Qualitative modeling for middle-school students

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

Karen Carney
School of Education
and Social Policy
Northwestern University
2115 N. Campus Drive
Evanston, IL, 60201, USA
k-carney@northwestern.edu

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

Abstract

Qualitative modeling offers the potential for engaging middle-school students in scientific modeling. This paper discusses the design decisions underlying VModel, a qualitative modeling environment that has been used in several studies in Chicago Public School classrooms. We discuss how these decisions were influenced by the constraints imposed by the conceptual development of the students and the middle school curriculum and environment. We describe the simplified subset of QP theory used, pedagogical agents and other software scaffolds, and how within-state qualitative simulation (i.e., influence resolution) helps students improve their models. Some evidence that VModel can improve learning is provided, and plans for future work discussed.

Introduction

Modeling is a central skill in scientific reasoning. Supporting students in articulating and reasoning with models can lead to deeper, systematic understanding of science [Penner, 2001]. Unfortunately, as typically taught, modeling requires high-school level mathematics, making it inaccessible to younger students. Qualitative modeling offers the potential for engaging younger, middle-school students (ages 9-13) in scientific modeling. Achieving this potential requires careful design. The constraints of the students' conceptual development, the middle school curriculum, and the school environment must all be taken into account.

This paper describes some of the key design decisions in Vmodel, a qualitative modeling environment that uses QP theory [Forbus, 1984], expressed in a concept map notation, to enable students to express their conceptual models and get feedback about them. VModel has been successfully used in several studies in the Chicago Public Schools, and will soon be available, along with sample curriculum materials, as open-source software.

Section 2 discusses the subset of QP theory used, focusing on fit with middle-school student capabilities and needs. Section 3 discusses the ontology of the system. Section 4 discusses the interface design, and how it scaffolds the modeling task. Section 5 discusses the use of qualitative simulation to provide students with feedback on their models. Section 6 outlines some results from school studies, and Section 7 provides a summary and discusses future work.

2. What aspects of QR do middle-school students need?

Qualitative representations capture the intuitive, causal aspects of many human mental models. The QR community has explored a wide range of representations and techniques, pursuing its goal to capture the breadth of qualitative reasoning, ranging from the person in the street to the expertise of scientists and engineers. Not all of QR can be relevant to middle-school learning goals, since students of that age simply do not have the mathematical sophistication of scientists and engineers. We have found a subset of qualitative representations and reasoning techniques that seem useful in bringing students into the community of modelers, speaking in some ways the same language as scientists, but leaving out that which they do not seem ready for. The constraints that led us to this subset come from three sources:

1. The cognitive capacities of the students. One reason for teaching modeling is to help students learn to deal with formalisms and abstraction in a more tractable way. The idea of a variable, either logical or mathematical, does not come naturally to most students. It is a struggle, and they have to be led to it in stages.
2. The middle-school curriculum. Middle-school science is concerned with the basic entities and processes that occur in the world, and how they are related to each other. Surveys of middle-school curricula suggest that indeed QR representations are an excellent fit for a large portion of what is learned in middle-school inquiry [Schwarz & Sherin, 2002].
3. The middle-school environment. US schools vary widely in the amount of resources available. We worked with some of the poorest in the Chicago Public School system. Classroom computers are shared and typically several years behind those available elsewhere. Many students do not have access to computers at home or outside the classroom. Network access is sporadic. Moreover, there are significant

constraints on teacher time and the topics they must cover, which make many experiments that would be useful in principle impossible.

The three key QR design decisions we made for VModel were the use of within-state qualitative simulation, hand instantiation of models, and single-rate physical processes. We discuss each in turn.

Within-state qualitative simulation. One of the basic choices in using qualitative modeling is the kind of qualitative reasoning to be performed by the system. Middle-school students are learning basic causal mechanisms of the world, and how to formulate a model out of what starts as a fairly incoherent set of ideas. Reasoning through a causal chain within a situation (e.g., the microwave oven does heating. Heating causes the water's heat to increase, which causes its temperature to rise) is a central task for them. Middle-school curricula rarely deal with complex dynamics, and students are not expected to learn those notions. Such notions need to be grounded in an understanding of what parts comprise a system and how those parts are related, which is what they are in the process of learning. Consequently, we use within-state qualitative simulation, more similar to early QR work [cf. Forbus & Stevens, 1981; de Kleer & Brown, 1984] than more commonly used multi-state qualitative simulators [cf. Kuipers 1994; Bouwer & Bredeweg, 2001].

