AI Magazine Volume 24 Number 4 (2003) (© AAAI)

Qualitative Modeling in Education

Bert Bredeweg and Ken Forbus

■ We argue that qualitative modeling provides a valuable way for students to learn. Two model-building environments, VModel and Homer/-VisiGarp, are presented that support learners by constructing conceptual models of systems and their behavior using qualitative formalisms. Both environments use diagrammatic representations to facilitate knowledge articulation. Preliminary evaluations in educational settings provide support for the hypothesis that qualitative modeling tools can be valuable aids for learning.

Modeling is a central skill in scientific reasoning and provides a way of articulating knowledge. Learning to formulate, test, and revise models is a crucial aspect of understanding science and is critical to helping students become active, lifelong learners. Supporting students in articulating models of a domain and refining them through experience, reflection, and discussion with peers and teachers can lead to deeper, systematic understanding of science (for example, Reif and Larkin [1991]; Collins [1996]). However, modeling formalisms have traditionally been associated with creating mathematical models and deriving numeric results. Such approaches fail to capture many crucial aspects of models, such as the conditions under which a model is applicable, and are relatively inaccessible to younger children, such as middle school students. In contrast, qualitative reasoning formalisms provide ontological primitives capable of capturing a conceptual analysis of system behavior, including notions such as causality (de Kleer and Brown 1984; Forbus 1984). Recently, qualitative model-building environments have been proposed that allow learners to articulate knowledge using graphic representations of these intuitive notions (Bessa Machado and Bredeweg, 2002, 2001; Biswas et al. 2001; Forbus et al. 2001). In this article, we further explore this approach. We first review the use of qualitative models and simulations for educational purposes from a wider perspective. We then present specific implementations and uses of qualitative model-building environments.

Qualitative Reasoning and Education

Qualitative modeling is a valuable technology for education for two reasons. First, much of education is concerned with conceptual knowledge. For example, most of what is taught in science education in elementary, middle, and high school consists of causal theories of physical phenomena: What happens, when does it happen, what affects it, and what does it affect? Consider, for example, what is taught about the circulatory system. The anatomy of the heart, lungs, veins, capillaries, and so on, are explained, along with the behavior of the system and the contributions its behaviors make to the successful life of the organism. Traditional mathematical and computer modeling languages do not attempt to formalize such notions because they are designed for expert humans who already know such things. However, uncovering how we think about physical entities and processes is one of the central scientific goals of qualitative physics. Progress in qualitative physics has led to new modeling languages that describe entities and processes in conceptual terms, embody natural notions of causality, and express knowledge about the modeling process itself (cf. Falkenhainer and Forbus [1991]; Forbus [1996]; Rickel and Porter [1997]; Weld and de Kleer [1990]). These languages provide new capabilities for science education software. By embedding humanlike models of entities and processes in the software, the software's understanding can be used to provide explanations that are directly coupled to how specific results were derived.

These explanations can delve into topics that traditional software cannot handle, for example, why a process was thought to occur or why a specific approximation makes sense.

The second reason that qualitative modeling is valuable for education is that it provides the necessary grounding and framework for quantitative and traditional mathematical models (Elio and Sharf 1990; Frederiksen and White 2002; Mettes and Roossink 1981; Ploetzner and Spada 1998). Rare is the instructor who does not lament that students memorize formulas without understanding basic principles. Indeed, cognitive scientists have extensively documented the existence of persistent misconceptions that survive college training in domains such as physics (cf. Gentner and Stevens [1983]). Certainly, a strong sense of the quantitative and the ability to use mathematical models are hallmarks of expertise in many domains. However, the principles governing a domain (that is, the laws, mechanisms, and causal relationships) need to be mastered at the qualitative level to provide the foundation for the deep, robust understanding that is the goal of education. In many technical fields, the expositions are organized around mathematic equations and prooflike derivations of them. Although the equations are important, only a small subset of students gains the desired understanding from this method of presentation. An alternative is to first focus on teaching qualitative principles directly, and some good textbooks attempt to do this. Unfortunately, even the best textbooks tend to shortchange qualitative understanding because of the lack of a systematic, formal vocabulary for it. Qualitative modeling provides such vocabularies, which we hope (as they become more widespread) educators will be able to use to express aspects of their expertise that are currently described as "intuition" or "art."

Survey of Recent Applications

There have been several surveys of qualitative reasoning and education (for example, Forbus [1996]; Bredeweg and Winkels [1998]). Here we mention only recent work. An example of how natural qualitative modeling can be for students comes from the Teachable Agents project at Vanderbilt University (Biswas et al. 2001). Their BETTY'S BRAIN system uses qualitative representations expressed in concept maps to foster learning. The task they use—"teaching" BETTY (their software) by building concept maps so that BETTY can produce explanations—is inspiring. Their qualitative modeling framework uses qualitative mathematics, with tables for composing discrete values to provide qualitative simulation. This system turned out to be intensely motivating for students.

D'Souza et al. (2001) report on ALI, a tool that uses qualitative representations to coach students while they interact with a virtual laboratory. ALI is based on the qualitative process theory (Forbus 1984) and uses visual representations of direct influences (I+/I–) and indirect influences ($\propto_{Q+}/\propto_{Q-}$). Course authors provide ALI the qualitative knowledge relevant to a specific quantitative model. When the quantitative model is simulated, ALI automatically infers the applicable causal dependencies and uses them to interact with the learner, both in terms of asking questions and showing graphics. ALI is claimed to be domain independent so that it can be attached to any quantitative simulation. A pilot study suggests that ALI does provide important guidance during discovery learning.

Syed, Pang, and Sharifuddin (2002) discuss an application of the qualitative process theory in chemistry classes. Students have to mix substances, causing chemical reactions, producing a required end product. If an incorrect mixture of chemicals is taken, the reaction fails, producing undesired products. The students can use a qualitative model alongside their classroom experiment to determine optimal mixtures of substances. The qualitative model shows the end result using color coding and explains why the result is obtained. Students successfully use the qualitative tool to discover what to do in their real chemical experiment.

Researchers have emphasized the importance of multiple models of system behavior, particularly mixing multiple qualitative and quantitative models of the same system that show (slightly) different behaviors based on different assumptions. Typically, the qualitative models are used to create the foundation on which the quantitative models can be understood and explained (for example, Frederiksen and White [2002]; Sime [1998, 1996]; Sime and Leitch [1992]; White and Frederiksen [1990]). The usefulness of this approach is usually proved by showing that learners acquire a better understanding of the phenomena and that they have fewer misconceptions. In a recent study, Sime (2002) takes a different perspective toward evaluating the usefulness of this approach by asking the learners what they thought was useful for them while they worked on problems using multiple models. It turned out that learners easily accepted qualitative models and that they regarded them as worthwhile. However, Sime also points out that generating insightful visualizations of

qualitative models is sometimes difficult.

