Using modeling to support integration and reuse of knowledge in school science: *Vmodel*, a new educational technology

Karen Carney
Qualitative Reasoning Group
Northwestern University
1890 Maple Avenue
Evanston, IL, 60201, USA
k-carney@northwestern.edu

Kenneth D. Forbus
Qualitative Reasoning Group
Northwestern University
1890 Maple Avenue
Evanston, IL, 60201, USA
forbus@northwestern.edu

Leo C. Ureel II
Qualitative Reasoning Group
Northwestern University
1890 Maple Avenue
Evanston, IL, 60201, USA
ureel@cs.northwestern.edu

Bruce L. Sherin
School of Education and Social Policy
Northwestern University
2115 N. Campus Drive
Evanston, IL, 60201, USA
bsherin@northwestern.edu

Introduction:

Computer-supported modeling can enable powerful student learning about systems. It can help a student to be articulate about relationships between entities in systems and how these relationships give rise to system behavior. A wide variety of modeling tools have been developed successfully and deployed in pre-college classrooms. (Stratford, 1997, Jackson et. al, 1996, Miller & Brough, 1991, Mandinach 1989, Ogborn, 1996, Wilinsky and Resnick 1999.) Modeling software can enable learning through reflection and revision models. Making a model may help students to recognize areas in which their understanding may be incomplete, and, as with other representations, the model may serve to make ideas available to others for discussion and collaboration (Roth and McGinn, 1998).

However, existing educational modeling systems often treat each modeling task in an isolated fashion, with no connection to other models made under different circumstances. This misses opportunities to help students see that the same principles and processes operate across a broad range of situations. (For example, the basic idea of heat flow is relevant to chemistry, biology, atmospheric physics and many other areas that, on the surface, appear unrelated.) We believe that modeling software can be designed so that it can foster students' reuse of knowledge across domains and integration of models with each other.

This paper outlines design steps being undertaken to create *Vmodel*, modeling software that allows middle and high school aged students to create, use and reuse models. Through a combination of software and classroom support, we believe that students can learn to actively reuse and integrate different models with each other. This can include inserting a model into another model (for example, a model of a heat flow could be put into a model of a solar house). This could also mean reusing models across domains, recognizing when systems have underlying causal or system similarities even when they differ in specifics.

Dimensions of design:

The *Vmodel* software currently under development at Northwestern University combines elements from traditional systems modeling software, concept mapping and argumentation environments to help students in creating and justifying models. There are several important dimensions of design in Vmodel: the modeling elements and underlying modeling ontology available to students, feedback and coaching for students, and library functionality which allows reuse and integration of models or parts of models.

The modeling elements and underlying ontology

In order for any modeling software to generate feedback for the user, the user must construct models from a limited, machine-understandable ontology of modeling elements. We believe that this constraint can help students learn. Breaking down the world systematically is the first step in making a model, and one with which that students often have no experience. By providing a flexible but limited set of elements into which they can break the world will give students a set of terms with which they can break down any modelable situation they encounter. This in essence gives them a vocabulary for thinking about systems, and keeps them from having to relearn ways of breaking down the world in each new situation. Using the same elements to make different models can serve to make models ontologically compatible with each other, facilitating integration of one model with another, and allowing students to more easily compare models. Using a limited suite of elements can underscore structural and causal similarities across models representationally, helping students recognize when models are analogous more easily.

Vmodel contains a modeling ontology that attempts to cut the world into appropriate, graspable domain-neutral chunks. We are designing for young (pre-algebra) students who lack experience with formal representation and higher mathematics. This means that the modeling ontology must make close contact with the student's common sense physical notions. For that reason, we base the ontology on qualitative process theory (Forbus, 1984). Qualitative modeling provides formalisms for expressing intuitive, causal models and the reasoning techniques needed to generate predictions and explanations from them. A student making a model using qualitative modeling can specify partial knowledge about causal relationships, building up a model over time, without having to specify in advance all potentially relevant functions. Also, this modeling ontology allows a greater range of expressions than a purely mathematical system would allow, including representation of the circumstances under which a model is relevant and the conditions that define states and enable processes.

Visually, the modeling ontology is divided into two different basic kinds of representational forms: entities (represented by boxes) and relations among them, (represented by arrows.) Students assemble their models from these elements.