Using within-state qualitative reasoning has several significant advantages. It is of course extremely fast and efficient. It avoids the complexity of multi-state generation, where an ill-constrained model could lead to a depressingly huge number of states for the student to deal with. It simplifies the generation of supporting textual arguments (see Section 5).

While this choice may seem surprising, earlier efforts to use QR with students indicate that this is a reasonable choice. In Betty's Brain [Biswas *et al,* 2001], all reasoning occurred within a single qualitative state. Explorations by Bredeweg and his colleagues with high-school students [Goddijn *et al* 2003; Machado & Bredeweg 2003] suggest that understanding multi-state qualitative simulations is not straightforward even for high-school students. There is always the possibility that someone will discover methods for making the results of multi-state qualitative simulations crystal-clear to the vast majority of middle-school students, but evidence to date suggests that this would be quite a discovery indeed.

Hand-instantiation of models. One of the hallmarks of QR has been the formalization of automated modeling techniques, where a model for a scenario is automatically constructed from a structural description and task constraints [cf. Falkenhainer & Forbus, 1991]. However, we ended up eschewing automated modeling techniques in favor of letting students build scenario-specific models by hand. There are several reasons for this. First, our experience with middle-school students to date suggests that most of them may not be ready for creating logically quantified descriptions, even when supported by a visual notation. Second, the model formulation problem for an open-ended set of structural descriptions has not been solved, nor is it likely to be anytime soon. It would require a broad understanding of everyday life, rather than the more narrowly constrained structural descriptions found in science and engineering. We believe that such algorithms are possible, but that is an empirical research question. Third, even if such algorithms were available, it would be wise to only make them available to students once they have done some modeling by hand. We have found that having them work through the details of multiple models helps them to appreciate the underlying common principles [White, 1993].

In order to lead the students towards the idea of formulating generally applicable principles, VModel does incorporate a *Model Library*. The Model Library serves two purposes. First, all of a student's models are stored there, to facilitate comparison between the current modeling task and prior modeling assignments. Second, students can select a subset of their current model to store in the library as a new building block, which can then be used in subsequent models by wiring it into their current model. Again, we kept this feature as simple as possible, based on what our experience suggested most middle-school students could handle. Students can only add a single layer of more specific entities and processes, rather than, say, constructing a lattice of types as commonly found in KR systems.

Single-rate physical processes. QP theory places no limit on the number of continuous parameters a process can have. This freedom unfortunately causes many problems for students. Students initially tend to leave out rate parameters, since their linguistic descriptions of phenomena rarely mention them (e.g., consumption increases the stored energy). When they do mention rates, they may not tie the rate to a physical process. Looking over the middle-school curriculum, we were unable to find cases where a process with more than a single rate parameter was required. Should such a case be found, an additional process could be introduced (as an "effect" of the main process) to provide another parameter. Consequently, we restricted processes to always have a single rate parameter.

The combination of hand-instantiation of models with within-state qualitative reasoning leads to dramatic simplifications in the semantics of the models. Every statement in the model is assumed by default to be true. Every physical process in the model is considered to be active. General-purpose handling of contradictions, negations, and unknown facts are all unnecessary. This makes the implementation drastically simpler, which is important given the kinds of machines typically found in middle-school classrooms.

3. Fitting QP theory to middle-school students

While the concepts of qualitative modeling provide a natural language for the modeling needs of middle-school students, the notations traditionally used in qualitative modeling do not. Asking students to learn predicate calculus in order to express intuitive ideas is of course counterproductive, which is why we, like others [cf. Biswas *et al* 2001, Bouwer & Bredeweg 2001], developed a visual notation for modeling. The visual properties of the notation are discussed in Section 4 since they are heavily connected to the interface design; here we focus on the ontological choices made.