The idea of using multiple representations is related to using multiple models. They differ because multiple representations can also be made of the same underlying model. This approach is used by van Joolingen and Simone (2001) in their study of collaborative learning. They use a textual representation (in fact, constructing formulas), a qualitative graphic representation, and an *output representation*. The output representation can be used to show the results of simulating models articulated in the other two representations. The general idea is that multiple representations aid learning (for example, Ainsworth [1999]), but in this specific case, the results are preliminary and do not yet allow one to draw strong conclusions.

Finally, Tjaris (2002) reports on an experiment in which learners successfully learn about ecosystems using VisiGarp (Bouwer and Bredeweg 2001) to simulate a complex qualitative model. This result is further discussed in the section on VisiGarp.

VModel: Helping Middle School Students Build Conceptual Models

Early science education is essentially qualitative. Children must learn causal theories: what kinds of things happen, when they happen, and what their consequences are. This early learning provides a solid conceptual foundation for later science education and is also directly useful in dealing with their world. Computers have had less impact on early science learning than in more advanced instruction, in part because most educational science software efforts have relied on computation's traditional strengths in numeric and mathematical modeling, which are too advanced for young learners. Our approach to modeling is to create a student-friendly visual notation for qualitative process theory (Forbus 1984) and create a software environment that helps students express their qualitative, conceptual models as an aid to learning.

Why create another visual modeling language? There are three aspects of modeling that existing languages do not address:

The importance of broadly applicable principles and processes: Existing educational modeling systems treat each modeling task as a new problem, with no connection to other situations. This approach misses the opportunity 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. Existing modeling systems do not help students see the importance of creating a systematic body of knowledge as opposed to a series of ad hoc explanations concerning specific systems.

Understanding when a model is relevant: Knowing when a model is appropriate is a crucial skill. For example, treating plant life as essentially infinite is fine in many predator-prey models but inappropriate when modeling an island or space station. Existing educational modeling systems do not address this issue and thus do not help students connect their models to real-world concerns. For example, public policy debates often rest on the correctness of assumptions underlying competing models (for example, Is global warming really occurring? How much refuge land is needed to preserve biodiversity?).

Qualitative understanding of behavior: Modeling systems tend to be numeric (for example, stella [Grant et al. 1997]), although they sometimes include a qualitative layer on top to simplify model creation (for example, model-it [Jackson et al. 1996]). Understanding how numeric data plots depict behavior is certainly an important skill. However, using these tools requires that students think in terms of detailed mathematic ideas, as well as at the conceptual level, and they must provide significant amounts of numeric data. These requirements can be serious distractions for students who have not yet mastered a phenomenon conceptually.

Qualitative reasoning research has developed theories, representations, and reasoning techniques that enable us to address these issues. Enabling and encouraging students to create their own domain theories, which can be instantiated to model specific situations, should help them understand the broad applicability of scientific principles and processes. The techniques of compositional modeling (Falkenhainer and Forbus 1991) provide the expressive power needed to state modeling assumptions and reason about relevance. Qualitative modeling provides formalisms for expressing intuitive, causal models and the reasoning techniques needed to generate predictions and explanations from them for helping students see the consequences of their ideas. Making these formalisms available through a visual notation is, we believe, the missing piece that will make this power accessible to young students.

Our visual notation is based on concept maps (Novak and Gowan 1984) but with some strong restrictions. Typically, nodes represent

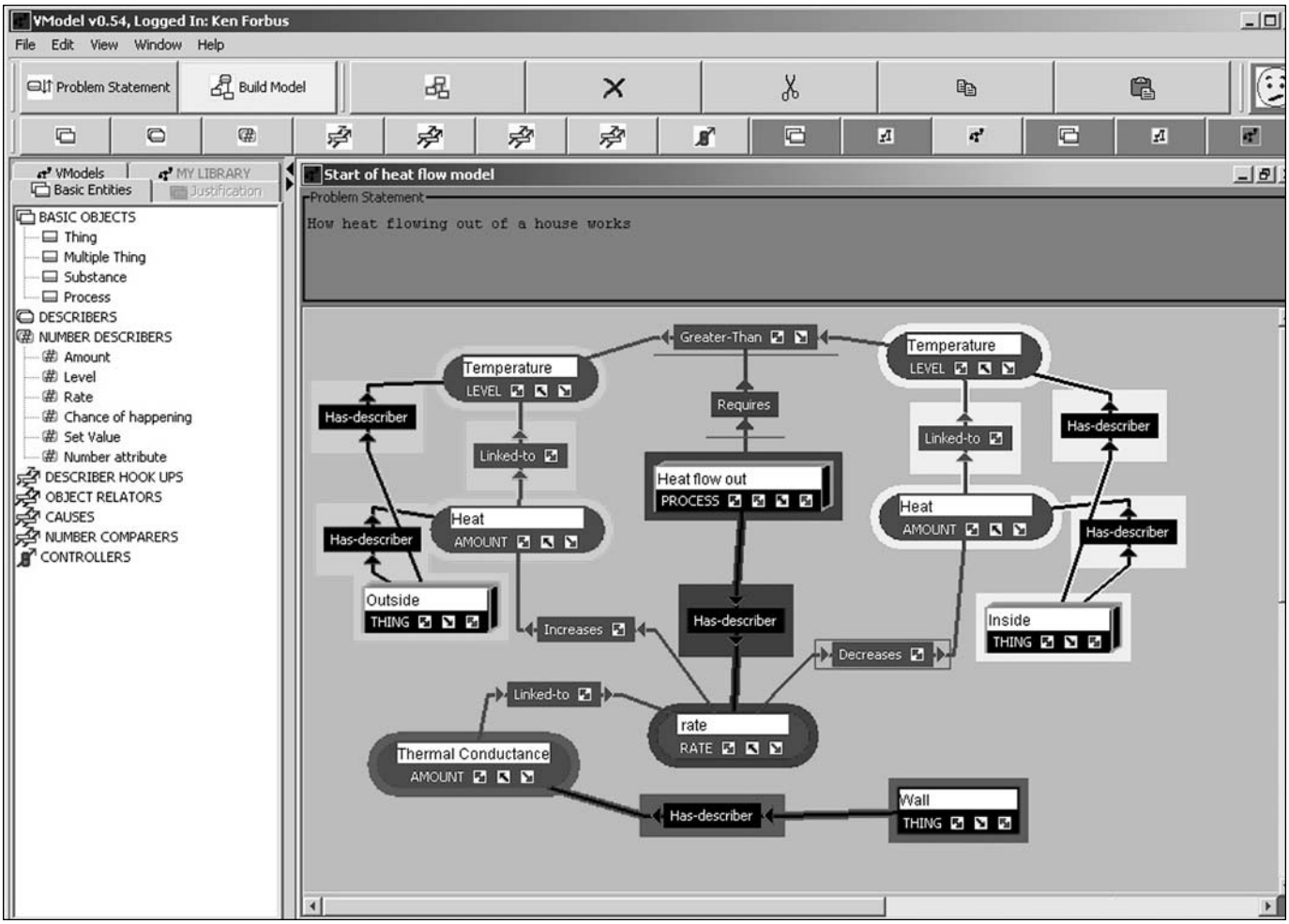

Figure 1. A Learner Using VMODEL.

entities and properties of entities. However, each node has a specified type, such as Thing or Process. These types are drawn from a general ontology provided with the system. This ontology can be extended by students. As usual, links represent relationships. However, the labels that can be used on links are drawn from a fixed set of relationships. This strong restriction provides a clearer semantics than traditional concept maps, which makes software coaching more feasible.

