AI-ED 2001 Workshop
External Representations in AIED: Multiple Forms and Multiple Roles
San Antonio, Texas, Sunday 20th May 2001

Toward Glass Box Educational Simulations: Reifying Models for Inspection and Design [1]

TOM MURRAY, LARRY WINSHIP, ROGER BELLIN, MATT CORNELL
Hampshire College, Amherst, MA 01002
tmurray@hampshire.edu, www.cs.umass.edu/!tmurray

Abstract: In this paper we describe the SimForest inquiry learning software and pedagogical principles behind its design. We have implemented both "black box" and "glass box" versions of the software, and focus here on the glass box software, which allows learners to inspect and modify the equations that constitute the simulation's model. We describe SimForest features that support several types of inquiry questions: concrete, relationship, explanatory, and modeling. We also describe our approach to the "simulation meta-model problem"–i.e. that simulation models have both underlying assumptions and emergent properties that are not part of the model itself, but are important topics of instruction. Our work is relevant to the topic of "external representations" in a number of ways. At this stage in our research we are looking for perspicuous external representations for constructs at both the domain and modeling levels. Our software design incorporates preliminary versions of the following features which we hope to improve: methods for viewing and editing equations; graphical ways of showing multiple representation of equation-based relationships; ways to portray and navigate among referential relationships among variables, tables, and equations; representations of underlying assumptions and emergent properties of simulation models. We also hope to add interactive representations of a student's hypotheses and inquiry processes.

DESCRIPTION OF SIMFOREST

Computer simulations have been used as learning environments for a number of reasons. Our focus is on educational simulations that promote inquiry-style learning in two contexts: black box and the glass box. We have developed an educational application called SimForest that simulates tree and forest growth, the succession of tree species over time, and the effects of environmental and man made disturbances on forest growth. It allows students to set various environmental and climate parameters, watch a forest plot grow over time, and analyze the resulting data. The simulation is rich enough and grounded enough in the experiential world of students (trees and forests) that we commonly observe students posing their own questions and engaging in the types of "sustained inquiry" described in [Soloway et al. 1997]. A significant value of simulation-based learning is that students can compete full inquiry cycles in a fraction of the time it would take to do the "wet" version of an experiment, and thus learn better inquiry skills because they can go through more inquiry cycles. Note that we strongly advocate that students experience phenomena in the real world first if possible before simulating it on a computer.

[1] Abridged version of the full length paper.

We have developed two versions of SimForest. The "black box" version, SimForest-B, is a traditional simulation incorporating adjustable parameters, measurement and analysis tools, and process visualization tools (see Figures 1, 2). The objectives of SimForest-B are for students to learn certain scientific concepts, skills, and principles, and to engage in (and be supported in) the various stages of the scientific inquiry process. The "glass box" version, called "SimForest-G," exposes the underlying simulation model (a set of about 25 equations) to the learner (see Figures, 3, 4, 5). This system has most of the features of the black box version, and could be used for the same educational purposes, but it also has model inspection tools and model editing tools.[2] The equation *inspection* tools allow the learner to "look under the hood" of the simulation model and see the equations underlying it. The equation inspector is basically a hypermedia "model information space" within which learners can browse to learn about the meaning and theory behind important parameters, see the connections and interdependencies between model variables, and track the values of variables as the simulation runs. The equation *editor* tools allow the learner to alter the model by changing or adding variables, parameters, and equations. It is a modeling tool and a simplified programming language of sorts.[3] While equation inspection allows learners to learn concepts and principles in the model space, the equation editor allows learners to do inquiry in the model space. They can test their hypothesis by changing parameters that embody basic assumptions and by encoding alternate theories.