Entities:

Each entity has a specified type, such as `Thing` or `Number-Attribute`. These types are drawn from Vmodel's general ontology. Students are expected to give entities instance-specific labels.

Figure one: Three boxes with instance specific labels.

Basic Objects	**Describers**
Thing	Attribute
Group-Thing	State or form
Substance	**Number Describers (Quantities)**
Process	Amount
	Rate
	Level
	Chance of Happening
	Number Attribute

Box one: Types of entities in the beta version of Vmodel

Library functionality allows students to extend the object ontology as they develop more sophisticated understandings and modeling techniques.

Links:

Students relate basic entities with links. Links can relate objects to descriptors, show configuration relationships or show causal or functional relationships among entities. The arrow labels are fixed. Unlike the boxes, students do not pick a domain specific label for relationships. We do this to reduce idiosyncratic expressions, to facilitate comparisons of models between students and to simplify the coaching software.

Descriptive Relationships	*Causal or Functional Relationships*
Describing basic objects:	**Relationships between quantities**:
Has-Attribute	linked-to
Does	linked-opposite-to
Can-do	**Relationships between processes and quantities:**
Has State or form	increases
Configuration relations between basic objects:	decreases
Touches	**Controllers on relationships**:
Contains	requires
Part of	prevents
Moves from/to	**Catch-all**
Magnitude Comparisons between number describers:	affects
Greater/less than	
Equals	

Box two, link types in the Beta version of Vmodel.

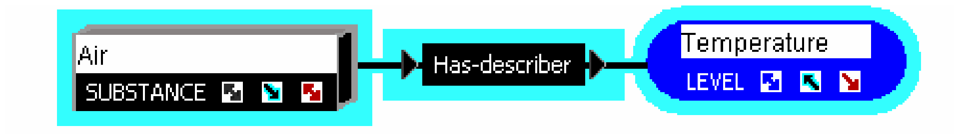

Figure 2: a simple relational statement in Vmodel. A colored shell around the element and its describer group them together and distinguish them at a glance from other described entities.

Coaching and feedback

All expressions made with the modeling software are done through the combination of entities and relations. Feedback is given according to the assembly of these elements. In order to aid students in learning how to use these elements, we are developing a combination of supports including ramp-up activities and on line help.

A coach is being implemented in the Vmodel software to effectively support the students in modeling activities. We envision two types of coaches:

1. Local coaches to provide immediate feedback to students based on their use of modeling elements,
2. A more powerful server-based coach capable of more powerful model analysis.

The local coach built into the software utilizes a rule-based scripting system to react to changes in a student's model throughout the design process. Triggered rules will help point out to students when they use modeling elements incorrectly. Students are notified of incorrect use of modeling elements by a screen icon, which changes from a smiling face to a confused face when students make a strange construction. This is accompanied by a screen indication as to which element is incorrectly used. The student can then access an automatically generated text hint or deeper hypertext help on model construction in order to fix their models. Preliminary classroom studies show that students and teachers respond favorably to this coaching, and correct their models when given prodding from the coach.

Future applications of the local coach are under design. They will help alert students as to when a more appropriate element is available, or when there are internal inconsistencies or vague constructions in student models.