There is a trade-off between providing freedom of expression and scaffolding for students. Providing their own names for entities and properties enables them to express their ideas more accurately (for example, "temperature" versus "hotness" versus "cold"). Requiring students to select a general type for entities and properties helps coaching software figure out which is which and avoids the need to do natural language understanding on their typed phrases. Enabling students to extend the ontology provides them with additional incentive to generalize the concepts in their models. Restricting links to a fixed set of relationships provides a powerful scaffold for students, ensuring that their ideas are at least in the ballpark in terms of form of argument.[1] It also forces students to enter a community of modelers, enabling their ideas to be more easily compared and contrasted with those of others.

Our system, called VMODEL, is illustrated in figure 1. VMODEL provides facilities for students to create and edit models and domain theories. The vocabulary they use to build models is based on qualitative process theory. Situations are described in terms of entities drawn from a simple ontology: thing, multiple thing (for example, the tigers in an ecosystem), substance, or process. Quantities are used to describe their continuous properties. In figure 1, the entities include the Outside and Inside and Wall of a building (shown as boxes). The quantities are displayed as ovals (for example, Heat and Temperature). To help students keep track of which quantity belongs with what entity, a colored

"skin" is placed around an entity and its properties. Thus, in figure 1, it is easy to see that this model consists of four distinct entities.

A small, fixed set of structural relationships (for example, Touches, Contains, Part Of) provides a vocabulary for expressing the circumstances under which processes occur. Ordinal relationships (for example, Greater Than, Less Than, Equal To) can also be used to express information about when processes occur. The Requires relationship is used to link a process to the conditions that enable it.

Causal relationships between parameters are described using the concepts of qualitative process theory, albeit expressed in student-friendly terms. For example, direct influences (I+/I–) are called Increases and Decreases, and qualitative proportionalities ($\propto_{Q+}/\propto_{Q}$) are called Influences and InfluencesOpposite. Although the concepts of functional dependency and integration would be daunting in the abstract for middle school students, we are careful to connect these ideas to their everyday experience using everyday language (for example, When one goes up or down, the other does too) and examples. To aid in brainstorming, we also provide a relationship that expresses a causal connection of some kind, without stating which (Affects). This relationship helps students incrementally refine their ideas.

VModel includes two coaches. The first is the modeling equivalent of spelling and grammar checking, for example, warning a student when he/she tries to connect an influence to an object instead of one of the object's parameters. The second coach uses qualitative simulation to help students see how well their model predicts what they are trying to explain. Students indicate in their model hypotheses about how one or more parameters will change. The changes that their model predicts are determined using qualitative simulation. (In VModel, only within-state behavior is considered, so qualitative reasoning is always efficient.) The student receives three sources of feedback from the simulation: (1) a visual step-by-step animation of the simulation process, (2) an English summary of behavior predicted by the model, and (3) an assessment of how well their model supports their hypotheses. The animation and summary demonstrate how to formally reason with a model. Teachers report that the textual summaries are especially useful because they encourage students to tune their model until it "sounds right."

A basic feature of our design is the use of a model library, which contains all the models they have created. Students can also build their own domain theory by adding entities, attributes, and processes from their models as new building blocks. The model library thus represents their evolving understanding, providing a portfolio and support for reflection.

We have been conducting experiments using VModel with students in the Chicago Public Schools since winter of 2001. Students generally use VModel in groups of two or three students, who later discuss and justify their models in front of the entire class. Although we are still gathering data, some encouraging observations have already been made:

Naturalness of the visual language: After some tuning (for example, using words such as *describer* instead of *attribute*), student questions tend to focus on contents of the models rather than on the use of modeling language primitives.

Generalization in modeling: We are finding that some students do indeed start exploiting abstractions. One student, for example, utilized their model of gazelles and grass to model the interaction between lions and gazelles using the appropriate substitutions, which is not an easy thing for students to grasp. In one classroom discussion, students were working out what to name a model they had created that described an astronaut's weight gain or loss in terms of their caloric intake and their exercise and metabolic needs. Could the model be used to explain more than just astronauts? How could it be made truly general if it weren't already? One girl ventured, "I think it should be called 'the calorie cycle' because you could take out *astronaut* and replace it with *dog,* and this model would explain both." This is the kind of insight that we want them all to attain.

The closest system to VModel is Vanderbilt's betty's brain (Biswas et al. 2001). The VModel qualitative modeling framework is richer, incorporating physical processes and a student-extendable ontology of types of entities. VModel also supports the creation of new abstractions from student models, which the Vanderbilt software does not.

Building and Inspecting Models using Homer and VisiGarp

VModel focuses on using a qualitative vocabulary as the basis for learners to articulate conceptual knowledge. This section discusses an approach that also allows learners to use a qualitative reasoning engine for running and inspecting simulations. The use of qualitative reasoning engines in classrooms is difficult because easy-to-use tools are not available. Building a complete model, and inspecting its simulation results, often requires programming

skills in Lisp or Prolog. The goal of the research discussed here is to develop a user-friendly interactive learning environment that provides the full potential of a qualitative reasoning engine to support learning about system behavior (Bouwer, Bessa Machado, and Bredeweg 2002). HOMER (Bessa Machado and Bredeweg 2002; Jellema 2000) and VISIGARP (Bouwer and Bredeweg 2001) are tools that have been developed in this way. They work on top of the domain-independent qualitative reasoning engine GARP (Bredeweg 1992) and use diagrammatic representations (cf. Kulpa 1994) for building and inspecting qualitative models and simulations. HOMER and VISIGARP are fully implemented systems and have been used in experiments to study the model-creation and model-inspection processes of learners.[2] The results of these experiments are being used to further develop and improve the use of qualitative reasoning engines in classrooms.

Model Building with HOMER

HOMER is organized as a set of builders and tools. Builders capture knowledge and use diagrammatic representations for this purpose. Tools are interactive dialogues for modifying the content of builders. However, building qualitative models is a complex task (cf. Schut and Bredeweg 1996), and additional support is probably required before learners can effectively use HOMER as a tool for learning. An experiment was conducted to investigate the model-building problems that learners have when using tools such as HOMER.

