Instructivism: Between CAI and Microworlds

Paul Horwitz

The Concord Consortium

There is a controversy in education as old, perhaps, as education itself. It is between the teaching of *content* and *process*. In science education, this conflict appears as an argument over whether to teach science as a series of "stories" about the workings of the natural world, or to concentrate instead on teaching the scientific reasoning behind the discovery, validation, and refinement of those stories. In truth, of course, one tries to teach both, recognizing that either extreme is counterproductive. Science is more than a collection of "facts," but it is pointless to teach students to think without giving them anything substantive to think about.

In terms of educational software, the same controversy often appears as a tension between two well known paradigms: computer-aided instruction (CAI) and the "microworlds" most closely identified with the LOGO programming language. The former of these is mainly concerned with teaching content; the latter, process. Again, neither is the final answer – what is needed is a way to strike a balance between them. The Modeling Center at the Concord Consortium has been experimenting with a new technology that bridges the gap between the explicit instructional approach typified by CAI and the constructivist style of the microworld. We call this technology a "hypermodel" (Horwitz, 1995; Horwitz and Christie, 1999).

The hypermodel is an approach to the design of science education software that integrates stored information in the form of multimedia materials, experimental data, and text, with a manipulable model of the subject domain, using each medium as a tool for navigating the other. Hypermodels combine the advantages of manipulable computer-based models with the structure and content presentation capability of multimedia. They enable a curriculum designer to create interactive problem-solving activities that embed questions and answers into flexible, performance-oriented challenges involving the manipulation of the underlying computer model. Many of these activities offer "outside" readings or links to web pages designed to associate students' explorations of the model to real-world analogs. Hypermodel activities track students' actions and react to them intelligently, offering general problem solving assistance such as metacognitive prompts, as well as specific, contextualized content information. This monitoring function is also used to create a log file of students' investigations and responses to questions, making a fine-grained time-series record available to researchers and teachers for analysis and assessment.

From the technology point of view, hypermodels are implemented in the form of scripts that communicate with an underlying manipulable model of a domain. Below, we present a specific example that may help to make this architecture clear.

GenScope[1][2] is a computer-based model of Mendelian genetics (Horwitz, Neumann, and Schwartz, 1996; Hickey et al, 1999). It is composed of six different levels, representing DNA, chromosomes, cells, organisms, pedigrees, and populations. Each of these levels is composed of *representations* and *affordances* – each, in other words, shows the student something and lets the student do something. The chromosome level, for example, enables the user to change a gene from one allele to another. Such a change may, or may not, produce a change in the organism that "owns" the gene; it will definitely show up as a change in the DNA sequence of the gene, and it will affect approximately half the offspring of the organism, just as it would in real life. GenScope embodies a model of genetics within a general-purpose interface that enables students to explore all aspects of the model. What GenScope does not have is any embedded curriculum or assessment functions. The program is designed as a "one size fits all" tool – it enables a student to explore the underlying genetic model, but it does not incorporate any specific pedagogical goal.

BioLogica™, in contrast, is a genetics *hypermodel.* Rather than presenting every student with the same general-purpose interface, BioLogica is actually a collection of instructional *activities* that make use of components that correspond to five of GenScope's six levels (the population level is not yet implemented). Each activity poses a question, problem, or challenge to the students, then monitors and reacts to them as they explore the model. Thus, where GenScope simply gave the students the ability to change genes (and tried to make the interface for doing this as straightforward as possible) BioLogica offers a specific challenge ("change this organism so that it looks like this one"), and then reacts to the successful completion of the task by offering a new activity that focuses the students' attention on the rules that determine phenotype from genotype. BioLogica activities ask questions – both multiple choice and essay – very much as a CAI program might, but they also allow the students free rein to explore a model space in the way they might with a microworld. The activities record students' responses — both their answers to questions and their purposeful manipulations of the model – and makes them available for analysis by the teacher and by the researcher.

Hypermodels divide responsibility for creating what the student sees between the model designer who writes the software that embodies the underlying model, and the curriculum designer who writes the scripts that implement particular activities[3]. In the case of the BioLogica example above, for instance, the model designer wrote the software that "knows" about organisms and genes, and when changes in one should produce changes in the other; the curriculum designer decided what the student should actually *do* with this model, and wrote the scripts that make that happen.

[1] You can learn more about GenScope, and download a copy of the software, by visiting our website at http://www.concord.org/genscope

[2] You can learn more about BioLogica, and download a copy of the software, by visiting the Modeling Across the Curriculum page at http://mac.concord.org

[3] These can, of course, be the same person, but usually they are not and even if they are that person operates under entirely different constraints and with different goals in the two capacities.

Among other desirable features, this modular software architecture greatly simplifies the design of the user interface. In thinking about the interface to an ordinary computer-based model, the hardest decisions one makes are those involving the configuration of the model – and the more complex the model the more such decisions there are. Which features should be turned on by default, and which reserved for use by \more "advanced" students? Should a graph automatically scale itself to adapt to the data, or should that be left up to the student? Should a student's use of the model be placed in some sort of real-world context, and if so which? How much information should a student be able to get: for example, in the case of a BioLogica activity, should the student be allowed to view the chromosomes of an organism, or be forced to infer its genotype from other information?