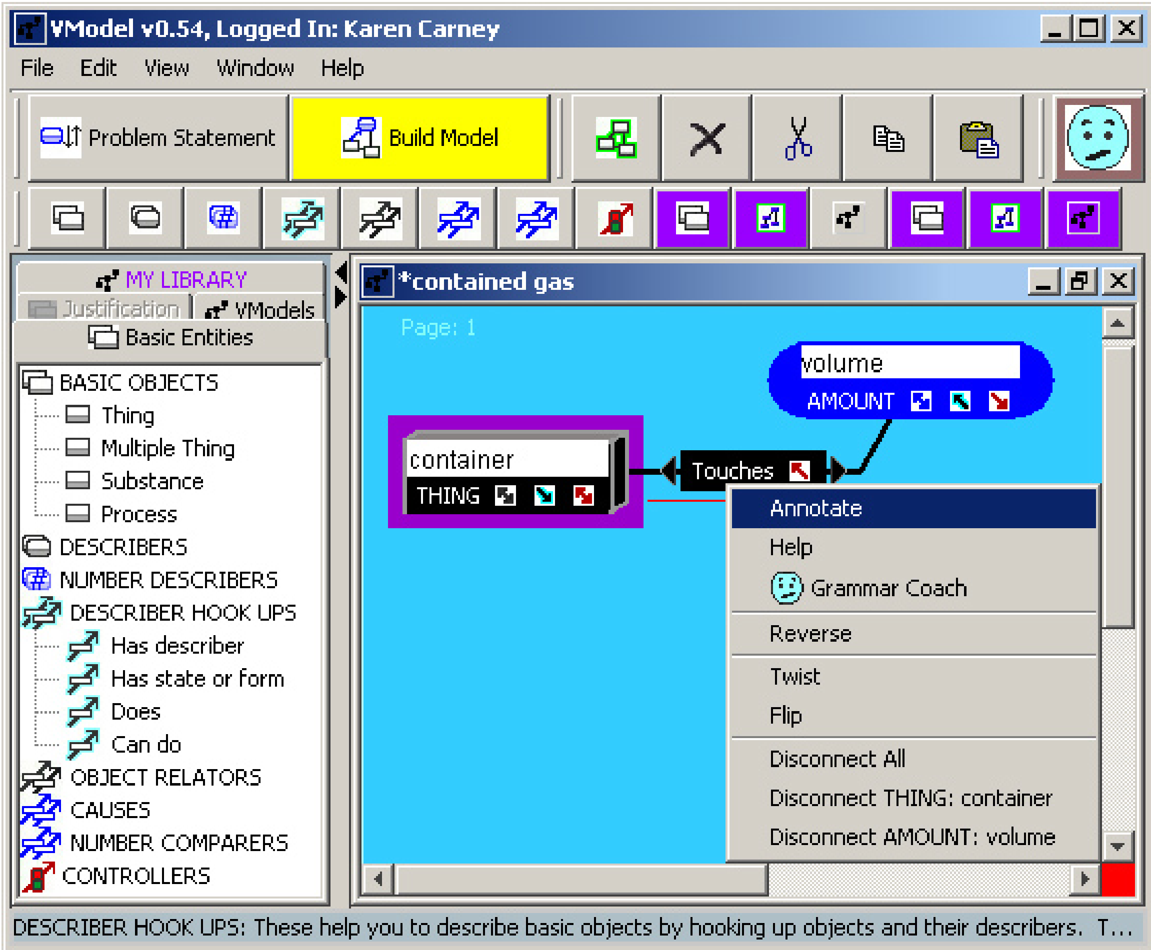

Figure three: Vmodel coaching in action. Note the face in the upper right hand corner and the underlined link. Students can then right click on the incorrect relationship and either access help directly receive further explanation from the grammar coach.

The server-based coach allows students to send questions or models to a remote system for help and analysis. This system currently forwards questions to human operators who can examine models by hand and reply to students via email. An automated coach is being developed that will utilize powerful qualitative modeling techniques to do a more sophisticated analysis of student models. The automated coach is based on portions of an existing server-based coach for engineering software (Forbus et al, 1998) that will utilize current analogical reasoning techniques (Forbus et al, 1995) and the student's model library to help students recognize when an existing model might be appropriate for inclusion in or analogous to the current model. A benefit to this server-based approach is the possibility of comparison between the current model and peer- or teacher-constructed models.

An important kind of feedback concerns how well a model explains something. Students using Vmodel will be asked to explicitly state a phenomenon or problem that is to be explained by the model, by choosing a quantity and a direction of change for that quantity (i.e. the temperature of the gas is rising.) Both kinds of coaching software will use standard qualitative reasoning techniques to analyze the relationships that the student includes in his or her model to provide feedback as to whether the model explains the observed behavior. From this feedback, students can refine their models until their observations are satisfactorily explained.

The model library:

The support for reuse built into *Vmodel* is a model library. The model library provides a catalogue of model elements (both primitive and student-built) and a portfolio for student work. Students may use these stored elements in the creation of new models. A student's model library comes to contain any models he or she has made, as well as any new entities he or she has created. The model library also has a *Vmodels* section that can be seeded by teachers or developers with entities that might be appropriate for an entire class to consider.