The task of building a qualitative model is to create a set of model fragments (stored in a library) and specify one or more *scenarios*. A scenario refers to a structural description of the system. When the simulator is called, it uses the model fragments to predict the behavior of the system defined in the selected scenario. The modeling is successfully completed when for each of the specified scenarios, the simulator generates the intended behavior graph. HOMER is based on a rational task analysis of the model-construction task (Bessa Machado and Bredeweg 2002) and consists of builders for creating building blocks (entity hierarchy, quantities, quantity spaces, and so on) and constructs (model fragments and scenarios). The constructs are assembled from the building blocks. As an example, consider the model-fragment builder shown in figure 2. The model fragment captures knowledge about an "open contained liquid" and holds the entities *liquid* and *container*. A configuration defines that the container contains the liquid, and an attribute definition specifies that the container is open. All quantities are assigned to the entity *liquid* and have a quantity space of two values, zero and plus. The quantities have *corresponding* quantity spaces, which means that they must have the same value from their quantity spaces (all zero or all plus). Furthermore, the Amount increases the Level, and the Level increases the Pressure (specified by dependencies of type *proportionality*). There is a distinction between Conditions and Consequences in model fragments. The conditions (see the model-fragment builder in figure 2) specify the requirements under which the consequences are true (colored blue in the model-fragment builder). The pull-down menu shows the possible manipulations for adding a conditional statement to the model fragment.

HOMER was designed to prevent learners from making syntactically incorrect models. The user interface is, therefore, context sensitive and restricts the possible user actions based on (1) the content and (2) the current selections in the builder the learner is working on. In the builder in figure 2, no model ingredient is selected. Thus, in the case of adding conditions, the only options are adding a new entity, a conditional model fragment, or an assumption. In contrast, for example, assigning a quantity can only be done after selecting the entity to which it must be assigned. As a result, a learner can only perform syntactically correct actions. It might, however, be the case that a particular action has side-effects that the learner is not aware of. For example, deleting an entry from the entity hierarchy requires that occurrences of that entity in model fragments (and scenarios) are also deleted to preserve the correctness of the model. Notice that this feature is recursive because model ingredients connected to this entity (for example, a quantity) must also be deleted (and so on). HOMER, therefore, investigates each user action with respect to such side-effects, notifies the learner about it, and gives the learner the option to either carry on with the action as planned (including the side-effects) or cancel it. As a result, a model made in HOMER is by definition always a syntactically correct model.

An experiment was conducted in which subjects had to construct a simulation model of a U-tube system using HOMER.[3] The subjects received documentation containing the assignment and a short explanation of the screens and icons used in HOMER. Each model-building session was recorded on video, capturing the activity on the computer screen and the verbal expressions uttered by the subject. Subjects were asked to think aloud as much as possible and thus verbally express what they were doing and

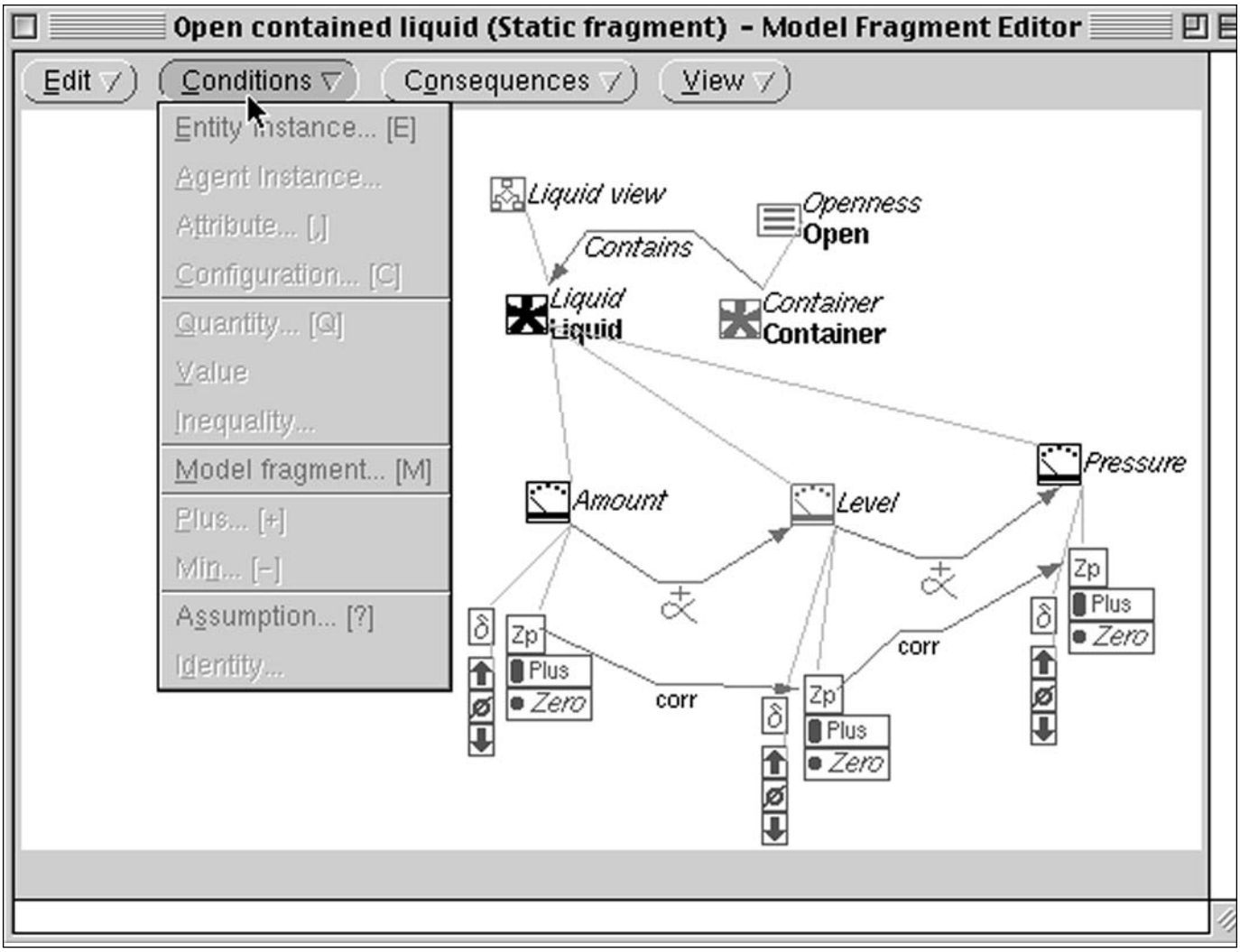

Figure 2. Model-Fragment Builder in HOMER.

the reasons for doing so. The subjects were also encouraged to ask questions during the experiment because questions are a valuable source of information about the problems encountered. A human tutor, monitoring the model-building activity, gave answers to requests for help. Each session lasted one hour. The subjects were four people from a computer science department. Two of them were researchers, and two were Masters students. All four subjects had experience with AI and, thus, with issues concerning knowledge representation. However, they had not built qualitative models before.