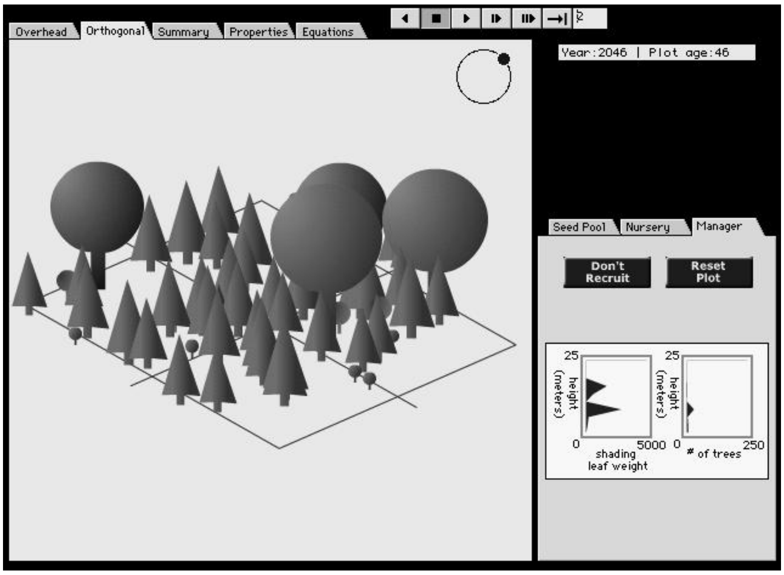


Figure 1. Sim-Forest-B Orthogonal view.

The model inspector and the model editor are not separate modules, they are integrated into one tool set, but in our description we distinguish the two modes of use. The editor includes the common set of algebraic and Boolean operators, a conditional operator, composite operators (sum, SD, average, count), and a random number function. Model variables (stored in tables) can

2 SimForest-B was built in Macromedia Director. SimForest-G is a second generation of the software, built in Java. The Java application has a more flexible and powerful underlying simulation engine with the additional functionality mentioned, but as yet has a less sophisticated interface and is missing some of the fancier visual features of the Director version (such as 3D perspective view of the forest).

3 The current version uses a hierarchical representation of equations (Figure 3). This was the easiest to implement but we do not expect it to be as easy to use as more graphical network-like or WYSIWYG equation editing tools which we hope to include in future versions.

be constants or can depend on the results of equations. By giving the model access to values from the previous simulation iteration we can perform iterative calculus (difference, slope, and integral computations). The representational formalism, and thus the phenomenon that we can model, is limited because we do no support differential equations, the coding of arbitrary iteration constructs (loops or recursion), or function-based model components (i.e. the components are equations that produce a result, not functions that take input parameters and can produce behaviors and side effects).

Current status. To date SimForest-B and an accompanying Microsoft Excel-based analysis package are complete and have undergone four rounds of formative classroom-based evaluation. Analysis of inquiry behavior observed in users will be reported on in an upcoming paper. A draft version of the accompanying curriculum is complete, and will be running a summer teacher's institute based on using SimForest in the classroom in July 2001. We have been studying how a veteran inquiry-based science teacher uses the simulation with students, and we are basing the curriculum design on these observations.[4]

SimForest-G is complete but has yet to undergo rounds of user testing and has yet to be evaluated. This paper focuses on SimForest-G, model inspection and model editing, and related pedagogical design considerations. As the software has yet to be used in realistic contexts, our suggestions are hypothetical and still forming.

