Chapter 3
Learning Genetics from Dragons: From Computer-Based Manipulatives to Hypermodels

Paul Horwitz, Janice D. Gobert, Barbara C. Buckley, and Laura M. O'Dwyer

This chapter continues a theme explored in earlier research (Horwitz & Christie, 2000) related to the use of a *computer-based manipulative* called "GenScope" for teaching high school genetics. The major finding from that work was that although GenScope proved immensely popular among both students and teachers,[1] the learning results associated with its use were initially disappointing and only improved after the software was accompanied by a customized curriculum and extensive professional development (Hickey, Kindfield, Horwitz, & Christie, 2003; Hickey, Kruger, & Zuiker, 2003). In the present chapter we focus on the changes that were made to GenScope in response to these findings, and describe research and development efforts toward and with a "new and improved" version of the software called BioLogica. One of the main thrusts of the research we report on here has been the addition to GenScope of logging tools that enable us to: (1) monitor students' actions, including but not limited to their answers to embedded assessment questions, and (2) analyze them to make inferences concerning their content knowledge and model-based reasoning. The results of these fine-grained analyses were mainly used to inform our research, but in future could form the basis for timely, insightful reports on student learning, targeted for teachers and students, respectively.

P. Horwitz (✉)
Concord Consortium, 25 Love Lane, Concord, MA 01742, USA
e-mail: paul@concord.org

J.D. Gobert
Department of Social Science and Policy Studies, Worcester Polytechnic Institute,
100 Institute Road, Worcester, MA, USA

B.C. Buckley
WestEd, 400 Seaport Court, Suite 222, Redwood City, CA, USA

L.M. O'Dwyer
Department of Educational Research, Measurement and Evaluation, Lynch School of Education,
Boston College, Chestnut Hill, MA, USA

[1] GenScope remains popular today, but its use is limited to pre-OS X Macintosh computers.

M.J. Jacobson and P. Reimann (eds.), *Designs for Learning Environments of the Future: International Perspectives from the Learning Sciences*, DOI 10.1007/978-0-387-88279-6_3,

The GenScope Project (Horwitz, Neumann, & Schwartz, 1996; Horwitz, Schwartz, & Neumann, 1998; Hickey, Kindfield, Horwitz, & Christie, 1999; Horwitz & Christie, 2000; Hickey, Kindfield et al., 2003) developed a computer-based model of inheritance genetics consisting of six interacting levels representing, respectively, DNA, chromosomes, cells, organisms, pedigrees, and populations.[2] The levels are linked so that changes made at any one of them will affect the others as dictated by the underlying genetic model. Thus, a change at the DNA level (i.e., a mutation) will usually[3] create a new allele that, alone or in combination with a similar allele on the homologous chromosome, may give rise to a new phenotype, thus affecting not only the chromosome and cell levels, but the organism level as well. The new phenotype, in turn, will be inherited in a stochastic but deterministic way that can be studied at the pedigree level, and the new allele may or may not prove adaptive in a given environment, which will govern its subsequent increase or decrease in frequency at the population level.

GenScope provides a powerful and extensible model of genetics,[4] but it lacks explicit pedagogical guidance. For example, one study tested it using three high school classes: a traditionally taught class that served as the control group, and two GenScope classes, one in which the students exclusively used GenScope, and one in which the students used GenScope less intensively, but added a set of pencil-and-paper activities to the treatment (for a review of this study see Hickey, Kindfield et al. (1999)). Both GenScope classes outperformed the traditional class on the posttest, but the GenScope class that also did the paper-and-pencil instructional activities outperformed the GenScope-only class. These and other findings highlighted the need for a system that would include instructional activities with the software, guiding students' interactions with the genetics model, posing questions, and making explicit the connections between the behavior of the computer model and corresponding real-world processes.

In addition, our experience with GenScope led us to believe that the software might be designed to interpret and react to students' actions using context-sensitive algorithms, thereby providing individualized instruction. With this as a starting point, we were able to formalize the components and characteristics that would be needed in a more adaptive program. Two key features of such a system are:

- *Student feedback*. The students should receive context-sensitive assistance from the software, so that they need not rely entirely on the teacher, either to help them use the software effectively or to guide them to draw appropriate conclusions from their investigations.

[2] The population level was not used in the research described in Horwitz and Christie (2000).

[3] So-called silent mutations, which do not alter the encoded sequence of amino acids, have no effect in GenScope or in the real world.

[4] The model includes recombinant processes – such as crossing over between homologous chromosomes during meiosis – as well as interspecific interactions such as predator–prey and competitive relationships.

- *Teacher feedback*. The teacher should receive feedback about the students' use of the software in order to identify who is "getting it" and who is "stuck." This is important because it is difficult even for an exceptionally well-prepared teacher to determine what each student in the class is doing during online learning activities and to react accordingly.

Addressing these issues required us to depart from the open-ended discovery approach underlying GenScope in favor of explicitly scaffolding students' learning activities (more on this later). In order to accomplish this goal, we created an infrastructure that monitored students' actions and reacted to them in real time. By logging and subsequently analyzing these actions, we were able to create reports from the formative and summative interactive assessments for use by the researchers. Only limited reports, consisting primarily of students' answers to embedded assessment questions, were available to teachers. The technology required to do both the scaffolding and the assessment is what we have come to call a "hypermodel."

Hypermodels

Hypermodels (Horwitz et al., 1996) occupy a position in the educational technology spectrum somewhere between the highly linear, explicitly didactic approach characterized by the term "computer-assisted instruction" or CAI (Suppes, 1969; Steinberg, 1977; Kinzie, Sullivan, & Berdel, 1988), and the more open-ended, student-centered technologies often termed "constructivist" (Magoon, 1977; Papert, 1980; Driver & Oldham, 1986; Blais, 1988). The development of hypermodels was motivated by perceived drawbacks at the two extremes. CAI technologies, though they have been demonstrated to be effective in enhancing procedural skills, are less successful at teaching higher-order thinking and eliciting students' mental models (White & Frederiksen, 1998). On the other hand, the research literature has shown that students who use open-ended constructivist tools with little or no structure may become proficient in using the tool, but they often fail to convert that success into deep understanding of the subject domain (Horwitz 1999; Horwitz & Christie, 2000; Hickey, Kindfield et al., 2003; Hickey, Kruger & Zuiker, 2003; Kirschner, Sweller, & Clark, 2006).

Hypermodels are intended to respond to the demands placed on teachers when they use open-ended inquiry tools like GenScope. These demands often present significant barriers to the successful implementation of such technologies in real classroom settings. Although open-ended applications such as GenScope often "demo well," the practical difficulties of using them in the classroom may overwhelm the teacher, who must keep track of what students are doing, guide them through difficulties, encourage and praise them when they succeed, and help them reflect on the broader significance of what they are doing (Aulls, 2002).

As previously stated, hypermodels are designed to alleviate these problems by combining the best aspects of the CAI and constructivist approaches. Properly used, they give students the freedom to engage in open-ended investigations, while monitoring their actions and reacting to them in contextually meaningful ways – offering suggestions, asking questions, and using text or multimedia materials to link the computer activities to real-world analogs. Hypermodels integrate text, audio, animations, or video materials with a manipulable model of the subject domain, using each medium as a tool for navigating the other. The association with "hypertext" is intentional: just as clicking on a word, phrase, or graphic serves to navigate through a website, students' manipulation of a computer-based model can be used to navigate in an interactive model-based inquiry environment – triggering the presentation of a video clip, for instance, or bringing up a relevant question. In turn, students' answers to questions or choices of multimedia presentations can affect the configuration of the model.

Hypermodels are *scriptable* by curriculum developers and researchers, and thus provide a flexible tool for the creation of a wide variety of activities that can challenge students to solve problems, and then monitor and react to their actions. The activities structure students' investigations of a domain and offer metacognitive prompts as well as links to real-world science at appropriate "teachable moments."