Again, our goal is to keep things as simple as possible, encouraging students to think in the terms of causal models involving continuous parameters. To that end, our ontology has six top-level categories:

- *Physical process*. We want students to articulate what processes are occurring in a situation (e.g., consumption, reproduction, heating, etc.), ergo this must be an explicit category.
- *Basic stuff*. These are types of entities. We have broken this down into three sub-types. *Thing* is a generic type for individual objects. *Multiple Thing* is for representing populations and other groups of things where it makes sense to count them. *Substance* is for representing continuous media.
- *Parameters*. These are continuous quantities. There are four subtypes: *Parameter* is the most general, for brainstorming when the student isn't worried about fine distinctions yet. *Amount* and *Level* correspond to extensive and intensive parameters, respectively. Distinguishing these in the system is important because it is a distinction that students must learn. *Rate* is used for rate parameters of processes.
- *Connectors*. These provide a rudimentary language of structural descriptions. *Touches* and *Contains* enable geometric information to be expressed. *isPartOf* expresses part-whole relationships. *Does* ties a process to a particular entity when appropriate.
- *Requires*. This relationship links a process to the statements that are required for it to be active.
- *Causes*. These are the influences of QP theory. *Increases* and *Decreases* are direct influences, i.e., I+ and I- respectively. *Influences* and *InfluencesOpposite* are indirect influences, i.e., $\propto_{Q+}$ and $\propto_{Q-}$ respectively.
- *Comparisons*. These are the usual ordinal relations, *GreaterThan*, *LessThan*, and *Equals*.

The top-level structure reflects the logic of the modeling task rather than an optimally structured ontology. After all, Connectors, Causes, and Comparisons are all binary relations, so standard practice in AI would reify this intermediate vocabulary and place them as children under it. But this is not a distinction that is important to what the students are learning at this point, and needlessly complicates the ideas for them.

This ontology also reflects three years of simplification and tuning, first through pencil and paper studies in classrooms and then through multiple versions of the software. For example we used to have a category of Controllers, relationships that linked a process to the conditions under which it is applicable. We found that distinctions beyond the single Requires relation were unnecessary. We also have tried a variety of weaker causal relationships, including those which didn't specify sign or even direction of causation, to make it easier for students to use VModel even in the earliest stages of brainstorming. We found that once students used these very weak relationships they rarely refined them, and consequently we eliminated them.

The astute reader will notice that there is no relationship in the ontology that links a quantity to an entity. There is of course such a relationship inside the system, but we found that reifying it for students did more harm than good. We never found a name and explanation for them that didn't confuse students, and it turns out students are perfectly happy not articulating this connection, given the intimate nature of the relationship between an entity and its continuous properties.

One critical problem was finding student-friendly names for the concepts. The words used have to be close enough to the students' everyday usage so that they do not become another barrier for them. Yet they must learn to use the terms in a technical way, just as a physicist uses "force" differently. This enforcement comes out of the software coaches in the system, as discussed in Sections 4 and 5.

4. Interface design decisions

Educational software, perhaps more than other software, lives or dies by its interface. We start by describing why the visual elements of models are depicted as they are, then discuss the decomposition of the software into modes to better support the modeling process, and the pedagogical agents that help students with modeling.

Visual notation

Our visual notation uses standard concept map conventions, i.e., nodes represent entities and labeled arcs represent relationships, but with some extensions that exploit the built-in ontology. As usual, nodes can have labels, text strings that that the student uses to describe their intended meaning (e.g., "population" or "reproduction"). The text strings are used in explanations generated by the coaches, which helps enforce the intended semantics (see Section 5). However, relationship labels cannot be edited, since we want to regularize student models and enable them to be simulated. The ontological types of nodes is redundantly displayed, using both text in the node and by varying its shape and color, with entities displayed as black square boxes, parameters as blue ovals, and processes as black square boxes with an embedded blue oval representing its rate.

Relations are also coded by shape and color:

- Connectors: black in a thin square box.
- Requires: black with square box.
- Comparisons: blue with square box.
- Causes: blue with oval box.

This color coding is used to help remind students what kinds of entities can be related. Except for Requires, all relationships connect nodes of the same color. (See Figure 1).