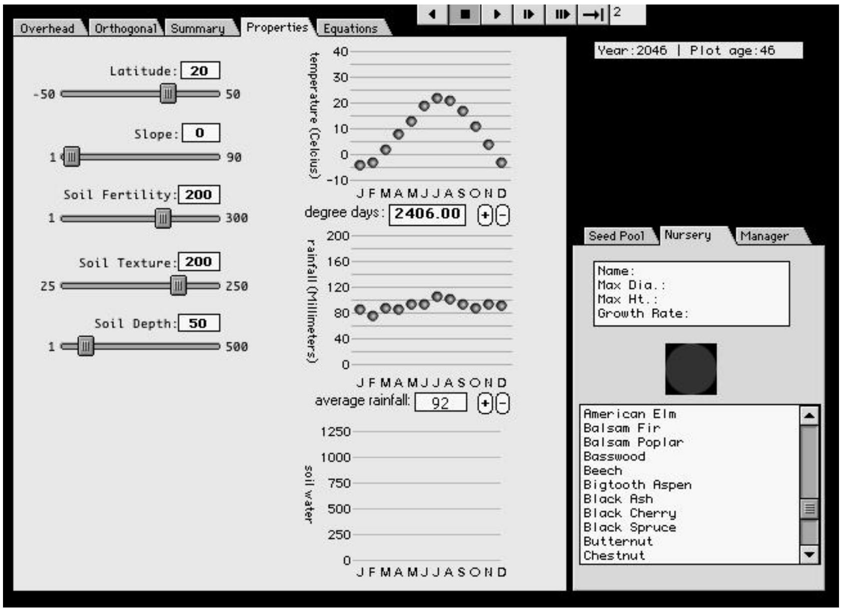


Figure 2. SimForest-B Site Property Editor

[4] An inquiry-based educational philosophy is deeply embedded in Hampshire college's academic structure and undergraduate course offerings, as well as in the faculty's teaching methods.

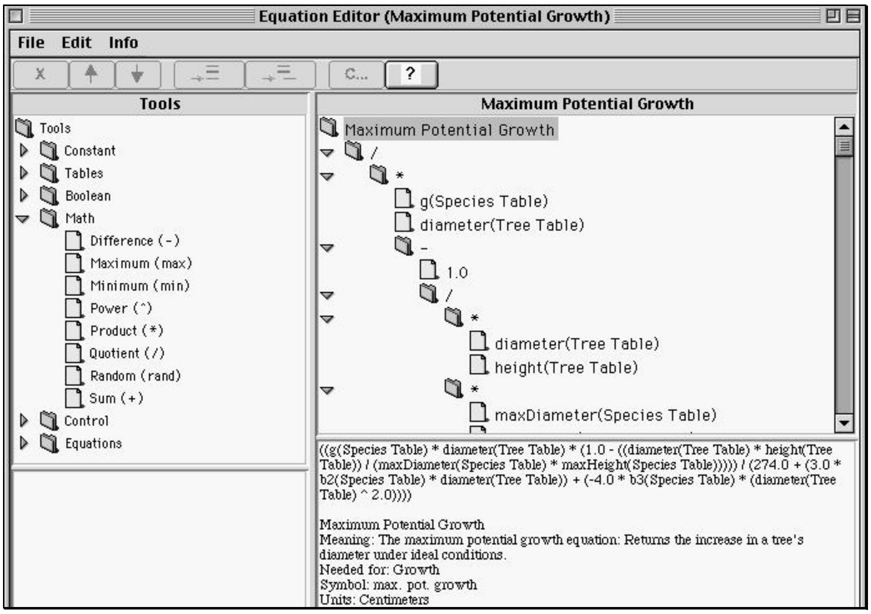
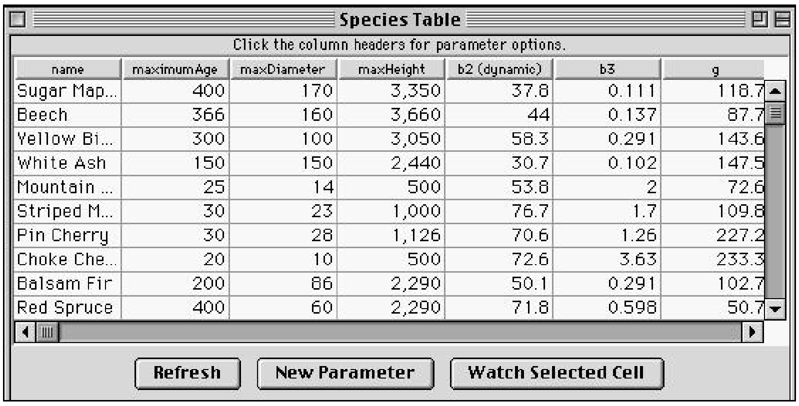