Three subjects were able to complete the assignment satisfactorily. They constructed two model fragments, one for *contained liquid* and one for *liquid flow*; one scenario; and the model ingredients needed to actually fill these three constructs. The fourth subject also came far but did not complete the task of creating a scenario within the available time. Without a scenario, it is not possible to run a simulation. From the participants who successfully completed their assignments, two of them actually succeeded in simulating their models using VISIGARP. That is, their models produced behavior (a graph of qualitatively distinct behavior states) when simulated. That subjects were able to produce such a result within an hour is encouraging because the construction of a full qualitative simulation is a complex task.

The results of the experiment can be analyzed from two perspectives: (1) problems caused by poor use of the tool and (2) problems caused by subjects not (fully) understanding how to perform a task. The first can be analyzed and repaired in new implementations of the tool. The second, referred to as model-building problems, require augmentation of the modeling tool with online help and other

interactive means to support the learner. The usability of HOMER was assessed using the heuristic evaluation method (Nielsen 1994). The analysis showed that HOMER is usable. That is, even though some usability factors were violated in the realization of HOMER, it did not prevent subjects from building complex models (Bessa Machado and Bredeweg 2002). Here we discuss some of the model-building problems subjects encountered. A model-building problem was scored when a subject clearly showed an uncertainty concerning the creation of a model ingredient. These problems were clustered into four categories.

Model scope: This problem involves determining which features of the real-world system to include in the model. It might, for example, focus on finding the relevant quantities of the system. For example, one subject worried about which quantities to use and to which entities they belong: "I wonder if I need to create (entity) liquid and if I need *Level* to be a property of a container; I don't know if I need *flow* in this model fragment.... I want to represent (quantity) *Pressure-Difference* as the difference between the two levels. I am confused. Do I need to create a new quantity?"

Model structure: This problem involves determining what to put where in the model, for example, the issue of deciding on the type and number of model fragments needed. The notion of reuse is important in this respect because it provides guidelines for thinking about how to structure the model. For example, it is possible to capture all the details of the U-tube system in a single model fragment. However, such a model cannot be used for reasoning about the behavior of containers, substances, and flows in general.

Model-building concepts: This problem involves understanding the model-building concepts as provided by the tool, for example, the difference between attributes and quantities, the meaning of an influence, or the difference between generic and instance knowledge. Model builders need to understand the qualitative ontology as it is made available by the tool. To become modelers, they have to acquire knowledge concerning what concepts are available and how they can be applied to articulate knowledge about system behavior. A protocol excerpt illustrates a subject's "mental struggle" in this respect when the modeler looks at the list of possible dependencies that can be defined: "The flow is directly proportional to the Pressure difference … I guess so … I want to have … proportionality … no … inequality … equal … not really equal but … qualitatively equal I guess...."

Model representations: This problem is related to the model-building concepts category, but now it refers to the actual representation of an idea using the model-building ontology. The subject wants to articulate something, knows the concept, but does not know how to actually formulate that with the options provided by the environment. One subject remarked (after selecting the three quantities: Level for both containers and Pressure Difference), "Now I expect proportionality. I mean the pressure difference is proportional to the level difference."[4] Thus, the subject "knows" what to represent but does not know "how" to represent it.

Our current research focuses on the development of a set of interactive software agents that are knowledgeable about the status of the model being built and, on the basis of this development, provide help. The experiment discussed earlier provides a focus on the type of help that must be given by these agents.

Inspecting Simulations Using VisiGarp

Learners can further benefit from modeling when they can also run simulations using their models. VisiGarp (Bouwer and Bredeweg 2001) is designed for this purpose. It provides a graphic user interface for running and inspecting qualitative simulations. These simulations might use models constructed by teachers or domain experts but might also use models created by learners. Figure 3 shows the main window of VisiGarp from which the simulation is controlled, and viewers can be opened for inspection. The state graph in the main control window shows a simulation of a U-tube for which three end states have been found: 2 (possibly using 1 or 3), 6 (using 1 and 5), and 7 (using 3 and 4). In the bottom of this window, a partial screen dump of the quantity value history is displayed. It shows, for example, that in state 4, the Level of Liquid2 is plus and decreasing, whereas the Level for Liquid1 is fully filled (max) and still increasing. Fully filled and still increasing refers to a full container spilling water.

The content of the dependency screen for state 2 is shown in figure 4. It shows the quantities for each entity, their quantity spaces, and values they currently have. The black triangles denoting the current value also depict the direction of change. The dependencies show the causal relationships between the quantities. The Flow Rate, by means of an influence, causes the Amount of Liquid1 to increase (*I*+) and the Amount of Liquid2 to decrease (*I*–). For both liquids, this effect is propagated by a proportionality (*P*+), causing Level and Bottom-

Pressure to change in the same direction as Amount. Figure 4 also shows that the difference between the Bottom-Pressures is becoming smaller.

The visual design of VisiGarp differs from homer. A noticeable difference concerns the causal model in figure 4 and figure 2, respectively. This difference is the result of the different tasks that the two tools support.

First, in VisiGarp, the distinction between Conditions and Consequences can be ignored because the learner is looking at a full simulation of the system. All facts are integrated.

Second, the user interface only needs to facilitate the movement of ingredients on the screen and not selections followed by model-building actions whose results need to be shown on the screen.

Third, the inspection task has to deal with an integration of model ingredients (as generated by a simulator). Thus, an inspection tool has to provide easy access and create understandable overviews of the (usually large set of) facts derived by the simulator.

One of the key concepts used in the design of VisiGarp is the notion of a **visual container** (Dreyfuss 1972; Horton 1994), using the entities as containers. Thus, all model ingredients belonging to a particular entity are grouped into a single area (for example, in figure 4, the quantities belonging to Liquid1). Ingredients relating aspects from different entities (often dependencies) cross the boarders of these bounding boxes (for example, in figure 4, the Flow-Rate of Fluid-Path1 influencing Amount of Liquid1). An alternative could have been to use the notion of model fragment as the basis for the visual container (as done in homer). An argument in favor of this approach is that certain conceptual notions, such as processes, are more apparent in the interface. However, when running a simulation, many model fragments can apply, and all of them can, in principle, refer to the same set of model ingredients. The lines relating model ingredients present in different model fragments might then significantly clutter the screen, making the diagrams difficult to read.