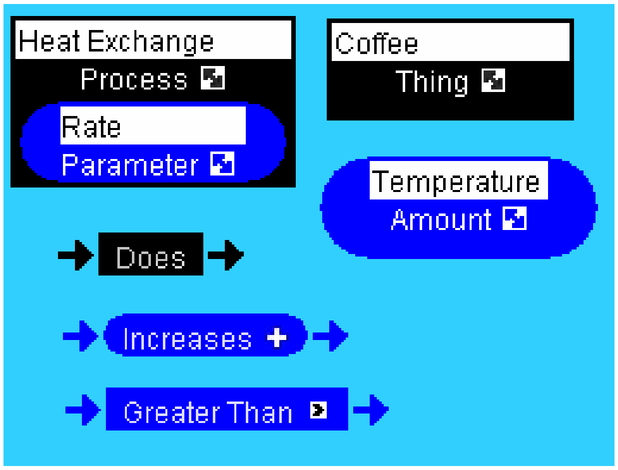

Figure 1: Visual iconography for entities and relationships.

As noted in Section 3, no explicit relationship linking a parameter to its entity is available in the ontology. Visually such connections are depicted by unlabelled lines, a good convention because this kind of relationship is extremely common. These links are introduced via a right-click gesture on the entity in the interface, which then provides a menu of parameters available to link to. (The exception is the rate parameter of a process, which is displayed inside the box representing the process itself, as shown in Figure 1.) The intimate connection between an entity and its parameters is further reinforced by differently colored "skins" that surround each entity and its parameters (see Figure 2).

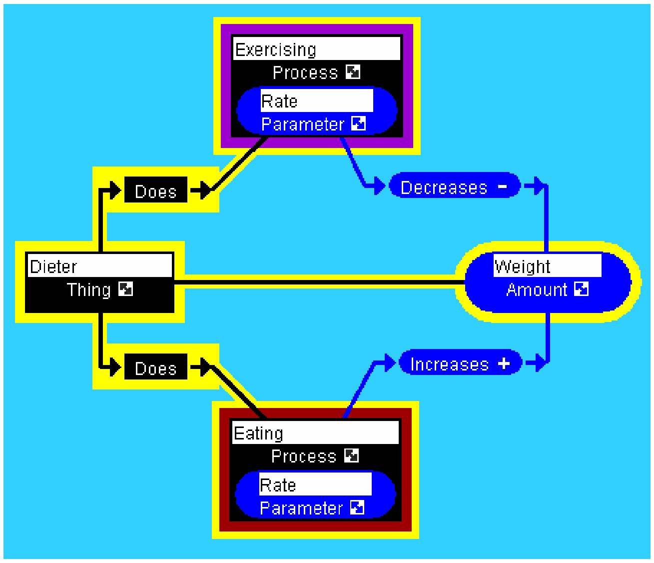

Figure 2: Reinforcing relationships with "skins".

Interface modes for scaffolding modeling

Students need structure. We found that it was extremely useful to organize the interface into a sequence of modes, each representing a distinct phase in modeling. The modes are *target*, *build*, and *analysis.* We discuss each in turn.

Target Mode: Students are asked to describe their modeling problem in natural language, in the form "the ________ of ________ is increasing/decreasing/staying the same", that is, to make a prediction about the sign of the derivative in what will be a parameter of their model. When they next build the model itself, this target is communicated to the coaches formally via interface gestures on the parameter involved. This enables them to compare the result they eventually achieve via qualitative simulation. A model is considered successful when it produces the target prediction via a plausible explanation.

Target model came about as a direct result of classroom experience. Without it, students tend to treat modeling as a "brain dump", putting in everything they can think of about the subject. The result was a collection of disconnected entities and relationships instead of a single coherent causal story.

Build mode: This mode supports the creation and editing of models. Here is where students have access to both the library of primitives (Section 3) and their Model Library. Most of a students' time is spent in Build mode. To help students, a pedagogical agent, the Builder, is available to provide feedback. The Builder also acts as a gatekeeper, preventing the student from proceeding to the next mode until it is satisfied with the student's model. (The Builder will be discussed in more detail below.)

Analysis mode: This is where qualitative simulation takes place, providing results for the student to reflect upon. Analysis mode is discussed in detail in Section 5.