Figure 3: SimForest-G Equation Editor



name	maximumAge	maxDiameter	maxHeight	b2 (dynamic)	b3	g
Sugar Map...	400	170	3,350	37.8	0.111	118.7
Beech	366	160	3,660	44	0.137	87.7
Yellow Bi...	300	100	3,050	58.3	0.291	143.6
White Ash	150	150	2,440	30.7	0.102	147.5
Mountain ...	25	14	500	53.8	2	72.6
Striped M...	30	23	1,000	76.7	1.7	109.8
Pin Cherry	30	28	1,126	70.6	1.26	227.2
Choke Che...	20	10	500	72.6	3.63	233.3
Balsam Fir	200	86	2,290	50.1	0.291	102.7
Red Spruce	400	60	2,290	71.8	0.598	50.7

Figure 4: SimForest-G Species Table

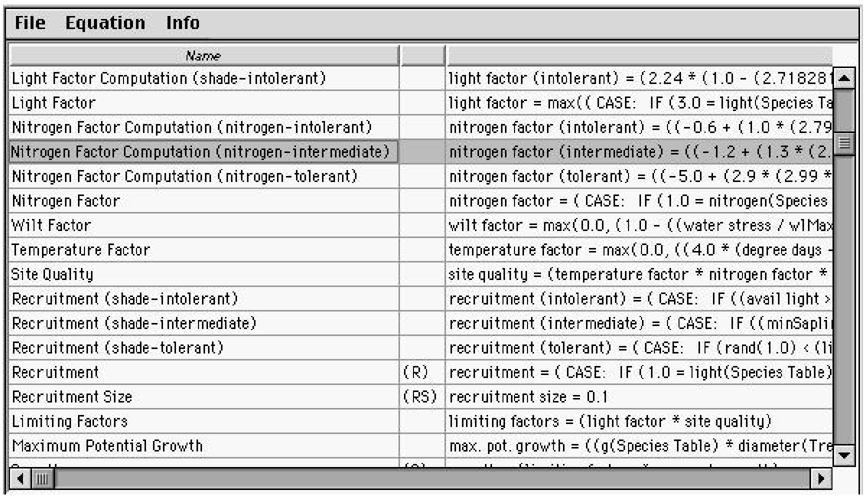


Figure 5: SimForest-G Equation Set Editor.

PEDAGOGICAL DESIGN

Straford [1997 pg. 4] notes that "creating and running dynamics models should help clarify ones own mental models and foster deeper understanding of complex systems." The model inspector tires to "bridge the semantic gap between [users] conceptual model of the problem and the computational model of the [formalism]" by providing a "domain-oriented representation" as is recommended in [Repenning & Summer 1995 p. 17].[5] Using the knowledge based approach to educational simulations gives students multi-modal access to various perspectives for learning about equations and their relationship to observed phenomena. The knowledge-based approach allows us to implement alternative formulations, some more sophisticated than others, and facilitates a sequencing of curricular activities that progresses students toward increasingly sophisticated models, as in the Model Evolution approach [White & Frederiksen 1995].

SimForest draws on methods found in existing simulations used in professional forestry, but adds several features that address pedagogical shortcomings in existing software. The focus of existing forestry software is on the simulation itself, and inadequate attention is paid to pedagogical aspects and to supporting the inquiry process. This is true of most simulation-based educational software, as well as all existing forest simulations. Thus our findings will have implications for all simulation-based science education software. And, as mentioned, the model inspection and editing modules of the SimForest-G software are domain independent, and we plan to test our ideas in additional domains. Existing simulation-based educational software has two almost universal shortcomings, which we will call the "black box problem" and the "meta-model problem," as described below. First we will distinguish several types of inquiry questions that lead to SimForest design features.

Students engaged in inquiry learning about natural phenomena ask a variety of questions and inquiry curriculum encourages them to ask such questions [Collins & Stevens 1993]. We distinguish four categories of inquiry questions that arise using a simulation: "concrete," "relationship," "explanation," and "modeling." Below we describe them in order of increasing intellectual sophistication or abstraction. Later we will show how our software addresses learning for each type of question.

