs to the calculations. As in illustration, Figure 1 shows a model for the following system:

> The initial population of bacteria is 100. The number of bacteria born each hour is 10% of the population. Thus, as the population increases, the number of births increases, too. Model the system and graph the population over 20 hours.

Clicking on a node opens an editor with these tabs (and 2 others not described here):

- *Description*: The student enters a description of the quantity represented by the node.
- *Inputs*: The student selects inputs to the calculation of the node's value.
- *Calculation*: The student enters a formula for computing the node's value in terms of the inputs.

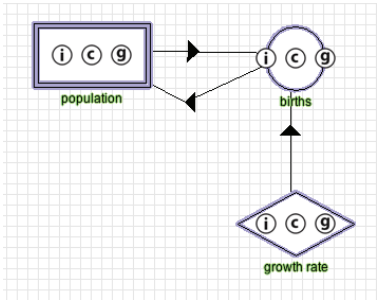


**Fig. 1.** A simple model.

There are three types of nodes in models:

- A *fixed value* node represents a constant value that is directly specified in the problem. A fixed value node has a diamond shape, never contains incoming links, and its calculation is just a single number. For instance, "growth rate" has 0.1 as the calculation of its value.
- An *accumulator* node accumulates the values of its inputs. That is, its current value is the sum of its previous value plus or minus its inputs. An accumulator node has a rectangular shape and always has at least one incoming link. For instance, the calculation tab of "population" states that its initial value is 100 and its next value is its current value + births.
- A *function* node's value is an algebraic function of its inputs. A function node has a circular shape and at least one incoming link. For instance, "births" has as its calculation "population * growth rate."

The students' task is to develop a model that represents a system described by a short text. They can create, edit and delete nodes using the node editor. When all the nodes have calculations, students can click the Run Model button, which performs calculations and draws graphs of each nodes' values over time. The system described so far is just a model development tool.

AMT has a simple tutoring capability. Each tab of the node editor has a *Check* button which turns its fields red if they are incorrect and green if they are correct. Each tab also has a *Give up* button that fills out the tab correctly. Thus, the system described so far is just a simple step-based tutoring system with minimal feedback on demand and only one kind of hint: a bottom-out hint.

### 1.2 The meta-tutor

Unfortunately, it is a rare for students to think semantically in terms of what the nodes, inputs and calculations mean actually mean. Students prefer to think of model elements syntactically, like puzzle pieces that need to be fit together. This shows up in a variety of ways, including rapid guessing, nonsensical constructions and the use of syntactic rather than semantic language to refer to model elements. The literature on model construction (reviewed in [5]) sometimes refers to these two extremes as Deep vs. Shallow modeling. The objective of the AMT system is to increase the relative frequency of Deep modeling.

A variety of methods for increasing the frequency of Deep modeling have been tried [5]. For instance, nodes can bear pictures of the quantities they represent, or students can be required to type explanations for their calculations. One of the most promising methods is *procedural scaffolding*, wherein students are temporarily required to follow a procedure; the requirement is removed as they become competent. This technique was used by Pyrenees [2], where it caused large effect sizes.

We adapted Pyrenees' procedure to our modeling language and called it the Target Node Strategy. The strategy requires students to focus on one node, called the target node, and completely define it before working on any other node. This decomposes the whole modeling problem into a series of atomic modeling problems, one per node. The atomic modeling problem is this: Given a quantity, find a simple calculation that will compute its values in terms of other quantities without worrying about how those other quantities values will be calculated. This is a much smaller problem than the overall challenge of seeing how the overall model can be constructed.

As an illustration, let us continue the bacteria population example and suppose that the target node is "number of bacteria born per hour." The ideal student might think:

> "It says births are 10% of the population, so if I knew population, then I could figure out the number of births. In fact, I could define a node to hold the 10%, and then the calculation would multiply it and population. But do I need initial population or current population? Oh. The number of bacteria born is increasing, so I must need current population, because it is also increasing."

This is one form of deep modeling. By requiring students to finish one node before working on another, the Target Variable Strategy encourages students to examine the system description closely because it is the only resource that provides relevant information. When they are allowed to work on any tab on any node, then they jump around trying to find a tab that can be easily filled in. This is a common form of shallow modeling, and the Target Node Strategy discourages it.

In addition to requiring the students to follow the Target Node Strategy, the meta-tutor nags students to avoid guessing and abuse of the Give Up button, just as the Help-Tutor [3] did. Because neither the strategy nor the advice on help seeking are specific to the domain (e.g., population dynamics), we consider them to be meta-cognitive instruction.