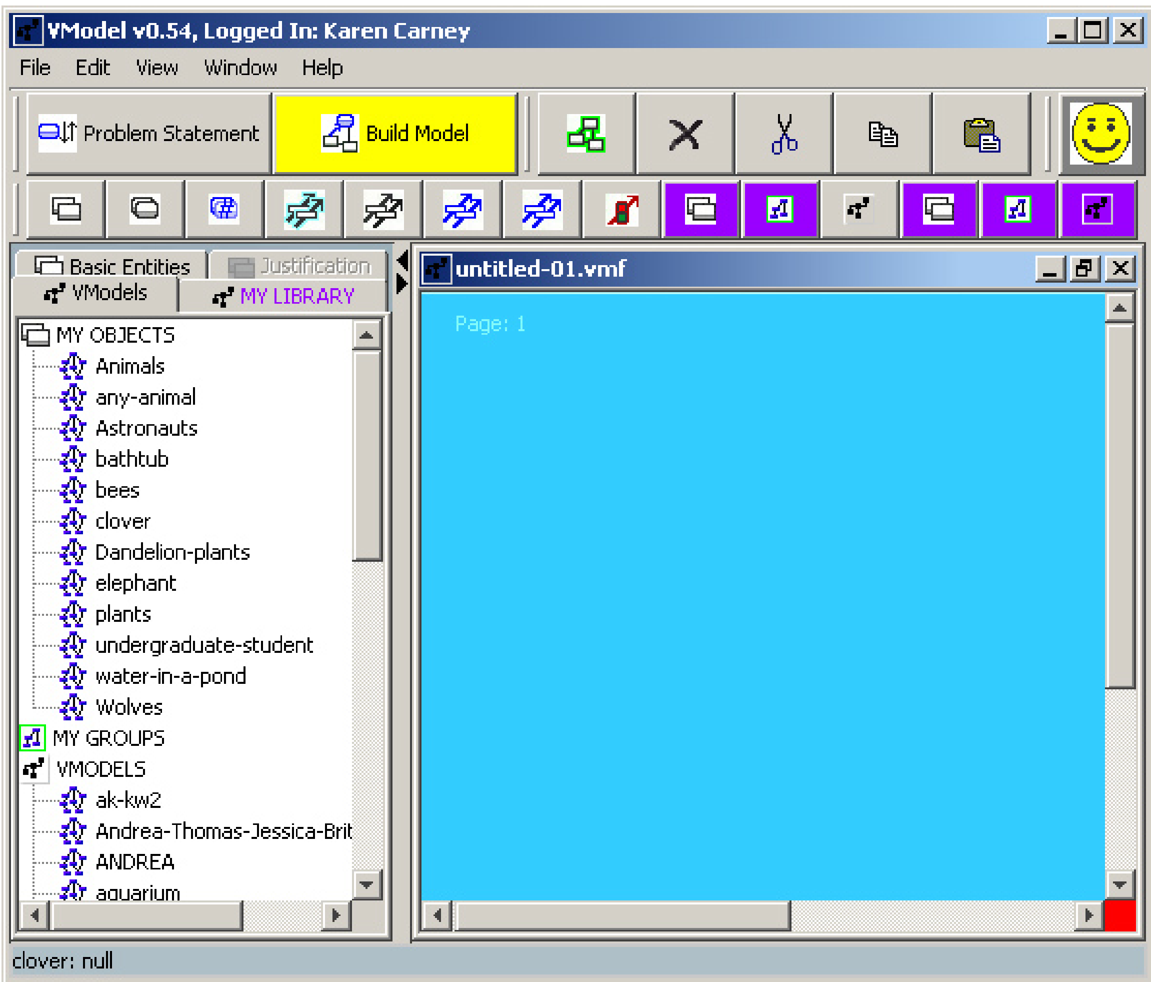

Figure four: The Vmodel interface with the Vmodels section of the model library open.

Supports for reuse

The simplest form of reuse is using part or all of one model in another. For example, a model of a piston may include a more general model of a contained gas. Including fully specified models in models, however, quickly becomes overwhelming visually. In addition, there are many instances when an existing model is not applicable wholesale in a new model, but aspects of it can and should be retained and applied. Consequently, we provide tools for students to "box up" a model or parts of it to create new primitive modeling elements. Students can take these models and use them as modeling building blocks in other models. These student-created entities can be created at various levels of resolution ranging from fairly specific (e.g. "pet" or "contained gas") to more general ("living thing" or "fluid medium"). These new types, like any other modeling entities, can be labeled with instance-specific labels in a new model.