Since hypermodels monitor students' interactions with the learning environment, they can also log them. The raw data produced by this process is too fine-grained to be of immediate practical use, but it can be analyzed and summarized so as to produce insightful progress reports for teachers, researchers, and the students themselves. We use these data to assess students' understanding of the subject matter, as well as to provide indices of their model-based inquiry within a domain. Logging students' data in this way provides researchers with a "bird's eye view into the black box" (Gobert, 2005), permitting a different lens on human learning than think alouds (Ericsson & Simon, 1980), which are often used to "get at" real-time learning processes. Logging students' interactions in this way provides us a trace of what students are doing without the face validity problems that can be encountered when using think aloud protocols (Nisbett & DeCamp Wilson, 1977). Additionally, logging complements think aloud protocols in that the two sources of data can be triangulated; in fact, early in the design phase of BioLogica, think aloud protocols were collected as indices of what the students were thinking as they proceeded through the activities. Think aloud data provided us some critical information about what scaffolding would be needed for students.

The first hypermodel we produced, and the one to be described in this paper, was *BioLogica,* a scriptable version of GenScope written in Java so as to run on the Windows, Macintosh, and Linux operating systems. In contrast to GenScope, BioLogica was designed as a more tightly scaffolded sequence of activities designed to teach students about Mendelian genetics through their interactions with the software. This scaffolding was intended to not only improve the students' model-based learning (Gobert & Buckley, 2000), but also to strengthen their inquiry skills in the context of their exploration of the underlying genetics model. Next we describe in more detail the theoretical framework underlying our activities and scaffolding.

Theoretical Framework: Model-Based Learning

The theoretical framework that guided the development of BioLogica activities and the scaffolding implemented in the Modeling Across the Curriculum (MAC) project stems from a synthesis of research in cognitive psychology and science education. As shown in Fig. 3.1, model-based learning (MBL) is a dynamic, recursive process of learning by constructing mental models (Gobert & Buckley, 2000). In the MAC project, it occurs through the interaction with the hypermodels of each domain. Model-based reasoning (MBR) involves the formation, testing, and reinforcement, revision, or rejection of mental models during interaction with hypermodels and other representations. MBL requires modeling skills and reasoning during which mental models are used to create and/or understand representations, generate predictions and explanations, transform knowledge from one representation to another, analyze data, and solve problems. It is analogous to the hypothesis development and testing observed among scientists (Clement, 1989).

In the classroom many factors influence the learner's mental models including characteristics of students and teachers such as their understanding of the nature of scientific models (Justi & Gilbert, 2002; Lederman, 2006; Gobert, O'Dwyer, Horwitz, Buckley, Levy, & Wilensky, revisions submitted). We now discuss evidence that students' use of hypermodels such as BioLogica can provide important information about both classroom usage and student learning.

Using a progressive model-building approach (White & Frederiksen, 1998), we developed 12 BioLogica activities that guide students through interaction with basic models of meiosis and fertilization and progress through increasingly elaborate models of inheritance.

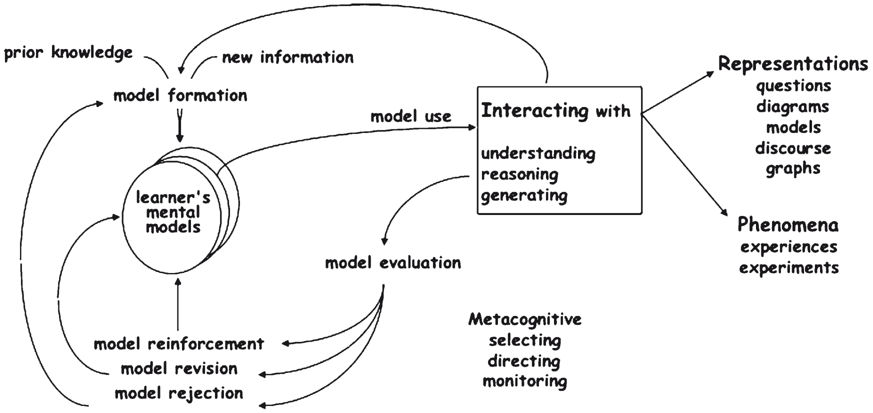


Fig. 3.1 Model-based learning framework

Scaffolding Model-Based Learning

In the MAC project we formalized the scaffolding that guides feedback to students. General scaffolds are based on a large research base in educational psychology and include: (a) advance organizers to evoke prior knowledge and provide students with a structure to fill in the concepts, (b) orienting tasks to give the student a cognitive goal for the task, (c) post organizers to encourage students to reflect on and concretize what they have just learned, and (d) a glossary of terms.

We implemented five model-based scaffolding elements to support the knowledge acquisition and reasoning required for progressive model-building (Gobert, Buckley, & Clarke, 2004).

- *Representational assistance* to guide students' understanding of representations or domain-specific conventions.
- *Model pieces acquisition* to focus students' attention on the perceptual components of the representations and to support their learning of one or more aspects (spatial, causal, functional, temporal) of the phenomenon or process under study.
- *Model pieces integration* to help students combine model components in order to come to a deeper understanding of how they work together as a causal system.
- *Model-based reasoning* to guide students' reasoning with their models.
- *Reconstruct, Reify, and Reflect* to encourage students to refer back to what they have learned, reinforce it, and then reflect to move to a deeper level of understanding.

Scaffolding of each type was implemented in the form of questions, assigned tasks, or explanations that focused on a phase of model-based learning, followed by feedback. The nature of the feedback varied according to the pedagogical purpose of the scaffolding. For example, we sometimes taught learners how to check their own answers, and we also used students' actions or answers to tailor the feedback that they received.

Activity Description

In all, we developed 12 BioLogica activities. Here we present a detailed description of the introductory activity. The remaining activities are described briefly in the appendix to this chapter.

The first activity in the BioLogica sequence is intended to introduce students to the idea that observed differences among organisms may be due to their genes. As we did with GenScope, we illustrate this and other concepts using dragons as a fictitious species.[5] The introductory activity starts off with a blank screen and just two buttons: one for creating male dragons, the other for creating females. Once the students have created their first dragon, they are asked to describe it, and then to make enough additional dragons to fill up the screen. BioLogica's dragons can differ in several ways: presence or absence of horns or wings, shape of tail, number of legs, color, and ability to breathe fire among others. These physical traits (which represent the dragon's *phenotype*) are randomly chosen each time a dragon is created, so the dragons on the computer screen tend to look quite different from one another. The students are asked to describe these differences. They are then introduced to some vocabulary, following which they are requested to "Click a dragon to see that dragon's genotype." (See Fig. 3.2.)

The students are then shown representations of chromosomes as box-like objects with lines drawn across them to represent the locations of genes. This depiction of chromosomes is common in biology textbooks and is intended to represent the linear nature of the DNA molecule that is at the core of the chromosome. BioLogica's chromosomes differ from the ones in the textbook, however, as they are "active": that is, one can alter the form of their genes and immediately observe the effect of the change, if any, on the organism. Most of the genes in the fictional dragon genome are actually modeled on those Mendel investigated in his famous pea experiments. They come in only two variants, or "alleles," and these combine according to Mendel's First Law.[6] The students are not told this, however. Rather, they are led to uncover this pattern, as well as the more complicated patterns of the other genes, by direct experimentation.[7] Once the students have made a certain number of changes in their dragons, BioLogica takes them back to the screen where they made the original eight dragons. It requests that they click on a dragon of the opposite sex, monitors to make sure they have done so, then puts up a new screen in which the students can compare the chromosomes of the two genders of dragon, and discover for themselves the differences between them.