1. Concrete/Situational ("What if?"). Questions that deal with particular observable variables or situations. For example: "what would happen if I started a forest with almost all birches and just two maples?" Concrete questions can lead to open ended trial-and-error "fiddling" with simulation parameters, or can form the basis for more systematic testing of hypothesis. But since learners are interested in specific (not general) questions, less systematicity in investigation is required as compared with the other question types. Questions of the form "why did X happen?" can fall under this category if they are about specific observable variables. Both qualitative and quantitative inquiry can occur at the Concrete level.

2. Relationships ("How?"). Relationship questions focus on the relationship between parameters of the system, and represent more abstract conceptual understating of the domain than the concrete questions. These questions are key to a conceptual understanding of the system as a whole. For example: "How does soil quality affect species diversity?" "What is the relationship between soil nitrogen and leaf size?" Like the concrete questions above, relationship questions can be answered through the inquiry process including data collection and analysis.

3. Explanatory ("Why?"). For example "Why does increased soil quality decrease tree diversity?" These questions delve deeper into the causal relationships and underlying assumptions beneath the model. Unlike relationship questions, explanatory questions (under our definition of the term here) can't be determined by measurement alone in the learning

[5] We do not yet have a visually rich graphical way to portray and interact with the components of our model, as in the AgentSheets, Model-It, SimQuest systems.

environment. Students must hypothesize underlying principles and mechanisms, or learn them from the teacher or textbook.

4. Modeling. Modeling questions deal with creating new models or critiquing existing models. At this level the learner is considering the system as a whole, rather than looking at one or a small number of variables. It requires an understanding that a model, formula, or simulation is an imperfect and/or approximate representation of the world [Soloway et al. 1997]. Inquiry occurs a meta level in comparison with concrete and relationship type questions. Examples: "What would happen if we replaced the Basal Area equation with a more complicated one that takes tree density into account?" "Can I build a model that causes birches to out-compete maples instead of the other way around as happens in nature?"

Associated with these question types is a parallel set of hypothesis types. Inquiry learning involves posing and testing hypotheses, and underneath each hypothesis is a question. The correspondence between question types and hypothesis types is: Concrete: prediction; Relationship: (scientific) hypothesis; Explanation: Explanation/theory; Modeling: a model. Students working on answering all of the question/hypothesis types use and improve their inquiry skills. They must articulate clear questions that can actually be answered. They must decide when they have collected enough data to be able to make an inference. They must be able to analyze data to infer trends, patterns, or rules. They must be willing to modify their hypotheses and preconceptions in the light of new data. With each successive inquiry question type the subject of inquiry is at a more abstract level.

Glass Box Simulations and the Model Inspector

Educational simulations are usually "black boxes" that do not give the student access to the underlying equations or models that run the simulation. In some cases this is purposeful. However, in most situations, for example a simulation of weather systems or genetic recombination, it would be beneficial for the student to be able to inspect the form and function of the equations or rules that drive the simulation.

In SimForest-G we open up the black box and create a "glass box" simulation in which students can inspect the equations that constitute a model of natural phenomena. Rather than encoding simulation equations in raw "code" we represent them as inspectable and editable "objects" in the simulation. The student uses the Model Inspector to access information about equations. The Inspector provides a consistent framework for accessing multiple representations of formulaic relationships (based on multiple epistemic forms as described in [Collins & Ferguson 1993]. For example imagine that the student asks the system to list the equations that refer to the tree's total basal area. A list of equations is shown, from which she picks the equation "SQI = (1-BAR)/BAMAX" for further inspection. The Model Inspector allows the student to see the following information:[6]

[6] Our original design specification (not yet implemented) also calls for multiple representations of these equations, including tabular and graphical forms to illustrate the relationship in multiple ways [Ainsworth 1999].