Pedagogical agents as scaffolds

We have found the use of pedagogical agents [Lester, 1997] to be extremely useful for scaffolding students. We discuss the Builder here; the Professor – the agent for Analysis mode – is discussed in Section 5.

The Builder uses a simple set of rules to enforce good modeling practices. These rules include constraints that enforce the semantics of QP theory (e.g., quantities cannot be both directly and indirectly influenced, only processes can have direct influences) and check the consistency and completeness of their model (e.g., does it make a prediction?). Importantly, the Builder is designed to be non-intrusive. When a model is problematic, the Builder's expression changes accordingly, through a variety of facial expressions. However, it does not offer advice to the student unless asked, or unless the student tries to enter analysis mode when the model is not yet ready. This is important because creating and editing a model often involves temporary inconsistencies and incompleteness.

5. Qualitative simulation to support student learning

As noted above, the representational assumptions we have imposed on VModel drastically simplify reasoning. That is, every statement explicitly appearing in the model is true, and every physical process in it must be active. In effect, the student has specified the process and view structures, so that all that remains to do is resolve influences to find out how the parameters are changing. We further (tacitly) assume that all rates are positive. With these assumptions, the simulation boils down to standard influence resolution in QP theory (cf. [Forbus, 1984]). That is, we start at directly influenced parameters, calculate their Ds (sign of derivative) values, and propagate Ds values through the indirect influences.

Ambiguous influences, both direct and indirect, are handled by asking the student what the result should be. This method is very straightforward and easy for students to understand. A drawback is that a more sophisticated resolution algorithm, drawing on for example order of magnitude representations, could potentially enhance learning by allowing for software-supported discussions of relative rates and magnitudes of effects. However, our experiences in middle-school classrooms to date suggest that this is not a central topic for these students yet.

While much research has been invested in making qualitative simulation fast and efficient, our main problem was slowing it down. That is, we animate the reasoning process, highlighting each parameter being considered in turn, and displaying each result as it is generated. Students seem to find observing this animation helpful in learning how to reason with a causal model. In addition to the animation, students also receive a textual summary of the simulation. For example, this explanation is generated from a student's model of heating water in a microwave oven:

```
There is a process called Heating which

Increases the Heat of Water that is
   INCREASING and which

Influences the Temperature of Water that is
   INCREASING as you predicted
```

The text summary is generated based on the order of computation used for influence resolution. Each ontological type has an associated natural language template that uses text labels associated with the node to produce a string. The templates or physical properties are

```
;; Process Template
[“There is a process called”
    <label-of process>
(when
 <has-outbound-direct-influences process>
   [“which”])]

;; Property Template
[“the” <label-of property>
 (when <belongs-to-entity property>
   [“of” <label-of entity> “that”
    <sign-of property>
    (if (= <sign-of property>

           <prediction-of property>)
      [“as you predicted”]
     [“but you predicted it would be”
      <prediction-of property>])))]
```

The generated text serves several purposes. First, along with the bitmap of the graphically annotated simulation, it provides a concrete result for their modeling effort. The importance of such concrete "take home" products is hard to overestimate. Second, it encourages students to decompose their entities appropriately, because the templates use conventions that we expect them to learn, e.g., "the *<label of property>* of *<label of entity>*" to refer to a parameter of something. This enlists their language skills to help overcome a student tendency to lump together elements of their model: "Carbon dioxide of Oxygen" grates. Finally, the text flags problems found by the simulator. For example, the simulator checks if its results differ from what

Curriculum	Grade	Setting	Modeling platform	N	Pretest correct	Post test correct	Gain
ReNUE (Investigation into causes of local environmental crises)	6 - 7	Chicago Public Schools	Model-it	163	44.73%	51.93%	+7.20%
Water Quality (Investigation into influences on stream quality)	7	Chicago Public Schools	Model-it	34	51.05%	63.63%	+12.58%
Mars Survival Station (Closed systems and resource use)	7	Chicago Public Schools	VModel	27	43.52%	51.80%	+8.28%
River ecosystems	5	Nashville public schools	Betty's brain	no comparable data available			

Table 1: Comparison of modeling environments for middle-school students

the student predicated. If there is a mismatch the problem is noted, e.g.,

```
Level Parameter: Temperature

You predicted that the Level Parameter,
Temperature of Water, would be DECREASING
but instead it remained CONSTANT
```