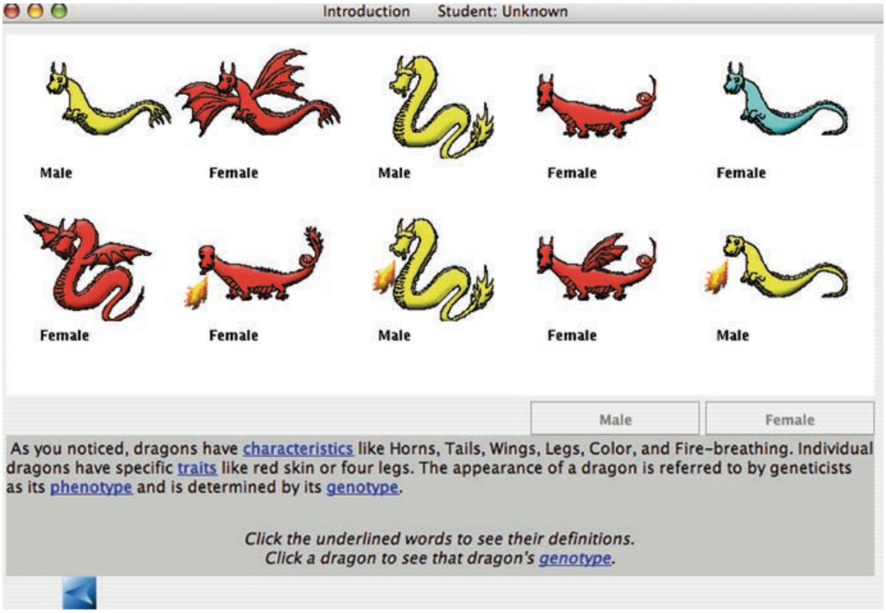
After some questions and simple tasks, this introductory BioLogica activity eventually challenges the students to match their manipulable dragons to a "target" dragon of fixed phenotype. We explicitly chose not to mention that this can only be done with one of the dragons (because male and female dragons are different colors), but instead allow the students to uncover this fact independently.

All data collected by a BioLogica activity (which, in addition to students' answers to questions, can include the number of times they change a gene from one allele to another, or whether or not they examine the chromosomes of a particular organism using the "chromosome tool") is stored and made available for research. It is very easy, for example, to administer pre- and posttests in this way and to collect and score the data automatically. Indeed, from a software point of view, the assessments that were administered to the students were simply BioLogica activi-

[5] We use dragons for two reasons: (a) since everyone knows that they are imaginary, we can simplify their genetics without doing violence to the complexity of real organisms, and (b) by avoiding even a vague reference to real organisms (e.g., humans), we are able to postpone discussions of ethical and legal issues until the students have learned the science underlying them.

[6] One of the alleles is "dominant," the other "recessive." If an organism inherits one or two copies of the dominant allele it will exhibit the dominant trait; if it has two copies of the recessive allele it will exhibit the recessive trait.

[7] Note that this kind of computer activity is not a "simulation"; that is, even if one could alter an organism's genes, one would not expect the organism itself to react to the change.



As you noticed, dragons have characteristics like Horns, Tails, Wings, Legs, Color, and Fire-breathing. Individual dragons have specific traits like red skin or four legs. The appearance of a dragon is referred to by geneticists as its phenotype and is determined by its genotype.

Click the underlined words to see their definitions.
Click a dragon to see that dragon's genotype.

Fig. 3.2 A screen shot from the Introduction activity

ties consisting entirely of questions, which were made to look just like traditional paper-and-pencil items. It is important to note that for all BioLogica activities, answers to open-response questions are not parsed or analyzed in any way by the computer. The reason for including such questions, therefore, is not to influence future actions on the part of the computer, but to give students a chance to explain their state of knowledge, and to encourage them to reflect on what they have learned. All freestyle answers to essay questions are recorded by BioLogica, however, and made available to the teacher (as well as to researchers). This enables the answers to be evaluated and used as a component of the student's grade.[8]

Throughout all the BioLogica activities, we scaffolded students' interactions with the hypermodels as they worked their way through increasingly complex tasks. Within each activity we faded the scaffolding as they progressed. In the following section we describe the technological infrastructure underlying BioLogica, which permits fine-grained monitoring and logging of students' interactions within the activities.

Technological Details

To understand how a hypermodel works, it is helpful to take a look at the structure of the software itself. Hypermodels consist of three separate software layers embedded in an architecture designed to separate functions relating to domain content from more general ones relating to pedagogy (see Fig. 3.3).

At the lowest level of the hypermodel architecture is the *domain content engine*. This consists of a set of loosely coupled components, or *views,* which may be combined and integrated in a variety of ways. For instance, in the BioLogica hypermodel described above, the chromosome view and the organism view share a common database that contains, among other things, the genotype of every organism created so far. One of these views uses this information to display alleles on chromosomes, whereas the other, operating with a set of built-in rules, determines and displays the phenotype of each organism. Manipulations performed in the chromosome view that change a gene, say from a dominant to a recessive form, will be reflected, as appropriate, as changes in an organism's phenotype, represented in the organism view by, for example, the presence or absence of horns on a graphic of a dragon. Each view in BioLogica is implemented as a Java class, and each is capable of saving its state using the XML markup language. BioLogica's views are purposely kept quite simple. They are incapable of operating independently, but must be placed on the screen and configured by the next level up in the hierarchy, *Pedagogica.*

Pedagogica, as the name suggests, handles all things pedagogical. It is responsible for all interface details, including the placement of text boxes, buttons, and domain engine views in various locations on the screen. Pedagogica also controls the flow of an activity by shifting from one set of views to another in response to student

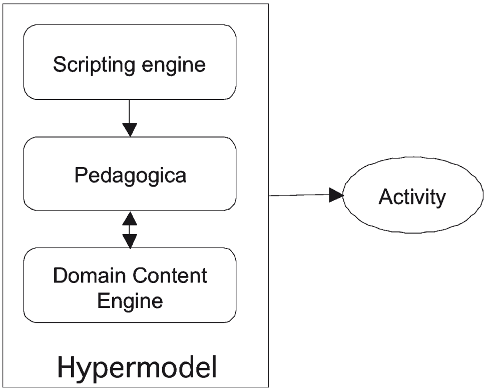


Fig. 3.3 Architecture of a hypermodel

[8] Without such formal accountability, we have found, students tend to ignore both the question and the answer.

actions. Pedagogica can set up "listeners," which are software agents that monitor views and other objects, and report at runtime on changes in their contents or properties. This enables the software, for instance, to react whenever a new organism is created, or when the student selects a gamete for fertilization. Since Pedagogica can communicate with the student through graphics and text, curriculum developers can use it to pose multiple-choice, survey, and open-response questions. It also controls the collection and storage of data, maintains and controls access to student records, and manages the registration and login functions.

Pedagogica is itself controlled by the third software layer, the *scripting layer*, which has the job of interpreting short scripts written in the Javascript language. These scripts implement the activities that the students actually see and interact with. They introduce the subject matter and configure the domain content engine appropriately (for instance, presenting two "parent" organisms with particular genotypes). They then monitor the students' actions, reacting to them as appropriate, and communicating occasionally with the students as they proceed with their investigations.

Processing Log Files to Support Research

The BioLogica hypermodels enabled not only just-in-time feedback to students as they worked through instructional activities, but also multilevel, longitudinal classroom-based research of student learning. The log files generated when students used BioLogica activities provided evidence about how BioLogica was used in classrooms as well as about students' developing knowledge and inquiry skills.

In order to do this, we first had to be sure that the data from which we were generating inferences and conclusions were accurate representations of what students were doing as they used BioLogica. This was accomplished by a series of verification and reduction steps, beginning with the comparison of log files with video of the computer screen recorded as students used the learning activities (Buckley et al., 2004). After we were certain that the log files accurately captured student actions and answers, we began the process of reducing them to forms and formats that were useful for reporting data to teachers and for importing data into a statistical package.

By creating activities with objects that automatically logged the same types of data each time they were used, we were able to structure the log files to support the data reduction algorithms used to make them useful for teachers and researchers. Each session generated hundreds of pages of raw log files, which would have been intractable were it not for the XML tags used to structure the output. Fig. 3.4 provides an excerpt depicting the data from a student crossing two dragons while looking at the wings pedigree.