Equation	SQI = (1-BAR)/BAMAX
Textual representation of the equation	Soil Quality Index = (1 - Total Basal Area)/Maximum Plot Basal Area
Description	SQI is soil quality index, which determines how the intrinsic fertility of the site limits the growth of trees. It is a measure of how close the soil is to the maximum possible growth capacity for a given plot. For more information see [text book or URL reference].
Units	(The units in which the variable is measured)
Referents	SQI is referred to in these equations: … SQI refers to these variables: …
Assumptions, simplifications, and limitations to the equation	The equation assumes that tree circumferences are perfect circles.
Alternative equations	For a more complex equation that takes into account circumferences that are not perfect circles, see …

Table 1: Features of the Model Inspector

The forest growth simulation makes use of data tables listing parameters for each tree, species, and site (see Figure 2). We store pedagogical information for each of these parameters similar to that for the model's equations. As mentioned above, the student can also edit and add equations and table parameters. This comprises using the simulation in Model Editing mode as opposed to Model Inspection mode. When students add new equations they can also specify the description, units, etc. listed in Table 1. The relationships between the variable, equations, and table parameters constitutes a model information space. The software supports easy navigation from reference to referent (e.g. if an equation contains a variable defined in another equation, the user can easily navigate between these two equations). Below we describe how the knowledge based approach is used for each of the inquiry question types mentioned above.

1. Concrete/Situational ("What if?"). To answer concrete questions, students simply run the simulation, use the data gathering and analysis tools provided, see what happens, and come to conclusions. The Model Inspector is not needed.

2. Relationships ("How?"). Students can run experiments without inspecting equations to infer relationships between variables, but they can also learn much about these relationships through the Model Inspector. To observe emergent properties, they must run experiments.

3. Explanatory ("Why?"). Students can not inquire at the explanatory level with traditional educational simulations. This is due to the "meta model problem" which we will discuss below. With our knowledge based approach, students can get information about the underlying assumptions and principles behind an equation (see Table 1).

4. Modeling. A model is a set of equations that describe a phenomena. The knowledge based approach allows students to practice modeling by modify existing equations or create their own, thus allowing them to perform experiments in the "model space." They can load in equations or entire models created by the teacher or other students that embody alternate theories. They can also perform equation or model verification by running a model and comparing the results with what they observe in real world forests or historical data sets.

Emergent Properties and the Simulation Meta-Model Problem

Simulations of natural phenomena incorporate a particular model of phenomena and expose certain variables and processes to observation. There are also underlying *assumptions* and *emergent phenomena* that are underneath or implied in the model, and outside the scope of the simulation, yet important educational topics. All models of physical systems have such limitations. Most equations represent empirically derived laws or underlying scientific assumptions. The equation itself does not tell you anything about where it came from or what assumptions are built into it. From any equation or model one can always go one level deeper and ask "why?" For exampleStudents studying biology need to be quite facile with biological models, and need to have just some familiarity with the chemistry that underlies them. For every discipline it suffices to have introductory information about the next deeper level of causality. In SimForest-G this is provided in the textual information associated with equations (see Table 1). Note that the "why" level of information *must* be in canned text form. If it was represented in a computational way it would be part of the model itself. This would push the "why" question to the next deeper level, but a "why" question would still remain.

At the opposite side of things, in addition to its underlying assumptions every model has emergent properties that deal with a causal level "higher" than the equations specified in the model. For example, we want students to be able to observe or infer global trends such as "species diversity decreases with the age of the stand." This relationship is not to be found explicitly in the model, yet it can be observed by running a number of simulations trials. Figure 6 illustrates the meta-model problem: that for any mathematical model of a natural phenomena, there will always be lower level fundamental assumptions, and higher level emergent properties that are not represented explicitly in the model's equations, yet are important for students to become familiar with.

Fundamental assumptions (e.g. physics)	→	SIMULATION MODEL (e.g. chemistry)	→	Emergent properties (e.g. biology)

(→ increasing model granularity →)

Figure 6: The Meta-Model Problem

Observed emergent properties often involve variables or properties not specifically mentioned in the model (such as "species diversity"). Using the model editor's composite operations the learner can define new variables and observe and analyze emergent properties. Also, we address the emergent properties issue via the curriculum that accompanies the software. Student activities and teacher guides facilitate asking questions, making observations, and coming to conclusions related to emergent properties.