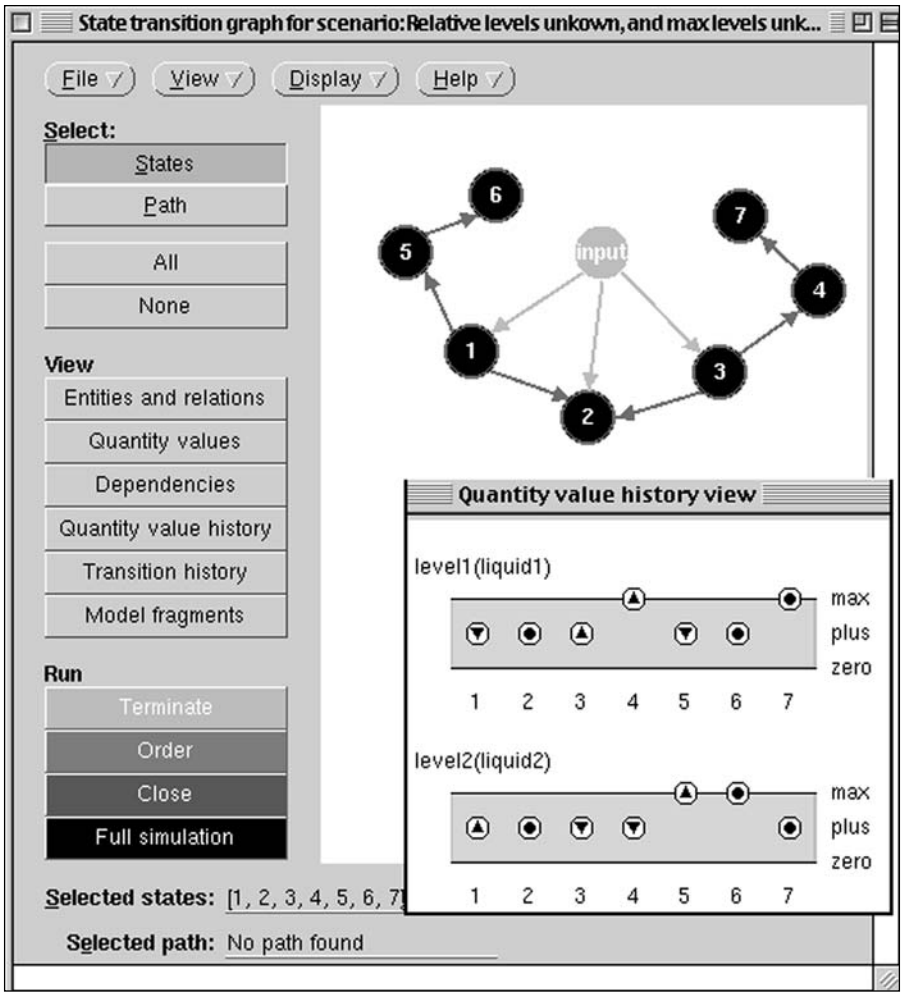

Figure 3. VisiGarp Simulation Control and (Part-of) Quantity Value History.

An experiment was conducted testing the usability of VisiGarp in a learning situation using 30 first-year university students (Tjaris 2002). The subject matter was a simulation using the Cerrado succession hypothesis (CSH) model (Salles and Bredeweg 1997). This is a rather large model (see also this special issue) about population dynamics and community behavior. As psychology students, the subjects had no significant knowledge of qualitative reasoning and AI. The experiment took one subject at the time and started with two questionnaires testing the subject's preknowledge of the CSH domain and the meaning of the icon language used in VisiGarp. Next, the subject had to answer prediction exercises using simulations produced by VisiGarp. Two scenarios were selected for this purpose, one concerning the behavior of a single population and one concerning the full CSH model. After

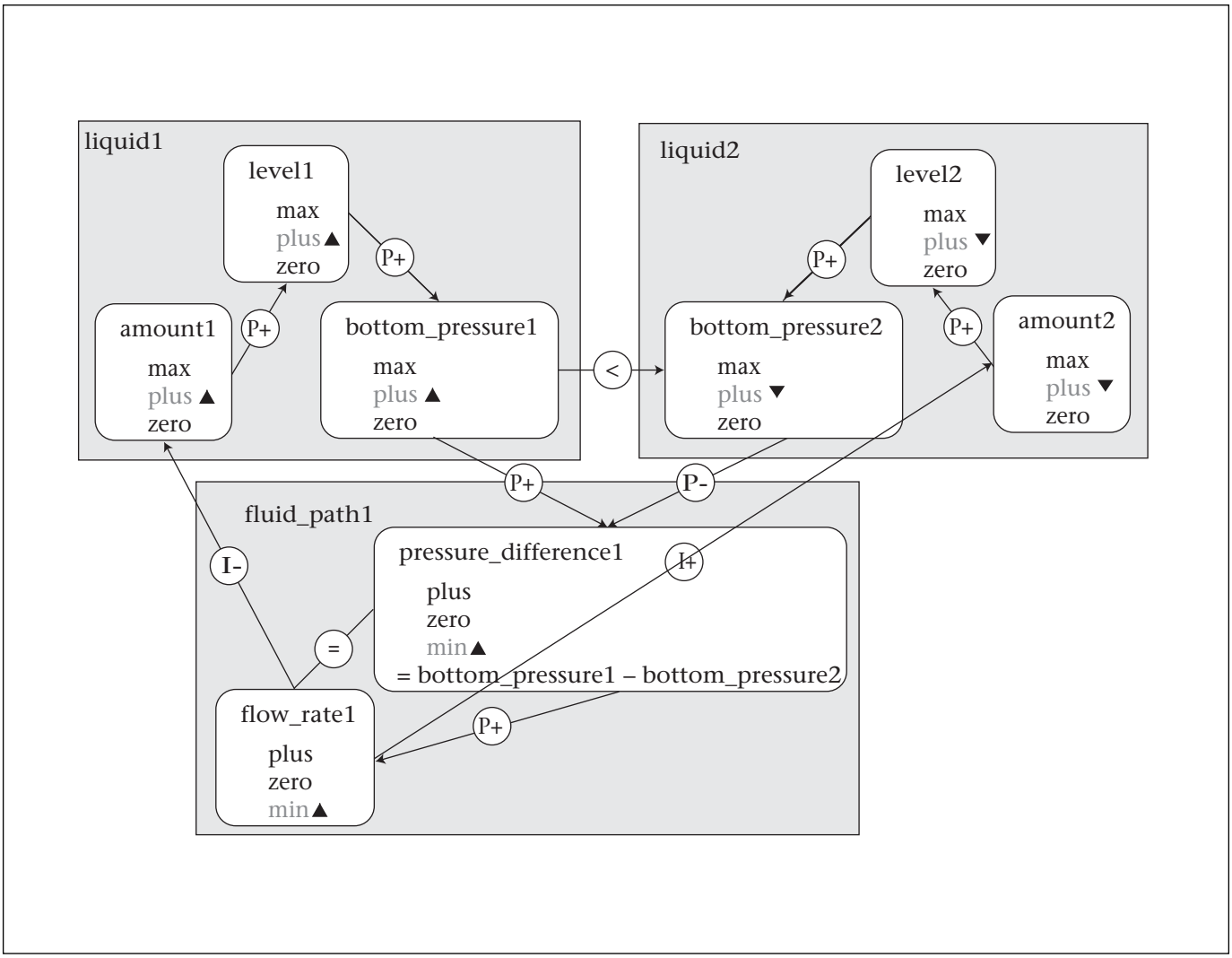

Figure 4. VisiGarp Showing Dependencies.

this treatment, the subject's domain knowledge and knowledge of the icon language was tested again using a third and a fourth questionnaire. Finally, the subject was asked to evaluate VisiGarp, both using an attitude questionnaire and having a discussion with the experiment leader. Each session lasted two hours and was taped on video for later analysis. For most subjects, two hours was just enough, but a few had to be instructed to stop running the simulations and proceed with the third questionnaire.