As can be seen, data in this form are difficult to read, but can be used to verify accuracy. To provide a more accessible format, we processed the raw logs to produce a chronological report of the student's actions and answers. The example

```
Characteristic being observed: Trait: Wings
     </message>
    </action>
     <date> 2005.02.15.13.23.10&nbsp;02/15/05 I 13:23:10 </date>
     <message>
       Genotype of mother: Hh,SS,ww,Ll,Tt,pp,Ff,Aa,BB
     </message>
    </action>
     <date> 2005.02.15.13.23.10&nbsp;02/15/05 I 13:23:10 </date>
     <message>
       Genotype of father: Hh,SS,WW,Ll,Tt,p,F,a,B
     </message>
    </action>
     <date> 2005.02.15.13.23.10&nbsp;02/15/05 I 13:23:10 </date>
     <message>
       Generation of mother: 0
     </message>
    </action>
     <date> 2005.02.15.13.23.10&nbsp;02/15/05 I 13:23:10 </date>
     <message>
       Generation of father: 0
     </message>
    </action>
     <date> 2005.02.15.13.23.10&nbsp;02/15/05 I 13:23:10 </date>
     <message>
       number of offspring: 40
     </message>
    </action>
```

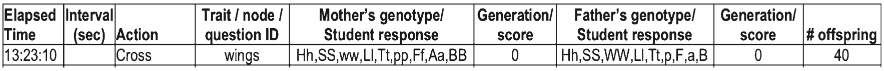
Fig. 3.4 Example from raw log file depicting data from one cross

shown in Fig. 3.5 is the same cross shown in the raw log. This format is much easier to read and compare to student actions.

The report provides information in a useful form concerning a single student working on a single activity. In order to compare student performances on the same activity, we integrated chronological reports across students to produce a summary report like the one shown in Fig. 3.6 in which each student's use of an activity is reported in a single row in a table. The excerpt shown includes data about one task (T4) during which the student successfully completed the task by performing three crosses in 11.3 min. The excerpt also shows the autoscoring of student performance (T4cat) and the student's use of the various tools available for completing the task.

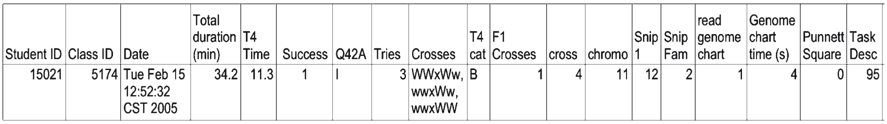
We also generated statistical reports that contained similar information about each student's interaction with each learning activity, but aggregated all of a student's sessions with one activity into one record. Statistical reports concatenated all answers and applied the autoscoring algorithms to the aggregated actions.

All of these various data reductions and report formats were useful in developing the algorithms for autoscoring and summarizing student actions, and for verifying the accuracy of the statistical records. As shown in Fig. 3.7, each raw XML file was



Elapsed Time	Interval (sec)	Action	Trait / node / question ID	Mother's genotype/ Student response	Generation/ score	Father's genotype/ Student response	Generation/ score	# offspring
13:23:10		Cross	wings	Hh,SS,ww,Ll,Tt,pp,Ff,Aa,BB	0	Hh,SS,WW,Ll,Tt,p,F,a,B	0	40

Fig. 3.5 Excerpt from Chronological Report depictingv the same cross shown earlier



Student ID	Class ID	Date	Total duration (min)	T4 Time	Success	Q42A	Tries	Crosses	T4 cat	F1 Crosses	cross	chromo	Snip 1	Snip Fam	read genome chart	Genome chart time (s)	Punnett Square	Task Desc
15021	5174	Tue Feb 15 12:52:32 CST 2005	34.2	11.3	1	I	3	WWxWw, wwxWw, wwxWW	B	1	4	11	12	2	1	4	0	95

Fig. 3.6 Excerpt from Summary Report summarizing student performance on Task 4

parsed to produce a concise report (the chronological file of student actions and answers). After carefully analyzing 6–12 concise reports, we created specifications for generating summary records from the raw XML files. To verify that the summary records were accurate, we compared the summary records of a different set of 6–12 logs to their corresponding concise reports. We found that students often took unpredictable actions that our original specifications did not adequately address. At this point, we corrected any errors in the specifications and in the summary report generator. We then analyzed the summary reports for students who had used an activity more than once, in order to develop specifications for how to aggregate their records to reflect the nature and extent of their interaction with a given activity. Pretests and posttests were treated as special cases because their purpose was to measure the students' knowledge and understanding at a given point in time, rather than to document their learning process.

We also created an implementation report generator that calculated for each student the gap (in days) between their last activity use and when they took the posttest[9] as well as what percentage of the core activities they had used. We used this report to calculate each student's BioLogica "dose." We averaged these by class as needed for statistical analysis. For a more complete description of the decisions that went into the development of summary and statistical records, please see (Buckley et al., 2004; Buckley, Gobert et al. (in press)).

[9] Early in the implementation we found that some teachers delayed administering the posttest until they were reminded that they would not receive their stipend for the year until we had their data.

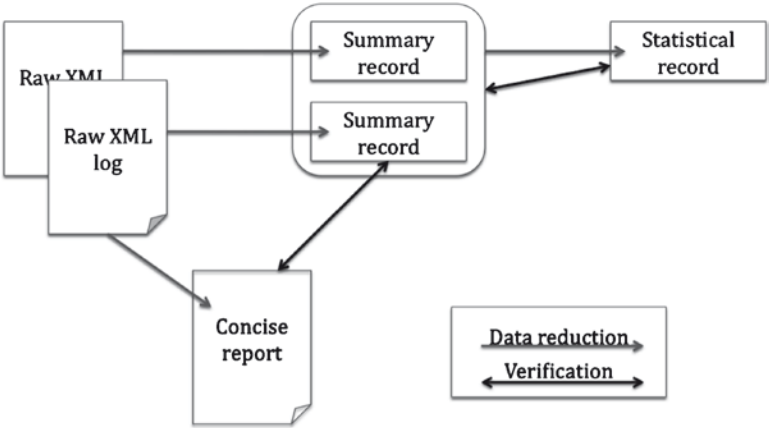


Fig. 3.7 Log file reductions and verification

Analyses Enabled by Logging Infrastructure

With the data available through the logging infrastructure, we conducted analyses that ranged from how many schools, classrooms and students were using our activities on any given day to what, for example, Student 32653 did when working on Task 2 in Monohybrid.

In the Modeling Across the Curriculum (MAC) project over the course of 3 years and three domains we collected log files generated by the work of 8,342 students in 411 classrooms in 61 schools taught by 146 teachers worldwide. For BioLogica alone we analyzed data from over 1,200 students in 54 classrooms. Due to the popularity of GenScope, we collected data from schools outside the group of recruited and supported member schools. Seventeen classrooms, comprising 573 students, were classified as Contributing Classrooms, which were classrooms whose teachers downloaded the software from the website, installed and used it with only minimal support from project staff. This very clearly demonstrated the scalability of the Modeling Across the Curriculum project.

We also implemented data quality criteria against which to judge the data included in our final analyses. For example, we excluded classrooms in which fewer than 50% of the class took both the pretest and posttest, reasoning that the data collected in such classrooms were unlikely to be sufficiently representative of the class as a whole. We also excluded logs that were less than 5 min in duration, on the basis that such a short time does not constitute a learning activity.

Learning Gains Versus Implementation Variables

We confined our exploration of student learning gains as a function of implementation variables to evidence that could be obtained by sifting through the massive amounts of log file data available. We considered time on task, but found that criterion to be a reliable indicator only in carefully controlled situations. For example, there are instances in which the student walks away from the computer without quitting the application and the log file isn't closed until the next time the computer is used (possibly as long as over the weekend). There are also the usual variations due to students' reading skills or stopping while a teacher talks; these are factors that we cannot tease out of our data.