RELATION TO PREVIOUS WORK

In general terms, we expect that beginning learners will be interested in using the simulation to run experiments (black box mode); intermediate learners will be motivated to open up the box to learn about the model and the scientific principles behind the model; and advanced learners will be motivated to alter and experiment with the model itself. This progression is reminiscent of the sequence (or cycle) of 1) inquiry (simulation), 2) reflection (model inspector), and 3) generalization (model editor) mentioned in [White & Frederiksen 1995], but at a larger granularity. In addition, moving between black box and glass box inquiry is reminiscent of the distinction between the experimental (or instance or data) space and the hypothesis (or rule) space in the scientific discovery dual search space (SDDS) paradigm described in [van Joolingen & de Jong 1996; [Klahr & Dunbar 1988]). The comparison is limited because in the SDDS paradigm all scientific discovery involves search in both the experimental space and the hypothesis space as part of the inquiry cycle, while in using SimForest-G the learner can focus on only the model space or the simulation space (or both).

As mentioned, there are many educational simulations designed as inquiry-based learning environments that use a back box model. In addition, a number of software applications have been developed that allow students to learn by creating new models or programs from scratch (Model it [Soloway et al 1997], Logo and its derivatives [Papert 1980; Resnick et al 2000], Stella [High Performance Systems], Boxer [diSessa & Abelson 1986], Smithtown [Shute & Glaser 1990], and AgentSheets [Repenning & Summer 1995]). Other systems allow the *teacher* to create models, and the students then practice inquiry within the simulation (SimQuest [van Joolingen & de Jong 1996], RIDES [Towne & Munro 1988], ThinkerTools [White & Frederiksen 1995]). Our project is distinguished from others in several ways. First, ours is the only system that supports students in doing inquiry in both the simulation space and the model space (with the possible exception of SmithTown). Second, we provide features that support for the acquisition of both general inquiry and modeling skills *and* domain specific concepts and principles. The SimForest project occupies a rare place right in the middle of the spectrum between black box simulations and completely open ended modeling environments. SmithTown and ThinkerTools also provide support for domain concept learning in specific domains, but neither are full glass box systems. Third, SimForest provides pedagogical-level support for student exploration of model components. SimQuest and ThinkerTools provide direct pedagogical support in the form of explanations, and hints, but these are associated with activities, not with model components.

Finally, SimForest is the only system among those that let *students* build models that starts with a fully functioning domain model. There has been little empirical evidence supporting the acquisition or transfer of general skills in modeling and programming, and current cognitive theories suggests that expert knowledge is closely tied to domain-specific contexts. Starting with a full simulation model is consistent with these findings. Other systems allow students to building models from scratch, and in starting from scratch most learners build only "toy" models. The "from scratch" systems do include complex example models, but these are provided as examples, not as robust starting points for inquiry (with the exception of Model-It, and perhaps some Stella projects). In the course of an instructional sequence learners might build a series of toy models in different domains. Our approach is to start off with a fully working simulation model–the approximately 25 Botkin equations for forest growth. This may seem to be more complex than starting from scratch and building a toy model, but in fact it acts as a form of scaffolding. Learners can inspect and alter just one equation and observe the results in a robust simulation of the forest. Instead of modifying equations users can load alternate versions of equations, as provided by teacher or peers, that embody competing theories. In either case, the learner can theoretically inspect or model each equation separately, then play with some equations in combination, until in the end they can try to build the entire model from scratch (an extreme that most students are unlikely to reach, but getting part of the way there has educational benefits). Curriculum that supports this increasing learner ownership of the model would be the "fading" that accompanies the scaffolding of providing the full model from the start. SimQuest and ThinkerTools both use "model progression" methods, in which the student is exposed to increasingly more sophisticated versions of a model as they discover that a simpler version is insufficient to answer one of their questions.[7] These systems expose the student to intermediate models (as opposed to from-scratch or complete models; but they do not let students modify the model). We might take a similar path is providing a series of complete but increasingly sophisticated forest growth models.