6. Some Lessons Learned in the Classroom

One of the most depressing lessons of working in classrooms is that controlled experiments are difficult to impossible to carry out. Teachers have a multitude of constraints on them, and controlling what is kept the same versus different from semester to semester to facilitate experiments is not high on their priority list: Their first priority must be, and rightly so, doing the best they can for their students, taking their whole situation into account. For example, we found through repeated experience that a certain length of introductory exercises led to much better modeling performance in students, but sometimes teachers simply could not make that much time available for the part of the curriculum that VModel was being used in. Consequently, the results from classroom experiments are often less solid than can be obtained under more controlled circumstances. On the other hand, it is also well-known in education research that many innovations which work well in laboratory settings or with hand-selected classes of "star" students fail when attempted in more typical classroom settings. Therefore experimentation in real classrooms, while harder to do, is invaluable in assessing such research.

There are two systems that are roughly comparable to VModel in terms of intent and age range of students. Betty's Brain [Biswas *et al,* 2001] is a product of the Teachable Agents project, and supports qualitative modeling of stream environments. The idea is that students are instructing Betty, the persona of the software, on how streams work. They quiz their Bettys to see how well they answer questions about streams, thus providing incentive to improve their models. Like VModel, Betty's Brain uses influence resolution for qualitative simulation, but it has a mechanism for marking some influences as dominating over others. The other system, Model-It [Jackson *et al,* 1996] was developed at University of Michigan. It provides systems dynamics [Forrester, 1996] simulation functionality, with a qualitative layer over it so that students need not see the exact algebraic functions constraining the parameters. Model-It uses numerical simulation to generate results, and does not otherwise incorporate qualitative reasoning or representations. Neither Betty's Brain nor Model-It provide explicit representations of physical processes or an equivalent to VModel's Model Library.

Table 1 illustrates some curricula that these programs have been used with. In the case of Model-It and VModel, they were used in curricula developed as part of the Learning Technologies in Urban Schools NSF Center. LeTUS curricula undergo various evaluations, so we have data available for pre and post tests assessing specific content and analytic ability gains. Post-test gains are modest, but comparable across curricula and modeling platform. To our knowledge, no similar data is available for Betty's Brain studies. Stronger conclusions cannot be supported by these data, since the evaluations were developed by researchers and teachers working on each curriculum independently, and were not standardized across trials. However, it does seem safe to conclude that VModel can help student learning.

One common difficulty encountered by students using all three modeling systems is creating coherent models with explanatory power. This can be seen as a failure to interrelate changes in one part of a model with overall changes of behavior [Davis *et al,* 2003], not understanding which entities are important to the overall causal story [Carney *et al,* in prep; Shrader *et al,* 2000], confusing a model which runs with a model which is correct [Zhang *et al*, 2002; Witcomb, CPS teacher, personal communication], or not focusing on an overall goal when modeling [Zhang *et al,* 2002]. This has become less of a problem with VModel now that Target Mode and the feedback from coaches has been introduced. Both of these systems help foster more correct representational choices and more focused models [Carney *et al*, in preparation].

6. Discussion and Future Work

We have described the key design decisions in VModel which enable it to help middle-school students learn scientific modeling at a conceptual level. Perhaps the biggest surprise is how much utility there is in very simple qualitative representations and reasoning: Many of the advanced techniques developed by the QR community, while crucial for scientific and engineering applications, seem to be overkill for many educational purposes. This is very encouraging news for those who want to see QR more broadly used in education.

To us it remains an open question whether or not the majority of middle-school students could learn to build general domain theories. On one hand, many of today's scientists and engineers cannot, relying instead on general models created by others. On the other hand, it was once thought that reading and writing were skills that most people could not master, and would have no use for.

While VModel is aimed at middle-school students, we think that similar modeling environments have potential for a wide range of students, engineers, and scientists. Consequently, we are developing *VModel Pro*, a qualitative modeling system that will use the full power of compositional modeling and multi-state qualitative reasoning. A pervasive theme in VModel is the importance of appropriate scaffolds and supports, and we plan to expand this in VModel Pro by incorporating analogical reasoning to build in *Socratic Assistants*, using principles of Socratic tutoring identified by the Why project [Collins & Stevens, 1982].