Many iterations regarding how to conceptualize implementation variables in this context resulted in our adopting the following variables:

- The percentage of the core (nonoptional) activities used by a student for at least 5 min (referred to as *%Core*).
- Length of the intervention (referred to as *LOI*); calculated by subtracting the pretest date from the posttest date.
- Number of calendar days that elapsed between the last instructional activity used and the posttest date (referred to as PostTestGap).
- Time spent with instructional activities (referred to as *netLOI*); calculated by subtracting the PostTestGap from the LOI.

These variables were computed for each student, and their means and standard deviations were computed for each class. Subsequent analyses were done using these as predictor variables in order to better understand learning gains. Note that as the predictor variables are related to each other and therefore not independent, they were entered in the regression models individually.

In Table 3.1, we display the averages for the classes in which students achieved significant gains and those classes in which students did not.

From this table, it appears that successful classes used more activities over a longer period of time. Since students were clustered within classrooms, a traditional OLS regression approach would likely underestimate the standard errors associated with the regression coefficients and therefore increase the Type I error rate (Raudenbush & Bryk, 2002). We estimated this clustering effect by calculating the intraclass correlation coefficient (ICC) and found that approximately 50% of the variability in the posttest scores existed between classes. To account for this effect we constructed hierarchical linear models using data from 684 students clustered within 37 classes and including a random class effect.

Although some teachers taught more than one class, we had insufficient statistical power to estimate the variability between classrooms within teachers. Out of the 16 teachers who had their students interact with the hypermodels, four taught one class, six taught two classes, three taught three classes, and three taught four classes. Although there is a chance that the confounding of the variability among classes within teacher could contribute to biased estimates of the standard errors for the regression coefficients, we expect this effect to be negligible.

Table 3.1 BioLogica implementation data clustered by significant vs. nonsignificant gains

				Implementation variables				Learning measures				
BioLogica	#Classes	#Students	Yield	%Core	LOI	PostTestGap	Net LOI	Pretest	Posttest	Gain	Hake	Effect size
Overall	37	708	0.86	62	28	11	17	14.8	19.2	4.4	0.23	0.65
Significant gains	28	532	0.85	68	35	14	21	15.3	21.0	5.6	0.31	0.84
Nonsignificant gains	9	176	0.89	42	10	3	7	13.2	13.5	0.4	–0.02	0.05

Upon running multiple regressions, we found that the pretest scores accounted for 28.22% of the variance in the posttest scores. Holding pretest constant, *%core* activities accounted for an additional 10.83% of the variance in the posttest scores (see Table 3.2). Neither *PostTest Gap*, *LOI*, nor *net LOI* were significant predictors of posttest performance, nor did they account for any more of the variance in the posttest scores after holding the pretest scores constant.

Since all of our data is collected automatically and online, we are able to conduct large-scale research involving potentially tens of thousands of students.[10] The only issue limiting the scale of the research is the necessity for assuring adequate fidelity of implementation across multiple sites that are geographically remote from one another. The data reported so far pertains to such fidelity and involves classroom-grained variables. However, as we will see in the following section, hypermodels also support research at a very small grain size, but still involving very large numbers of students.

Performance Assessments: Information Inferred from Actions

We illustrate this type of research by describing one task in Monohybrid, a core activity that is crucial to understanding monohybrid inheritance and Mendelian genetics. Monohybrid Task 2 is an embedded, formative assessment that enables us to determine whether students:

1. Hold the naïve view of genetics that “like begets like,”
2. Can complete a Punnett square,
3. Know the allele combination that produces dragons with two legs, and
4. Can use the Punnett square to estimate the number of two-legged offspring that result from breeding a pair of two-legged parents.

In Task 2 (see beginning screen in Fig. 3.8), we present students with a pair of two-legged dragons represented by the half-filled square and circle at the top of the window. Note that the cross tool is grayed out, indicating that it is inactive. We ask students to predict how many legs their offspring will have when they use the cross tool (represented by an X) to breed the two parent dragons. As they have previously learned, they can use the cross to breed dragons: selecting it and dragging from one dragon to another of the opposite sex. This opens up a “family tree” representation with a predetermined number of offspring generated by the content engine according to the rules of inheritance. When students have made a prediction they are required to fill out a Punnett square[11] correctly, accessing a tutorial if

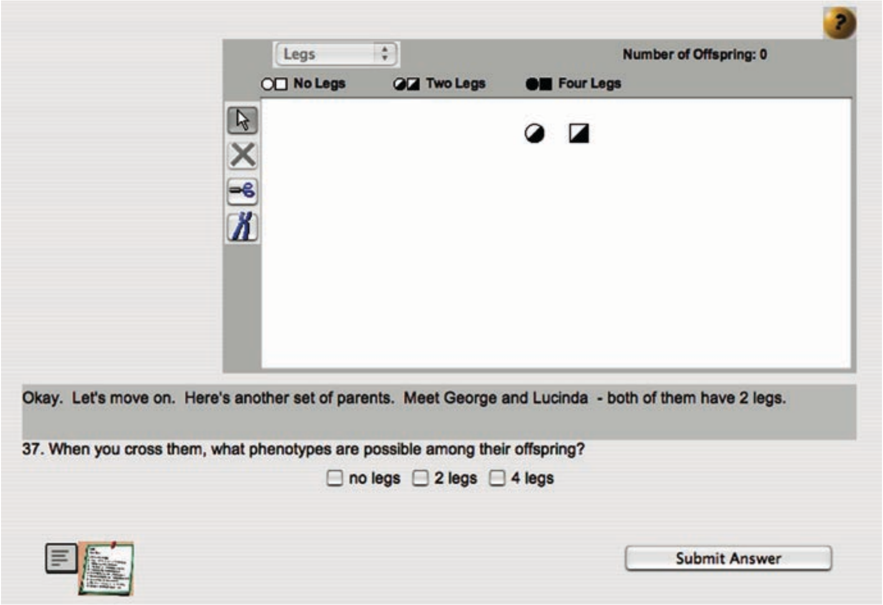
[10] The MAC Project ultimately involved over 8,000 students, spanning three different science disciplines.

[11] A Punnett square is a representation of a cross between two organisms. In its simplest form it consists of a two-by-two matrix representing all possible combinatorial outcomes involving the transmission of a single gene with two alleles from parents to offspring.

Table 3.2 Regression on implementation variables for BioLogica

	Model 1		Model 2		Model 3		Model 4		Model 5	
	Coeff. (s.e.)	Sig.	Coeff. (s.e.)	Sig.	Coeff. (s.e.)	Sig.	Coeff. (s.e.)	Sig.	Coeff. (s.e.)	Sig.
Intercept	12.53 (0.91)	<0.001	8.36 (1.09)	<0.001	12.83 (1.11)	<0.001	12.77 (0.94)	<0.001	12.35 (0.96)	<0.001
Pretest scores	0.45 (0.04)	<0.001	0.43 (0.04)	<0.001	0.45 (0.04)	<0.001	0.45 (0.04)	<0.001	0.45 (0.04)	<0.001
% Core			0.07 (0.01)	<0.001						
LOI					−0.01 (0.02)	0.651				
PostTestGap							−0.02 (0.02)	0.349		
Net LOI									0.00 (0.00)	0.553
R^2 (%)	28.22		39.05		28.22		28.22		28.22	

Dependent variable: posttest scores on BioLogica assessment
Coeff. coefficient, *Sig.* significance, *s.e.* standard error



Okay. Let's move on. Here's another set of parents. Meet George and Lucinda - both of them have 2 legs.

37. When you cross them, what phenotypes are possible among their offspring?

□ no legs □ 2 legs □ 4 legs

Fig. 3.8 Task 2 opening screen. The half-filled circle and square represent, respectively, a two-legged female and a two-legged male dragon. The icons along the *left side* represent, from *top to bottom*, the selection tool, the cross tool, the snip tool (used to delete unwanted organisms), and the chromosome-observing tool

needed. Students are then asked to identify which cells of the completed Punnett square correspond to offspring with two legs and to use that information to estimate the number of two-legged offspring resulting from the crossing of two-legged parents. At that point the cross tool is enabled and students are directed to use it to check their predictions against model-generated "experimental" data.