An important instructional goal of the *Vmodel* project is to help students master the art of creating and using good abstractions in models. We allow students to specify some aspects of their models as variables, or things that can change according to situation. In figure five, the entity “container” has been variablized, and in figure six, the piston is used as a local value for the container.

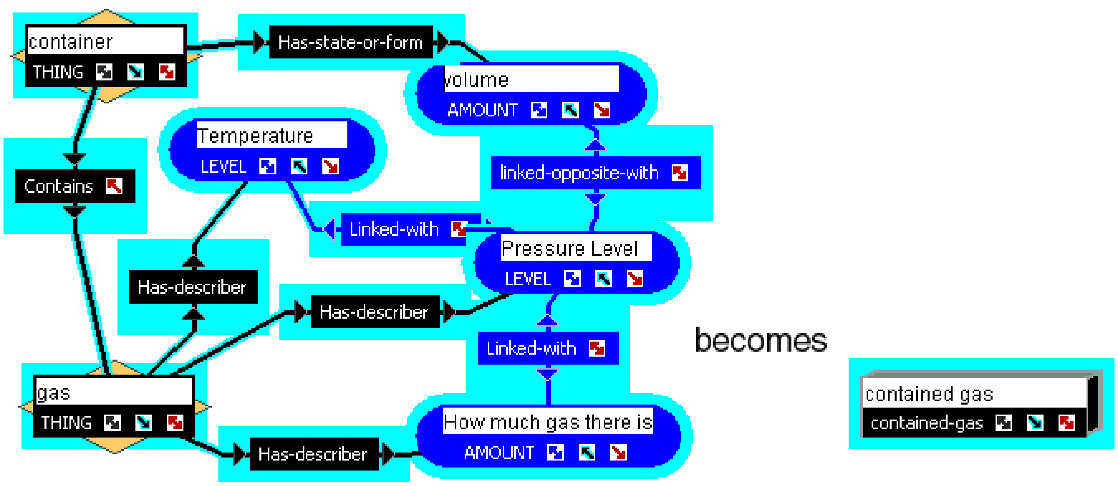

Figure five, the creation of a new entity type from a model fragment. Variablized entities (such the name of the gas or the name of the container are denoted by diamond shaped boxes.

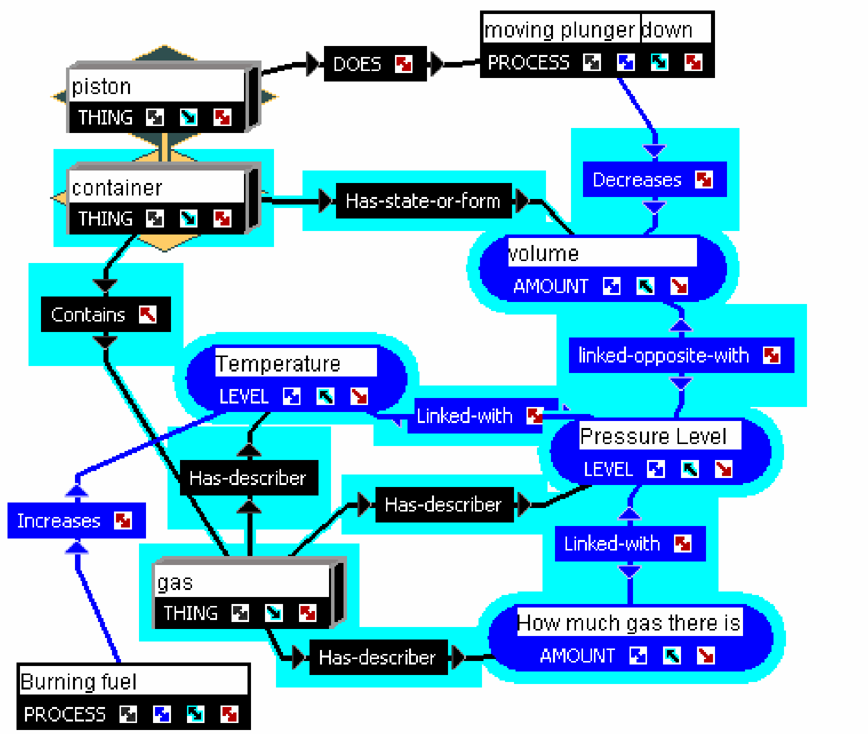

Figure six: a model of a piston containing a model of contained gas. The piston is a locally bound value of the variablized container.