Acknowledgments

This research is supported by the National Science Foundation under the REPP program. We thank Tom Hinrichs for many valuable contributions to the design and implementation of VModel. Tara Teich and Rob Harris provided additional programming assistance. Danielle Fisher and Marisa Ketzler assisted with data collection and analysis. We have had the invaluable support of several Chicago Public Schools teachers in developing curricula, and thank them for allowing us to use their classrooms for testing purposes. These teachers are Beverly Miller, Deborah Rogers-Green, Adam Dorr, Carol Scafide, Kiesha Korman, Carlos Rodrigues, and Judy Whitcomb.

References

Biswas, G., Schwartz, D., Bransford, J. & The Teachable Agents Group at Vanderbilt. (2001) Technology Support for Complex Problem Solving: From SAD Environments to AI. In K. Forbus and P. Feltovich (Eds.). *Smart Machines in Education*. AAAI Press/MIT Press, Menlo Park California, USA.

Bouwer, A., Bredeweg, B. (2001). VisiGarp: Graphical representation of qualitative simulation models. In J.D. Moore, G. Luckhardt Redfield, and J.L. Johnson (Eds.), Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future, pages 294-305, IOS-Press/Ohmsha, Osaka, Japan.

Carney, K, Ureel, L., Forbus, K., Fisher, D., Sherin, B., (in prep.) Design of computational and classroom-based supports for student learning modeling as a structured representation task

Collins, A., & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. In R. Glaser (Ed.), *Advances in instructional psychology* (Vol. 2, pp. 65-119). Hillsdale, NJ: Erlbaum.

Davis, J, Leelawong, K, Belynne, K., Bodenheimer, B, Biswas, G., Vye, N., Bransfore, J. Intelligent User Inerface Design for Teachable Agent Systems. (2003) International Conference on Intelligent User Interfaces, pp. 26-34, Miami, FL, January 2003.

De Kleer, J. de & Brown, J.S. (1984) A qualitative physics based on confluences. *Artificial Intelligence*, volume 24, number 1-3, pages 7-83.

Falkenhainer, B. and Forbus, K. (1991) “Compositional Modeling: Finding the Right Model for the Job”, Artificial Intelligence, 51 (1-3), October, 1991.

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

Forbus, K. (2001). Exploring analogy in the large. In Gentner, D., Holyoak, K. and Kokinov, B. (Eds) *The Analogical Mind: Perspectives from Cognitive Science.* Cambridge, MA: MIT Press.

Forbus, K. and Stevens, A. (1981) Using Qualitative Simulation to Generate Explanations. *Proceedings of the Third Annual Conference of the Cognitive Science Society*, pp 219-221, August 1981

Forrester, Jay W. (1996) System Dynamics and K – 12 Teachers. Lecture: University of Virginia School of Education, May 30, 1996

Goddijn, F., Bouwer, A. and Bredeweg, B. (2003). Automatically generating tutoring questions for qualitative simulations. *Proceedings of the 17th International Workshop on Qualitative Reasoning*, Brasilia, Brazil, August.

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.

Kuipers, B. (1994). *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA.

Lester, J. et al., (1997) "The persona effect: Affective impact of animated pedagogical agents," in Proc. Human Factors Comput. Syst., Mar. 1997, pp. 359--366.

Machado, V. and Bredeweg, B. (2003). Building qualitative models with HOMER: A study in Usability and Support. *Proceedings of the 17th International Workshop on Qualitative Reasoning*, Brasilia, Brazil, August.

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

Schwarz, J. and Sherin, B. 2002. Describing Science Content: Bridging the gap between content and process. *Proceedings of ICLS-2002.*

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.

White, B. (1993) Intermediate Causal Models: A missing Link for Successful Science Education? In R. Glaser Ed. Advances in Instructional Psychology New Jersey, Lawrence Earlbaum Associates

Zhang, BaoHui, Wu, Hsin-Kai, Fretz, Eric, Krjick, Joseph, Marx, Ron, Davis, Elixabeth. (2002) Comparison of modeling practices of experts and novice learers using a dynamic, learner-centered tool. Presented at Narst, 2002, Strand7