Subjects performed better on the posttest measuring the domain knowledge (M = 0.67 with SD = 0.47) than on the pretest (M = 0.42 with SD = 0.49). The difference was significant ($p < 0.001$). We did not expect a learning effect and, thus, did not include a control group for two reasons. First, we expected that the amount of knowledge to be learned using the CSH model was too large to be dealt with in less than two hours. Second, we were interested in usability, hence the video registration for further analysis of user behavior. The significant learning effect came as a pleasant surprise. It supports the hypothesis that qualitative simulations can support learners in effectively acquiring domain knowledge. The second questionnaire, testing the icon language used in VisiGarp, did not show a significant distinction ($p = 0.06$) between pretest ($M = 0.67$ with $SD = 0.47$) and posttest ($M = 0.73$ with $SD = 0.44$). Further analyzing the answers to the questionnaires suggests that most icons are easy to understand or learn (for example, ≤, <, =, >, ≥, *P*+, *P*–, *I*+, and *I*–) but that a few icons are not clear from the start and are more difficult to learn (for example, *Q*^, *Q*, *V*^, and *V*).[5] Moreover, solving the

prediction exercises during the treatment sessions for the most part did not require a detailed processing of these more difficult icons, which might explain why the meaning of these icons was not learned fully. Analyzing the videotapes showed that the automated layout of VisiGarp did not always produce insightful diagrams, particularly for the complex CSH model. The subjects also mentioned this problem during the closing discussion. Optimizing the layout and optimizing the means to automatically summarize the output of complex qualitative simulations are issues of our current research (for example, Bouwer and Bredeweg [2002]). Finally, on the five-point attitude questionnaire, subjects evaluated the usefulness and usability of the VisiGarp software quite positively (M = 3.70 with SD = 1.14). Thus, we are actually realizing our goal of creating a user-friendly interactive learning environment for dealing with qualitative knowledge.

Conclusion and Discussion

This article emphasizes the importance of conceptual knowledge in education, particularly concerning reasoning about system behavior. It argues that qualitative formalisms and reasoning engines provide the means necessary to support learners in developing such conceptual models. Two interactive environments for building qualitative models are discussed, VModel and the homer/VisiGarp combination. Both support learners in articulating their knowledge and thus foster learning as a constructive process. Special attention is given to the use of graphics and knowledge visualization to create learner-friendly tools. As these modeling tools, and others like them, evolve, our hope is that they will help students become full-fledged modelers, engaged in the joy of unraveling complex phenomena rather than frustrated by memorizing mountains of isolated facts. By keeping the entry barriers for use as low as possible, we hope to create tools that will be to modelers what word processors are to writers and spreadsheets are to accountants.

Acknowledgments

The VModel project is supported by the National Science Foundation under the REPP program. The members of the VModel team are Ken Forbus (co–principal investigator), Bruce Sherin (co-principal investigator), Leo Ureel (lead programmer), and Karen Carney (education). We have had the invaluable support of several Chicago Public School 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. The development and evaluation of VisiGarp is part of a Ph.D. project carried out by Anders Bouwer and a Masters project carried out by Peggy Tjaris. The development and evaluation of homer is part of a Ph.D. project carried out by Vania Bessa Machado and a Masters project carried out by Jelmer Jellema.

Notes

1. For example, "Pixies do it" and other anthropomorphic arguments are simply not expressible.

2. The software can be downloaded from www.swi.psy.uva.nl/projects/GARP/.

3. Additional trails have been undertaken but not in an experimental setting.

4. To express this notion, two proportionalities must be created, one from each Level to the Pressure difference. The proportionalities should be in opposite directions. A more articulate model would also include Pressure at the bottom of each container, as shown in figure 4.

5. For details on direct and indirect quantity correspondence and direct and indirect value correspondence, see Bredeweg (1992).

References

Ainsworth, S. 1999. The Functions of Multiple Representations. *Computers and Education* 33(2–3): 131–152.

Bessa Machado, V., and Bredeweg, B. 2002. Investigating the Model-Building Process with homer. Paper presented at Model-Based Systems and Qualitative Reasoning for Intelligent Tutoring Systems, International Workshop at ITS2002, 2–7 June, San Sebastian, Spain.

Bessa Machado, V., and Bredeweg, B. 2001. Towards Interactive Tools for Constructing Qualitative Simulations. Paper presented at the Fifteenth International Workshop on Qualitative Reasoning, 17–19 May, San Antonio, Texas.

Biswas, G.; Schwartz, D.; Bransford, J.; and The Teachable Agents Group at Vanderbilt. 2001. Technology Support for Complex Problem Solving: From SAD Environments to AI. In *Smart Machines in Education*, eds. K. Forbus and P. Feltovich, 71–97. Menlo Park Calif.: AAAI Press.

Bouwer, A., and Bredeweg, B. 2002. Aggregation of Qualitative Simulations for Explanation. Towards Interactive Tools for Constructing Qualitative Simulations. Paper presented at the Sixteenth International Workshop on Qualitative Reasoning, 10–12 June, Barcelona, Spain.

Bouwer, A., and Bredeweg, B. 2001. VisiGarp: Graphical Representation of Qualitative Simulation Models. In *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future,* eds. J. D. Moore, G. Luckhardt Redfield, and J. L. Johnson, 294–305. Amsterdam, The Netherlands: IOS.

Bouwer, A.; Bessa Machado, V.; and Bredeweg, B. 2002. Interactive Model-Building Environments. In *The Role of Communication in Learning to Model,* eds. P. Brna, M. Baker, K. Stenning, and A. Tiberghein, 155–182. Mahwah, N.J.: Lawrence Erlbaum.

Bredeweg, B. 1992. Expertise in Qualitative Prediction of Behavior. Ph.D. thesis, University of Amsterdam.

Bredeweg, B., and Winkels, R. 1998. Qualitative Models in Interactive Learning Environments: An Introduction. *Interactive Learning Environments* 5(1–2): 1–18.

Collins, A. 1996. Design Issues for Learning Environments. In *International Perspectives on the Design of Technology-Supported Learning Environments,* eds. S. Vosniadou, E. D. Corte, R. Glaser, and H. Mandl, 347–362. Mahwah, N.J.: Lawrence Erlbaum.

de Kleer, J., and Brown, J. S. 1984. A Qualitative Physics Based on Confluences. *Artificial Intelligence* 24(1–3): 7–83.