As students proceed through this task the hypermodel logs their prediction (multiple selections are possible), how many times they fill in the Punnett square and the alleles they enter for each attempt, how many times they try to select the correct alleles in the Punnett square and the alleles in the cells they select, their estimate of the offspring that will be produced, and whether they thought their prediction was borne out by the data generated when they crossed the dragons. This gives us considerable data regarding the nature of their initial predictions and their procedural knowledge for various inquiry skills, including interpreting their data.

In principle, these data enable us to provide information to teachers about their students' performances on this task. However, since the data analysis was not complete at the time of implementation, such reports were not actually produced in real time, though they were eventually made available to researchers. Table 3.3 provides an example of how the top level of such a report might appear. It shows the distribution of students by response or number of tries. Note that in the prediction column both the naïve and correct responses are marked. What this table tells the researchers is

Table 3.3 Report designed for teachers and researchers

	Prediction			Punnett square completion		Punnett square selection		Estimate offspring	
Time range (min)	Q37R	#Students	%Students	Tries	#Students	Tries	#Students	Q38R	#Students
2.1–6.5	A	3	1%	1	205	1	193	A	10
	B[a]	64	25%	2	9	2	5	B	14
	C	10	4%	3	3	3	29	C[b]	197
	CA	1	0%	>3	25	4	3	D	14
	CB	13	5%			5	6	E	6
	CBA[b]	127	50%			>5	11		
	BA	34	13%						
	Total answers	252			242		247		241
	Item difficulty		0.50		0.85		0.78		0.82

[a] Naïve conception
[b] Correct answer

that the prediction question has a difficulty level of 0.50 since 50% of the students got it right, but that 25% of them still hold the naïve conception that "like begets like." The remainder of the task shows us that a high percentage of the students have mastered the procedural skills necessary for the tasks that follow this one. A teacher given such a report on the web could double-click on the responses or number of tries indicating students having difficulty and get a list of those students for follow-up intervention.

For teachers' grading and our statistical analyses, student performances were also autoscored according to a rubric that allotted points for correct answers and completing the Punnett squares tasks with few attempts. We examined the relationship between performance on Task 2 (*Predict*) as represented by the Task 2 score and performance on the other Monohybrid tasks (3 and 4) and on the posttest (holding the pretest constant). Task 3 (*Produce*) asks "what parental genotypes would result in all the offspring having two legs?", requiring students to reason from effect to cause. Students must set the genotypes of the parents, breed them, and check the result. Task 4 (*Skip*) is an unscaffolded transfer task that asks the students to demonstrate the genetic mechanism that causes traits to appear to skip a generation, requiring that students reason over three generations.

Based on data from students who used the Monohybrid activity in 2005–2006 (including those who took both the pre- and posttests), we conducted correlations and found that all three tasks were weakly but significantly correlated with total pretest scores; additionally, Task 2 scores were significantly correlated with both Task 3 and Task 4 scores (Table 3.4). Further analyses are underway to examine the relationships between these three tasks (Buckley et al., in press).

Table 3.4 Correlations among total pretest scores and task scores

	Total score pretest	Task 2 – predict	Task 3 – produce	Task 4 – skip
Total score pretest	1.00	–	–	–
Task 2 – predict	0.29[a]	1.00	–	–
Task 3 – produce	0.29[a]	0.56[a]	1.00	–
Task 4 – skip	0.24[a]	0.34[a]	0.31[a]	1.00

[a] Correlation is significant at the 0.01 level (two-tailed)

As shown in the above analyses, students' interactions with hypermodels provide fine-grained data that can be analyzed so that researchers can examine the inquiry performances of individual students in a sophisticated way, as well as do this on a large scale. Additionally, teacher reports can both summarize and describe the performances of their students, and data can be aggregated to provide large-scale measures for administrators and policy makers.

Limitations of This Work

One of the goals of the MAC project was to provide reports on students' progress to the teachers in real time. We succeeded in doing this for the embedded question-and-answer assessments, however, the analyses and autoscoring of students' performance data was not complete in time for us to provide performance assessment data to teachers. For example, we were not able to point out to them, the students who consistently filled out Punnett Squares incorrectly, or made aimless crosses in the Invisible Dragons task. This was unfortunate because, as we have seen, performance data from specific tasks such as these turned out to be predictive of students' posttest learning. There remains an empirical question as to the level of teacher professional development that would have been necessary in order for teachers to benefit from these performance reports. Some potential barriers and/or difficulties here are: implementing inquiry-based activities into instruction (de Jong et al., 2005), and tailoring instruction for individual students (Fadel et al., 2007). Providing teachers with inquiry tasks that are scaffolded for student use, as we did with BioLogica, could potentially alleviate implementation difficulties – in fact, these were designed for this purpose. Simply providing performance assessment reports about students' learning is likely not enough support for teachers to tailor instruction for individual students, rather, these provide a necessary but not sufficient condition for teachers to determine what individual students need. Below we describe some recent work that may address these difficulties.

One of the most frustrating limitations of the BioLogica software initially was that the computer did not store and recall from one session to the next where the student was in an activity. Thus, at the end of a class period students who had not completed an activity would not be able to save it, and would be forced to start it at the beginning the next time they logged on. During the course of the MAC Project we were able to provide a partial solution to this problem by breaking the activities up into short, semantically meaningful chunks and enabling students to navigate directly between these. But the inability to "save state" was nevertheless a constant source of annoyance to students and teachers alike. We continue to work on this issue in our current research.

Finally, there is the issue of scalability. As a standalone application, BioLogica is relatively easy to install,[12] though it does require the prior installation of Java 1.4 or higher in order to run. However, in this mode the user receives few of the benefits of the data logging that have occupied so much of the foregoing discussion. True, the versions of BioLogica activities that we have put up on the server do create reports automatically whenever the user quits them, but these reports must be separately saved and retrieved, and their results are difficult to aggregate across multiple students or multiple activities. One reason for this is that when it runs as a standalone activity, BioLogica cannot take advantage of a server to use as a central repository for data.

Another, more fundamental, problem arises from the fact that in order to produce aggregated data across students, BioLogica must have access to those students' identities. If we are to produce a report that summarizes how an entire Biology class is doing, for instance, the students in that class must be recognizable entities to the software that produces and analyzes the data. This implies that each student must have a unique account that is accessible to the software. If the reports are to be used to give the students a grade, then they must be continually backed up and protected against vandalism and fraud. The difficulty of supporting such a complex data storage and retrieval process exceeds the resources of all but the most technologically advanced schools.

Next Steps

One of the major goals of our future work will be to "close the loop" by giving teachers the kind of detailed and insightful information that the MAC research team was only able to glean after months of careful analysis. Researchers at the Concord Consortium have started work on a 5-year project, funded by the National Science Foundation, called "Logging Opportunities in Online Programs for Science,"[13] which is creating timely, valid, and actionable reports to teachers based on logs of student actions generated in the course of using online curriculum materials. These reports will enable teachers to make data-informed decisions about alternative teaching strategies. An important goal of future research is to observe how teachers integrate such reports into their practice.

[12] Versions for Windows and MAC OS computers are available for free download at http://mac.concord.org/downloads/.

[13] NSF Project # 0733299.