There are a number of problems with giving novice learners the freedom to alter any part of the model. They might do something to the model that they can't understand and can't "back out of." We do not expect such problems to preclude the glass box approach. Rather, we will use the results of user testing to modify the software and curriculum to constrain how much trouble the learner can get into in their beginning use of the system (another form of scaffolding, which will be faded as the learner progresses).

---

[7] SimQuest is the only other system that supports *alternate* models embodying alternate theories. Only SimForest supports alternate versions of both entire models and model components and allows both teachers and peers to author these alternatives.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science foundation under Grant No. DUE-9972486. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. We would like to acknowledge the contributions of the following people: Ryan Moore for developing SimForest-B, Roger Bellin and Matt Cornell for developing SimForest-G, Ester Shartar for developing curriculum materials and assistance with evaluation, Steve Lester for software testing and model verification, Jesse Doane for developing graphic tools, and Neil Stillings for assistance with evaluation design and implementation.

BIBLIOGRAPHY

Ainsworth, S. (1999). Designing Effective Multi-Representational Learning Environments. University of Nottingham ESRC Tech report No. 58.

Collins, A. & Furguson, W. (1993). Epistemic Forms and Epistemic Games: Structures and Strategies to Guide Inquiry. *Educational Psychologist*, Vol. 28, pp. 25-42.

Collins, A. & Stevens , A. (1993). A Cognitive Theory of Inquiry Teaching. In Reigeluth (Ed.) *Instructional Design Theories and Models*, pp 2470278. Erlbaum: Hillsdale, NJ.

de Jong, T. & van Joolingen, W. (1998). Scientific Discovery Learning with Computer Simulations of Conceptual Domains. *Review of Educational Research*, Vol. 68, No. 2. pp. 179-201.

diSessa, A. & Ableson, H. (1986). "BOXER: A Reconstructible Computational Medium," *Communications of the ACM*, Vol. 29 No. 9.

Elliot, P. (1978). Computer `Glass Boxes' as Advanced Organizers in Mathematics Instruction. *Int. J. of Math, Science, and Technology Education*, Vol. 9 No. 1.

High Performance Systems, Inc. Stella software. www.hps-inc.com.

Khahr, D. & Dunbar, K. (1988). Dual Space Search During Scientific Reasoning. *Cognitive Science,* 12, 1-48.

Towne, D.M. &, Munro, A. (1988). The Intelligent Maintenance Training System. In Psotka, Massey, & Mutter (Eds.), *Intelligent Tutoring Systems, Lessons Learned*. Hillsdale, NJ: Lawrence Erlbaum.

Papert, S. (1980). MindStorms: Children, Computers, and Powerful Ideas, Basic Books, Inc., New York.

Repenning, A. & Summer, T. (1995). Agentsheets: A Medium for Creating domain-Oriented isual Languages. *IEEE Computer*, March 1995, pp. 17-25.

Resnick, M., Berg, R., Eisenberg, M. (2000). Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Investigation. *J. of the Learning Sciences,* 9(1), pp. 7-30.

Shute, V.J. and Glaser, R. (1990) A Large-scale Evaluation of an Intelligent Discovery World: Smithtown. *Interactive Learning Environments*, 1: 51-77.

Soloway, Pryor, Krajik, Jackson, Straaford, Wisnudel, & Klein (1997). ScienceWare Model-It: Technology to Support Authentic Science Inquiry. *T.H.E. Journal*. pp. 54-56.

Stratford, S. (1997). A Review of Computer-Based Model Research in Pre-college Science Classrooms. In *J. of Computers in Mathematics and Science Teaching* (1997) 16(1), 3-23.

van Joolingen, W., & de Jong, T. (1996). Design and Implementation of Simulation Based Discovery Environments: The SMILSE Solution. *Jl. of Artificial Intelligence in Education* 7(3/4) p 253-276.

White, B. & Frederiksen, J. (1995). Developing Metacognitive Knowledge and Processes: The Key to Making Scientific Inquiry and Modeling Accessible to All Students. Technical Report No CM-95-04. Berkeley, CA: School of Education, University of California at Berkeley.