Of course the answers to these questions depend on the level of expertise of the student and the particular concept one is trying to teach, and therefore they must be made in the context of particular activities. A hypermodel takes the responsibility for making such choices away from the model designer, and places it in the hands of the curriculum designer instead. This gives the curriculum designer the freedom to create each activity by starting from explicit teaching goals, configuring the model, reacting to particular student responses, and choosing specific challenges based on pedagogical considerations rather than software constraints.

All this discussion brings up some obvious questions: what constitutes a "good" hypermodel activity and how would we know that a given activity was good? What general principles of good instructional design should we turn to in creating this new kind of interactive curriculum? Can activities be written so as to accommodate different learning styles, and if so should they be? How can one best embed formative assessments within an activity? Can feedback from such assessments be used to "prep" students and increase their success in taking a conventional, paper-and-pencil summative exam?

One of the central questions that we are studying is how much structure to build into an activity. The two extremes referred to at the beginning of this paper are clearly sub-optimal: too much structure and an activity becomes boring, too little and the average student will flounder. Finding the "happy medium" between explicit instruction and radical constructivism is the central quest that informs much of our approach, which we characterize as "instructivist," to convey that it seeks a Hegelian synthesis of these two opposing philosophies.

Hypermodel activities are very general: depending on their creator's guiding philosophy, they can be written so that they amount to little more than canned demonstrations interspersed with multiple-choice questions; alternatively, they can simply offer on-line documentation to a complex model and leave it at that. Which route one chooses depends, in large part, on one's answer to the fundamental question we asked at the beginning of this document: is the goal to teach science facts or scientific reasoning?

Regardless of how one answers that question, there are some simple empirical questions to be answered that will bear on how one uses hypermodels. We need to determine how much, and what kind of, scaffolding will produce what kind of educational outcomes. Given enough time and other resources, we should be able to ascertain, for instance, what affect the presence or absence of metacognitive prompts may have on students' learning – either of science content or of higher order reasoning skills. The methodological problems of actually making such a determination are, of course, formidable, but some of them can perhaps be solved by making clever use of the technology itself.

As we have noted, hypermodel activities – even those directed at the same learning goals – can differ significantly. Different authors often make very different choices – e.g., with regard to the nature and frequency of scaffolds. At present these choices are being made somewhat intuitively, guided, to be sure, by instructional principles drawn from the learning literature (Wiggins and McTighe, 2001; Bransford, Brown and Cocking, 2000). However, the research base is not adequate to inform the design of the kind of interactive curriculum made possible by the hypermodel technology. What we need is solid evidence of differential learning gains in both content and process skills that can be correlated to alternative "treatments" in the form of hypermodel activities.

Our group at the Concord Consortium is preparing to use hypermodels to undertake a large-scale study to obtain this kind of evidence. We are preparing hypermodel activities in physics, biology, and chemistry which we hope eventually to install in dozens or hundreds of volunteer testbed schools. We will then randomly assign alternative versions of these activities to different students *within the same class.* Each activity will log the students' actions and send the files to us over the internet. We are currently working on ways to analyze these files. At present, we merely track students' answers to explicit questions, but we can also use the log files to trace their route through the investigations and we hope to triangulate these two very different sources of data and to correlate them to learning gains. If we can automate the analysis process sufficiently, we hope to repeat the experiment with a sufficient number of students, teachers, and schools, so as to collect reliable, replicable evidence bearing on the question of how best to construct hypermodel activities to achieve pre-defined learning objectives.

If educational goals may legitimately differ between different schools within the United States, how much more must national education systems differ? Within the EU, for example, it is likely that some member states tend to emphasize content knowledge over process skills and inquiry, while others opt to go the other way. As we have discussed, hypermodel technology, in contrast to the microworld or computer-aided instruction, is rather neutral in this regard, in that hypermodel activities can be written to fall anywhere along the spectrum. But that begs the question of "localization." If BioLogica, for instance, is to be used in countries other than the United States, it will probably take a lot more than merely translating the text from English into some other language.

Concord Consortium has an intense interest in understanding how the implications of advanced educational technology "play out" in different national contexts. We would greatly enjoy talking to anyone who wished to try out our hypermodels, and would be happy to cooperate in making them match a variety of educational goals.

References:

Bransford, Brown, and Cocking (2000), How People Learn, National Academy Press.

Hickey, D. T., Kindfield, A. C. H., Christie, M. A., and Horwitz, P. (1999), Advancing Educational Theory by Enhancing Practice in a Technology-Supported Genetics Learning Environment, *Journal of Education, Volume 181, Number 2*

Horwitz, P. (1995), Linking Models to Data: Hypermodels for Science Education, *The High School Journal, Vol. 79, No. 2*, *pp. 148 - 156.*

Horwitz, P., Neumann, E., and Schwartz, J. (1996), teaching Science at Multiple Levels: the GenScope Program, *Communications of the ACM, Vol. 39, No. 8*

Horwitz, P. and Christie, M.A. (1999), Hypermodels: Embedding Curriculum and Assessment in Computer-Based Manipulatives, *Journal of Education, Volume 181, Number 2(1999), pp. 1 – 23*

Wiggins, G. and McTighe, J. (2001), Understanding by Design, Merrill Prentice Hall