## 2 Evaluation

### 2.1 Experiment Design

The experiment was designed as a between-subject single treatment experiment with a control condition, where the meta-tutor was off, and an experiment condition, where the meta-tutor was on. The difference between the conditions occurred only during a training phase where students learned how to solve model construction problems. In order to assess how much students learned, a transfer phase followed the training phase. During the transfer phase, all students solved model construction problems with almost no help: the meta-tutor, the Check button and the Give-up button were all turned off, except in the Description tab where the Check button remained enabled to

facilitate grounding. Because system dynamics is rarely taught in high school, no pre-test was included in the procedure. We conducted two experiments with 44 students participating in the first experiment and 34 students in the second experiment.

### 2.2 Hypotheses and Measures

**Hypothesis 1** is that the meta-tutored students will use deep modeling more frequently than the control students during the *transfer* phase. We used the three measures below to assess it.

- The *number of the Run Model button presses* per problem.
- The *number of extra nodes* created, where extra nodes are defined as the nodes that can be legally created for the problem but are not required for solving the problem.
- The *number of problems completed* during the 30 minute transfer period.

**Hypothesis 2** is that meta-tutored students will use deep modeling more frequently than the control group students during the *training* phase. The three dependent measures used to evaluate this hypothesis are described below:

- *Help button usage:* was calculated as $(n_{wc}+3n_{gu})/n_{rn}$, where $n_{wc}$ is the number of Check button presses that yielded red, $n_{gu}$ is the number of Give-up button presses, and $n_{rn}$ is the number of nodes required by the problem.
- *The percentage of times the first Check was correct*.
- *Training efficiency:* was calculated as $3n_{cn} - n_{gu}$ where $n_{cn}$ is the number of nodes the student completed correctly ($3n_{cn}$ is the number of tabs), and $n_{gu}$ is the number of Give-up buttons presses.

**Hypothesis 3** is that the experimental group students, who were required to follow the Target Node Strategy during training, would seldom use it during the transfer phase. To evaluate this hypothesis, we calculated the proportion of student steps consistent with the target node strategy.

### 2.3 Results

Table 1 summarizes the results of experiment 1 and experiment 2.

## 3 Conclusion and future work

Although we achieved some success in encouraging students to engage in deep modeling, there is much room for improvement. If the meta-tutor had been a complete success at teaching deep modeling, we would expect to see students supported by the meta-tutor working faster than the control students. The stage is now set for the last phase of our project, where we add an affective agent to the system [4], in order to encourage engagement and more frequent deep modeling.

| *Measure (predicted dir.)* | *Experiment 1 (N=44)* | *Experiment 2 (N=33)* |
|---|---|---|
| *Transfer phase (Hypothesis 1)* | | |
| Run model button usage (E<C) | E<C (p=0.31, d=0.32) | E≈C (p=0.98, d=-0.0093) |
| Extra nodes (E<C) | **E<C (p=0.02, d=0.80)** | E<C (p=0.47, d=0.26) |
| Probs completed (E>C) | E≈C (p=0.65, d=0.04) | E<C (p=0.09, d=−0.57) |
| *Training phase (Hypothesis 2)* | | |
| Help button usage (E<C) | **E<C (p=0.04, d=0.68)** | **E<C (p=0.02, d=0.89)** |
| Correct on 1st Check (E>C) | Missing data | **E>C (p=0.015, d=0.98)** |
| Efficiency (E>C) | E<C (p=0.05, d=–0.70) | E>C (p=0.59, d=0.19) |
| *Transfer phase use of Target Node Strategy (Hypothesis 3)* | | |
| Usage (E=C) | Missing data | E≈C (p=0.59, d=−0.19). |

**Table 1. Results of Experiment 1 and 2:** E stands for the meta-tutor group, and C stands for the control group. Reliable results are bold.

## Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0910221.

## References:

1. CCSSO.: The Common Core State Standards for Mathematics, Downloaded from www.corestandards.org on October 31 (2011)
2. Chi, Min, & VanLehn, K.: Meta-cognitive strategy instruction in intelligent tutoring systems: How, when and why. Journal of Educational Technology and Society, 13(1). 25-39 (2010)
3. Roll, I., Aleven, V., McLaren, Bruce, Ryu, Eunjeong, Baker, R.S.J.d., & Koedinger, K. R.: The Help Tutor: Does metacognitive feedback improve student's help-seeking actions, skills and learning. In M. Ikeda, K. Ashley & T.-W. Chan (Eds.), Intelligent Tutoring Systems: 8th International Conference, pp. 360-369. Berlin: Springer (2006)
4. Girard, S., Chavez-Echeagaray, M. E., Gonzalez-Sanchez, J., Hidalgo-Pontet, Y., Zhange, L., Burleson, W. & VanLehn, K.: Defining the behavior of an affective learning companion in the Affective Meta-Tutor project. In Proceedings of AI in Education (2013)
5. Treagust, David F., Chittleborough, Gail, & Mamiala, Thapelo.: Students' understanding of the role of scientific models in learning science. International Journal of Science Education, 24(4), 357-368 (2002)
6. VanLehn, K. (in press). Model construction as a learning activity: A design space and review. *Interactive Learning Environments*.