One of the unavoidable consequences of moving from an open-ended, constructivist technology, as typified by GenScope, to the more tightly structured and scaffolded BioLogica activities is that, in their present form at least, the latter cannot easily adapt to individual students. Some students demand more structure than others, some are intimidated by a challenge that requires them to explore new territory, while others become bored when told step by step what to do. One solution to this problem would be to create several alternative versions of each activity and to leave it up to the teacher and/or the student to decide on which one to choose, or to modify the activity in real time as new performance is acquired. Such an approach might ultimately result in the creation of a technology capable of dynamically customizing itself to suit students' individual needs and desires. For instance, one can imagine an adaptive activity that starts out highly structured and then strips off its scaffolding progressively in response to student actions indicating increased understanding and self-confidence. The Science ASSISTments projects are developing assessments for science inquiry that both instruct and assess[14] (Gobert, Heffernan, Ruiz, & Kim, 2007; Gobert, Heffernan, Koedinger, & Beck, 2008). The assessment modules will examine students' log files to detect such "buggy" inquiry behavior as repeating trials or moving further from the target goal. The system is being designed to provide teachers with feedback as to their students' inquiry skills, as aligned to the NSES inquiry standards (National Research Council, 1996), as well as to intervene and tutor students on inquiry skills in real time.

Before such advances can be practical in educational terms, however, the process of creating sound educational activities must be simplified. It will never be trivial to design educationally effective curriculum materials, regardless of their technological basis, but the development of hypermodels such as those we have described requires a knowledge of programming far exceeding that of most teachers or curriculum developers. This severely limits the pool of potential designers of hypermodel-based curriculum. With this limitation in mind, the Concord Consortium has been experimenting with several alternative authoring environments that we expect will someday replace the cumbersome and obsolete Activity Construction Editor that forms part of Pedagogica. For example, the Technology Enhanced Learning in Science (TELS) Center, a joint project between Concord Consortium and the University of California at Berkeley, has worked on a Scalable Architecture for Interactive Learning (SAIL) a software environment that, when fully implemented, will support reuse and adaptation of interoperable components, making it possible to implement interactive curriculum and assessments by working at a very high level (Slotta, 2010).

The Concord Consortium's Molecular Workbench (MW) tool (http://mw.concord.org) is an example of an easily authorable environment. MW models can be embedded in a browser-like environment that links them to other interface objects like text boxes and buttons. A simple but increasingly powerful scripting language

[14] The terms "Assistments" was coined by Ken Koedinger for the Math Assistments program that was developed by him and Neil Heffernan.

enables an activity author to configure various aspects of the MW model, and offers a modest degree of runtime control. Another authoring environment produced by Concord Consortium, called DIY (for “Do It Yourself”), generalizes similar capabilities to other educational affordances, such as probeware and third-party interactive applets. We anticipate that all these different technologies will be integrated within a few years, making it possible for nonprogrammers to create complex and engaging curriculum activities and interactive assessments.

Lastly, none of the technologies discussed in this paper provide direct support to students for collaborative work. It would have been very helpful in the MAC Project, for instance, if students could have shared their activities as they were working on them. This would have enabled them to show their work to each other and to ask each other for help when needed. It would also have been useful if the teacher could have called up an individual student’s work, either to go over it with that student or to show it to the class as a whole. We have been working on ways to make all the models and other objects that we use “serializable” – that is, to translate them into a set of instructions that will enable a computer to recreate them in their current state. Once that goal is accomplished, a student who wishes to share any of these objects, either with another student, with the teacher, or with the entire class, will be able to do so simply by transmitting instructions that will enable the recipient’s computer to recreate the originator’s model.

This brings up the intriguing possibility that students could be prompted by the computer to ask for help as they progress through an activity. A classroom server would keep track of each student’s or group’s progress and could link up students, either automatically or under control of the teacher, in pedagogically productive ways. Any communication or object sharing initiated by students using this technology would itself be monitored and could be used either by the teacher or as input to a research project concerned with the effects of technology-mediated collaboration.

The lack of technological infrastructure in most schools, as exemplified in the limitations section above, poses the major obstacle to using the hypermodel approach to instruction and assessment and in particular, scaling up technologies such as ours and others mentioned here. The difficulties in scaling up should come as no surprise; it is as though we have built a high performance automobile and demonstrated that it can go 80 miles/h but there are as yet very few paved roads on which to drive it. The situation is by no means hopeless, but the solution may take a while – recall that the interstate highway system was not launched until nearly 50 years after the introduction of the first mass-produced cars. We can only hope that the significant potential impact of computers in education will be achieved in less time than that!

Appendix: Description of the BioLogica Activities

1. *Introduction*. The *Introduction* activity, described in detail above, enables the students to develop a familiarity with the software as well as with the basic concepts of genetics. It provides an initial guiding question: What do dragons look like and why?

2. *Rules*. The *Rules* activity introduces students to dominance relationships among alleles while helping them learn the rules of inheritance in dragons in order to understand how genes affect appearance. The activity is in three parts: *Dominant and Recessive Relationships Among Alleles* focuses on which alleles are dominant and therefore mask the presence of other alleles for a gene. The students identify all possible combinations of alleles that produce a particular trait in an organism. *Some Traits are X-Linked* focuses on genes that are located on the X chromosome. Students investigate the impact of different allele combinations for X-linked genes. *Color and Fatal Combinations* examines polygenicity and the affect of lethal alleles using the two Color genes of BioLogica's dragons. Students explore what happens when more than one gene contributes to a single characteristic and learn about an allelic combination that is lethal.
3. *Meiosis*. This activity builds on *Introduction and Rules* and requires students to use what they learned about dragon genotypes and phenotypes to complete a series of challenges in order to address why members of a family do not always look alike. The first subactivity, *Introduction to Meiosis,* focuses on learning to use the meiosis model of BioLogica, understanding how chromosomes and alleles participate in meiosis, and linking the meiosis model's representations of gametes and chromosomes with the representations of those objects in the chromosome model, introduced previously. The meiosis model simulates the process of meiosis in a fashion similar to the diagrams of the phases of meiosis found in textbooks. The second subactivity, called *Designer Dragons,* challenges the students to create specific offspring bvy examining the chromosomes in the gametes of each parent and selecting those that will produce the desired phenotype in the offspring.
4. *Horns Dilemma*. This optional activity may be used as an enrichment experience for students who are looking for a challenge or as an assessment of students' models of meiosis and fertilization. It focuses on the inheritance of recessive traits, posing the question, "Can two horned parents have a hornless baby?" and challenging students to produce a hornless dragon from two parents that have horns, using knowledge gained in the previous activities.
5. *Monohybrid.* This activity is at the core of the BioLogica curriculum. It is here that students encounter for the first time the intergenerational consequences of the genetic processes they have been studying. The activity is in four parts. *Introduction to Pedigrees* teaches students how to use BioLogica's pedigree level to create and analyze pedigrees. *Pedigrees and Punnett Squares, Oh My!* makes connections between independent assortment in meiosis, random selection in fertilization, predictions made with Punnett squares, and breeding experiments that use the pedigree level tools. *Studying Patterns of Inheritance Using Pedigrees and Punnett Squares* guides students' reasoning as they determine probabilities for the inheritance of particular traits. Part four of the activity, *An Inheritance Puzzle,* challenges

students to put everything they have learned in the previous three parts together in order to solve a puzzle.