Dreyfuss, H. 1972. *Symbol Source Book*. New York: McGraw-Hill.

D'Souza, A.; Rickel, J.; Herreros, B.; and Johnson, W. L. 2001. An Automated Lab Instructor for Simulated Science Experiments. In *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future,* eds. J. D. Moore, G. Luckhardt Redfield, and J. L. Johnson, 65–76. Amsterdam, The Netherlands: IOS.

Elio, R., and Sharf, P. B. 1990. Modeling Novice-to-Expert Shifts in Problem Solving and Knowledge Organization. *Cognitive Science* 14(4): 579–639.

Falkenhainer, B., and Forbus, K. 1991. Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence* 51(1–3): 95–143.

Forbus, K. D. 1996. Qualitative Reasoning. In *The Computer Science and Engineering Handbook,* ed. A. B. Tucker, 715–733. Boca Raton, Fla.: CRC.

Forbus, K. D. 1984. Qualitative Process Theory. *Artificial Intelligence* 24(1–3): 85–168.

Forbus, K. D.; Carney, K.; Harris, R.; and Sherin, B. L. 2001. A Qualitative Modeling Environment for Middle-School Students: A Progress Report. Paper presented at the Fifteenth International Workshop on Qualitative Reasoning, 17–19 May, San Antonio, Texas.

Frederiksen, J. R., and White, B. Y. 2002. Conceptualizing and Constructing Linked Models: Creating Coherence in Complex Knowledge Systems. In *The Role of Communication in Learning to Model,* eds. P. Brna, M. Baker, K. Stenning, and A. Tiberghein, 69–96. Mahwah, N.J.: Lawrence Erlbaum.

Gentner, D., and Stevens, A. L. 1983. *Mental Models*. Mahwah, N.J.: Lawrence Erlbaum.

Grant, W. E.; Pedersen, E. K.; and Marin, S. L. 1997. *Ecology and Natural Resource Management: Systems Analysis and Simulation*. New York: Wiley.

Horton, W. K. 1994. *The Icon Book: Visual Symbols for Computer Systems and Documentation*. New York: Wiley.

Jackson, S.; Stratford, S. J.; Krajcik, J. S.; and Soloway, E. 1996. Making System Dynamics Modeling Accessible to Pre-College Science Students. *Interactive Learning Environments* 4(3): 233–257.

Jellema, J. 2000. Ontwerpen voor ondersteuning—De rol van taakkennis bij ondersteuningsontwerp (Designing for Support—The Role of Task Knowledge for Designing Support). Masters thesis, University of Amsterdam.

Kulpa, Z. 1994. Diagrammatic Representation and Reasoning. *Machine Graphics and Vision* 3(1–2): 77–103.

Mettes, C. T., and Roossink, H. J. 1981. Linking Factual and Procedural Knowledge in Solving Science Problems: A Case Study in a Thermodynamics Course. *Instructional Science* 10:333–361.

Nielsen, J. 1994. *Usability Engineering*. San Francisco, Calif.: Morgan Kaufmann.

Novak, J. D., and Gowin, D. B. 1984. *Learning How to Learn*. New York: Cambridge University Press.

Ploetzner, R., and Spada, H. 1998. Constructing Quantitative Problem Representations on the Basis of Qualitative Reasoning, *Interactive Learning Environments* 5(1–2): 95–108.

Reif, F., and Larkin, J. H. 1991. Cognition in Scientific and Everyday Domains: Comparison and Learning Implications. *Journal of Research in Science Teaching* 28(9): 733–760.

Rickel, J., and Porter, B. 1997. Automated Modeling of Complex Systems to Answer Prediction Questions. *Artificial Intelligence* 93(1–2): 201–260.

Salles, P., and Bredeweg, B. 1997. Building Qualitative Models in Ecology. Paper presented at the Eleventh International Workshop on Qualitative Reasoning, 3–6 June, Cortona, Tuscany, Italy.

Schut, C., and Bredeweg, B. 1996. An Overview of Approaches to Qualitative Model Construction. *The Knowledge Engineering Review* 11(1): 1–25.

Sime, J. A. 2002. Learning with Qualitative Models and Cognitive Support Tools: The Learners' Experiences. Paper presented at Model-Based Systems and Qualitative Reasoning for Intelligent Tutoring Systems, International Workshop at ITS 2002, 2–7 June, San Sebastian, Spain.

Sime, J. A. 1998. Model Switching in a Learning Environment Based on Multiple Models. *Interactive Learning Environments* 5(1–2): 109–124.

Sime, J. A. 1996. An Investigation into Teaching and Assessment of Qualitative Knowledge in Engineering. Paper presented at the 1996 European Conference on Artificial Intelligence in Education, 30 September–2 October, Lisbon, Portugal.

Sime, J. A., and Leitch, R. R. 1992. Multiple Models in Intelligent Training. Paper presented at the First International Conference on Intelligent Systems Engineering (ISE '92), 19–21 August, Edinburgh, Scotland.

Syed, S. M.; Pang, J. S.; and Sharifuddin, Z. 2002. Application of Qualitative Process Theory to Qualitative Simulation and Analysis of Inorganic Chemical Reaction. Paper presented at the Sixteenth International Workshop on Qualitative Reasoning, 10–12 June, Barcelona, Spain.

Tjaris, P. 2002. Kwalitatieve simulaties als midel tot kennisoverdracht (Qualitative Simulations as a Means to Knowledge Transfer). Master's thesis, University of Amsterdam.

van Joolingen, J., and Simone, S. 2001. Representations in Collaborative Modeling Tasks. Paper presented at External Representations in AIED: Multiple Forms and Multiple Roles, Workshop at AIED 2001, 19–23 May, San Antonio, Texas.

Weld, D., and deKleer, J., eds. 1990. *Readings in Qualitative Reasoning about Physical Systems*. San Francisco, Calif.: Morgan Kaufmann.

White, B., and Frederiksen, J. R. 1990. Causal Model Progressions as a Foundation for Intelligent Learning Environments. *Artificial Intelligence* 42(1): 99–157.

Bert Bredeweg's research (University of Amsterdam, Department of Social Science Informatics) is driven by fundamental questions about computational intelligence. It includes issues related to interactive learning environments, human-computer interaction, and the use of AI technology to enable communicative interactions between humans and computers. His e-mail address is bert@swi.psy.uva.nl.

Kenneth D. Forbus is a professor of computer science and education at Northwestern University. His research interests include qualitative reasoning, analogy and similarity, sketching and spatial reasoning, cognitive simulation, reasoning system design, articulate educational software, and the use of AI in computer gaming. He received his degrees from the Massachusetts Institute of Technology (Ph.D. in 1984). He is a fellow of the American Association for Artificial Intelligence and an associate editor of *Cognitive Science* and serves on the governing board of the Cognitive Science Society and the editorial boards of *Artificial Intelligence,* AAAI Press, and the *Journal of Game Design*. His e-mail address is forbus@northwestern.edu.