DESIGN OF COMPUTATIONAL SUPPORTS FOR STUDENTS IN VISUAL MODELLING TASKS

LEO C. UREEL II[1] & KAREN E. CARNEY[2]

[1]Computer Science Department, Northwestern University.
[2]School of Education and Social Policy, Northwestern University.

ureel@northwestern.edu

Abstract. A goal of middle school physical science classrooms is teaching students to understand basic physical processes, such as diffusion or heat exchange, and to recognize the situations in which such processes can be used to describe, explain, or predict observed behaviour. One approach towards the realization of this goal is to emphasize the importance of modelling scientific phenomena. The VModel project at Northwestern University is conducting research into the design of educational technology that supports students learning by modelling in middle school physical science curricula. This paper discusses the design of supports for student modeling in the Vmodel software.

1. INTRODUCTION

Creating a model is a particularly good way of noticing where more information is required to understand a phenomenon (Penner, 2001). Models improve understanding of complex systems by revealing structures in the system through explication of entities and relationships. Conceptual modelling is a graphical representation technique that utilizes text and diagrammatic forms to describe systems. Conceptual modelling software has been widely deployed in schools as a tool for learning and articulation (Stratford, 1997). Our approach is to provide students with a software-based conceptual modelling environment that supports the articulation and qualitative simulation of their knowledge of physical systems.

VModel (Forbus, Carney, Harris and Sherin, 2001) uses a visual representation of modeling conventions similar to concept maps. The modelling ontology is based on the conceptual entities and operations described by qualitative process theory. This ontology can be used to represent common sense reasoning about physical systems (Forbus and Gentner, 1986). To guide students through the modelling process we have implemented coaching supports in response to student questions. This paper describes a framework for providing modelling support based on categories of questions that students ask and mistakes they make when using VModel.

2. SUPPORT FRAMEWORK

The VModel software is being developed and introduced into the classroom as part of an iterative design-based research project. The software is its third classroom trial in the Chicago public school system. While engaged in a conceptual modelling task using software, students must understand the conceptual modelling primitives, decompose the problem/task into modelling primitives, and lastly relate these primitives to one another. There are opportunities for confusion or errors at any of these points. One aspect of our post-trial analysis has been to feed interpretations of student questions as well as commonly made mistakes back into the software design in the form of coaching supports. Analysis of previous trials has revealed emergent categories of student questions and mistakes. (Table 1).

Table 1. Categories of Student Questions and Mistakes

Category	Example Student Questions or Mistakes
Software-based	How do I use the software?
	When can I switch modes?
Content-related	How many deer are in the population?
Task-related	How do I know when I'm done?
Representational	How can I show that the deer population is decreasing due to predation?
Unanticipated	I'm stuck and I need help with this model – What do I do now?
Syntactic	Intended syntax is violated, e.g. arrow is backwards "Can I use a requires link here?"
Grammatical	Entities are mislabelled, e.g. a process labelled "tree." "Is food a substance?"
System-level	Model as a whole does not tell a coherent causal story.
Explanatory/Predictive	Student cannot use the model to explain how a particular parameter is influenced
	Student does not make a connection between the model and observations or data.

2.1. External or Static Supports

Questions in the Software-based, Content-related, Task-related, and Representational categories are currently supported either through external curricula and materials, or via an embedded HTML-based help system. An email interface allows unanticipated questions to be sent to a remote, server-based coaching agent for analysis. But, at present, the remote agent only forwards VModel help requests to the appropriate team member for human intervention.

2.2. Dynamic Coaching Feedback

The other categories of questions are addressed through dynamic coach supports built into the software. These utilize a rule-based system, model-tracing, or qualitative simulation to detect problems amongst the elements of a model. Triggered rules help point out to students when they use modeling elements incorrectly, when a more appropriate element is available, or when there are internal inconsistencies or vague constructions in student models. When the coach detects a problem, it indicates the problem in two non-intrusive ways: 1. a mode-based, iconic (generally smiling) coaching agent in the corner of the screen becomes puzzled-looking or, with severe errors frowns at the student, and 2. the parts of the model related to the problem are coloured in red. Students can query the coach as to the nature of the problem by clicking on the coach icon or by right-clicking on the indicated modelling element. The coach responds by displaying its analysis of the current state of the model, including known problems, information concerning user-selected modelling elements, and an English-text synopsis of the model.

Feedback on Syntax: The syntax coach analyses a model's structure using a rule-based table look-up. Examples of a syntax rules are: "A direct influence relationship must be connected between a process/rate and any type of parameter else indicate a syntax problem" and "No type of parameter can have both a direct influence and an indirect influence as input."

Feedback on Grammar: Feedback on modelling grammar is given to the students by restating their model in a natural-language synopsis. For example: "The rate of heat exchange decreases the temperature of coffee and increases the temperature of cup." If grammatical problems exist, the English text will make little sense. For example: "Trixie eating does eating."

System-Level and Explanatory/Predictive Feedback: Even when modelling elements are connected with proper syntax, system-level feedback is required to inform the student of problems that can only be detected via model-tracing and/or simulation. For example, model-tracing is required to determine if the student have a loop of influences that exclude a process, and so will not change during simulation. Students are asked to make predictions about the direction of change in parameters, and predictions are verified based on the results of the qualitative simulation. To help students see the causal relationships better, this simulation is animated. Text explanations of each model are given by describing the causal inputs to a selected modelling element.

3. REFERENCES

Forbus, K, Carney, K, Harris, R, Sherin, B. (2001) Modeling Environment for Middle-School Students: a Progress Report. In G. Biswas ed. *Papers from the 2001 Qualitative Reasoning Workshop* Stoughton, WI, The Printing House.

Forbus, K, and Gentner, D. (1986) Learning Physical Domains: Toward a Theoretical Framework. In R. Michalski, J. Carbonell & T. Mitchell (eds.) Machine learning: An Artificial Intelligence Approach II. USA, Morgan Knaufman.

Penner, D. (2001) Cognition Computers and Synthetic Science: Building Knowledge and Meaning Through Modeling. Review of Research in Education, 25: 1-36

Stratford, Steven J (1997) A Review of Computer-Based Model Research in Precollege Science Classrooms. Journal of Computers in Mathematics and Science Teaching; v16 n1 p3-23 1997.