Vmodel in the classroom

We are working closely with teachers to develop ways of introducing and reinforcing modeling as a way of doing science in the classroom. The classroom portion of this research is being conducted as part of the NSF Center for Learning Technologies in Urban Schools, (LeTUS) a partnership involving Northwestern University, University of Michigan, and the Chicago and Detroit Public School systems. In collaboration with CPS teachers, over the last year we have been developing and using LeTUS curricula as the initial settings for our modeling work.

In parallel with the curriculum development work, we have been carrying out pilot studies in CPS classrooms to drive the visual notation design and the software design. In the early design phase, we used pencil and paper studies, which were very encouraging. We explored what relational vocabularies are most natural for students, by providing them with building blocks that have pre-built labels but also blank versions that they fill in. We are also beginning to characterize the kinds of difficulties students have with the Vmodel representation system, and cataloguing ways in which students describe complex phenomena. Student trials using the alpha version of Vmodel software (without Model Library or coaching) were completed in the 2000-2001 school year, and beta version trials are underway in classroom settings. These experiences are driving the design of the software coaches and other external supports.

Works Cited

American Association for the Advancement of Science (1989) Science for all Americans: a Project 2061 reports on literacy goals in science, mathematics, and technology. Washington, D.C., American Association for the Advancement of Science

Carney, K, and Kalathil, R. (2000) "Student Modeling in Science: Influences of Prior Experience with Representational Norms" Presented at the Apr. 2000 meeting of the American Educational Research Association, New Orleans, April, 2000

Forbus, K. (1984) Qualitative Process theory. Artificial Intelligence, **24**,

Forbus, K. (1996) "Qualitative Reasoning". CRC Handbook of Computer Science and Engineering. CRC Press, 1996.

Forbus, K., Everett, J., Ureel, L., Brokowski, M., Baher, J., and Kuehne, S. (1998). Distributed coaching for an intelligent learning environment. *Proceedings of the Twelfth International Workshop on Qualitative Physics* (QR98), Cape Cod, MA, USA.

Forbus, K., Gentner, D. and Law, K. (1995.) MAC/FAC: A model of Similarity-based Retrieval. *Cognitive Science*, 19(2), April-June, pp 141-205.

Jackson, S., Stratford, S.J., Krajcik, J.S. and Soloway, E. (1996). A Learner-Centered Tool for Students Building Models. Communications of the ACM, 39(4), 48 - 50.

Mandinach, E.B., 1989, Model-Building and the Use of Computer Simulation of Dynamic Systems, Journal of Educational Computing Research 5(2) 221-243

Miller, R.S., & Brough, D.R. (1991) "Quantitative and Semi-Quantitative Computer Tools For Exploratory Learning." Imperial College Research Report, DOC91/29

Ogborn, Jon. (1999) "Modeling Clay for Learning and Thinking." Chapt.1. p.5-37 In Feurzeig, W. and Roberts, N. (eds.) Modeling and Simulation in Science and Mathematics Education. New York, Springer-Verlag.

Reif, F. and Larkin, J.H., (1991) Cognition in Scientific and Everyday Domains: Comparison and Learning Implications. Journal of Research in Science Teaching, 28, 733-760.

Roth, W. and McGinn M. (1998) Inscriptions: Towards a theory of representing as social practice. Review of Educational Research. 68(1)

Shrader, G., Lindgren, R., and Sherin B. (2000) A pedagogicaly-oriented analysis of the modeling Process. Presented at the Annual Meeting of the American Educational Research Association in New Orleans, LA. April 24-28, 2000.

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.

Wilinsky, Uri, and Resnick, Mitchel (1999) Thinking in Levels: A Dynamic Systems Approach to Making Sense of the World. Journal of Science Education and Technology. V8 n1 p. 3-19