6. *X-linkage*. This is another core activity in the sequence. It includes three sections designed to address the central question: "What difference does it make if a gene is located on the X Chromosome?" *Introduction to Genes that are part of the X Chromosome* reviews how a gene is inherited when it is part of the X-chromosome. *X-Linked Traits* uses the fire-breathing gene in dragons to demonstrate the inheritance patterns of sex-linked traits from one generation to the next. *Determining if a Characteristic is X-linked* focuses on pedigree analysis as a tool for discriminating between autosomal and X-linked inheritance.
7. *Mutations*. The driving question for this activity is: "What happens when you change the DNA?" Students are introduced to mutations through the appearance of a novel trait in a pedigree. They then explore the role of DNA in mutations, modifying the base pair sequences of particular dragon alleles and examining the impact of these newly created alleles on the appearance of a dragon.
8. *Mutations 2*. This optional activity poses the question: "How are mutations inherited?" It builds on Mutations and Monohybrid by enabling students to investigate how a novel allele is inherited by offspring and its affect on the inheritance of the associated phenotype. It also gives the students more practice in using Punnett squares to determine the probability of inheriting a mutated trait.
9. *Dihybrid Cross*. This activity asks the question: "What is the likelihood that two traits will be inherited together?" It focuses on the inheritance patterns for two traits at a time, and examines the differences that occur when the genes for those traits are parts of the same chromosome or parts of different chromosomes.
10. *Scales*. This is another optional activity that challenges students to investigate the mode of inheritance of a new trait, posing a series of challenges designed to teach students to reason like geneticists.
11. *Plates*. This optional activity introduces another novel trait: scaly plates on the back of the dragons' neck. Students are challenged to determine the inheritance pattern of this new trait (which is X-linked and incompletely dominant) and the location of its gene by a process that approximates the reasoning of professional geneticists.
12. *Invisible Dragons*. Invisible Dragons presents a difficult problem for the students to solve using all the techniques they have learned throughout this series of activities. They must figure out the genetic make-up of two invisible dragons, one male and one female. They may cross the parent dragons, including making backcrosses (crosses between an offspring and parent), and view any of the resulting offspring. Their challenge is to deduce the parental genotypes by observation of the phenotypes of the offspring, using as few crosses as possible.

References

Aulls, M. W. (2002). The contributions of co-occurring forms of classroom discourse and academic activities to curriculum events and instruction. *Journal of Educational Psychology, 94*(3), 520–538.

Blais, D. M. (1988). Constructivism: A theoretical revolution in teaching. *Journal of Developmental Education, 11*(3), 2–7.

Buckley, B., Gobert, J., Horwitz, P. & O'Dwyer, L. (in press). Looking inside the black box: Assessing model-based learning and inquiry in biologica. *International Journal of Learning Technologies.*

Buckley, B. C., Gobert, J. D., Kindfield, A., Horwitz, P., Tinker, R., Gerlits, B., et al. (2004). Model-based teaching and learning with BioLogica™: What do they learn? How do they learn? How do we know? *Journal of Science Education and Technology, 13*(1), 23–41.

Clement, J. (1989). Learning via model construction and criticism: Protocol evidence on sources of creativity in science. In J. Glover, R. Ronning & C. Reynolds (Eds.), *Handbook of creativity: Assessment, theory and research*. NY: Plenum.

de Jong, T., Beishuizenm, J., Hulshof, C., Prins, F., van Rijn, H., van Someren, M., et al. (2005). Determinants of discovery learning in a complex simulation learning environment. In P. Gardenfors & P. Johansson (Eds.), *Cognition, education, and communication technology*. Mawah, NJ: Erlbaum.

Driver, R., & Oldham, V. (1986). A constructivist approach to curriculum development in science. *Studies in Science Education, 13*, 105–122.

Ericsson, K. A., & Simon, H. (1980). Verbal reports as data. *Psychological Review, 87*, 215–251.

Fadel, C., Honey, M., & Pasnick, S. (2007). Assessment in the age of innovation. *Education Week, 26*(38), 34–40.

Gobert, J. (2005). Leveraging technology and cognitive theory on visualization to promote students' science learning and literacy. In J. Gilbert (Ed.), *Visualization in science education* (pp. 73–90). Dordrecht, The Netherlands: Springer-Verlag.

Gobert, J., Heffernan, N., Ruiz, C, & Kim, R. (2007). *AMI: ASSISTments Meets Inquiry*. Proposal NSF-DRL# 0733286 funded by the National Science Foundation.

Gobert, J., Heffernan, N., Koedinger, K., & Beck, J. (2008). ASSISTments Meets Science Learning (AMSL). Proposal (R305A090170) funded March 1, 2010 by the US Department of Education.

Gobert, J. D., & Buckley, B. C. (2000). Introduction to model-based teaching and learning in science education. *International Journal of Science Education, 22*(9), 891–894.

Gobert, J., Buckley, B., & Clarke, J. E. (2004). *Scaffolding model-based reasoning: Representations, cognitive affordances, and learning outcomes.* Paper presented at the American Educational Research Association, San Diego, CA.

Gobert, J., O'Dwyer, L., Horwitz, P., Buckley, B., Levy, S.T. & Wilensky, U. (revisions submitted). *Examining the relationship between students' epistemologies of models and conceptual learning in three science domains: Biology, Physics, & Chemistry.* International Journal of Science Education.

Hickey, D. T., Kindfield, A. C. H., Horwitz, P., & Christie, M. A. T. (1999). Advancing educational theory by enhancing practice in a technology-supported genetics learning environment. *Journal of Education, 181*(2), 25–55.

Hickey, D. T., Kindfield, A. C. H., Horwitz, P., & Christie, M. A. T. (2003). Integrating curriculum, instruction, assessment, and evaluation in a technology-supported genetics learning environment. *American Educational Research Journal, 40*(2), 495–538.

Hickey, D. T., Kruger, A.C., & Zuiker, S. (2003). *Design experimentation with multiple perspectives: The GenScope assessment project*. Presented at the Annual Meeting of the American Educational Research Association, Chicago, IL.

Horwitz, P. (1999). Designing Computer Models that Teach. In N. Roberts & W. Feurzeig (Eds.), *Computer Modeling and Simulation in Pre-College Science Education* (pp. 179–196): Springer Verlag.

Horwitz, P., & Christie, M. (2000). Computer-based manipulatives for teaching scientific reasoning: An example. In M. J. Jacobson & R. B. Kozma (Eds.), *Innovations in science and mathematics education: Advanced designs for technologies of learning*. Hillsdale, NJ: Lawrence Erlbaum & Associates.

Horwitz, P., Neumann, E., & Schwartz, J. (1996). Teaching science at multiple levels: The GenScope program. *Communications of the ACM, 39*(8), 179–196.

Horwitz, P., Schwartz, J., & Neumann, E. (1998). *Implementation and evaluation of the GenScope™ learning environment: Issues, solutions, and results.* Paper presented at the Third Annual International Conference of the Learning Sciences. Charlottesville, VA: Association for the Advancement of Computers in Education.

Justi, R. S., & Gilbert, J. K. (2002). Science teachers' knowledge about and attitudes towards the use of models and modeling in learning science. *International Journal of Science Education, 24*(12), 1273–1292.

Kinzie, M. B., Sullivan, H. J., & Berdel, R. L. (1988). Learner control and achievement in science computer assisted instruction. *Journal of Educational Psychology, 80*(3), 299–303.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction dose not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist, 41*(2), 75–86.

Lederman, N. G. (2006). Syntax of nature of science within inquiry and science instruction. In L. B. Flick & N. G. Lederman (Eds.), *Scientific inquiry and nature of science: Implications for teaching, learning, and teacher education*. Dordrecht, Netherlands: Springer.

Magoon, A. J. (1977). Constructivist approaches in educational research. *Review of Educational Research, 47*(4), 651–693.

National Research Council. (1996). *Inquiry and the national science education standards*. Washington, DC: National Academy Press.

Nisbett, R., & DeCamp Wilson, T. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review, 84*(3), 231–259.

Papert, S. (1980). *Mindstorms: Children computers and powerful ideas*. New York: Basic Books.

Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods* (2nd ed.). Newbury Park, CA: Sage.

Slotta, J.D. (2010). Evolving the classroom of the future: The interplay between pedagogy, technology and community. In Mäkitalo-Siegl, K., Zottmann, J., Kaplan, F., & Fischer, F. (Eds.). *Classroom of the Future: Orchestrating collaborative spaces*. Rotterdam, the Netherlands: Sense.

Steinberg, E. R. (1977). Review of student control in computer-assisted instruction. *Journal of Computer-Based Instruction, 3*, 84–90.

Suppes, P. (1969). Computer-assisted instruction: An overview of operations and problems. Information processing. In A. J. H. Morrell (Ed.), *Information processing proceedings of IFIP congress*. Amsterdam: North-Holland.

White, B. Y., & Frederiksen, J. R. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction, 16*(1), 3–118.