Int J Artif Intell Educ (2016) 26:1033–1068
DOI 10.1007/s40593-015-0093-5

ARTICLE

Learning Science by Constructing Models: Can Dragoon Increase Learning without Increasing the Time Required?

Kurt VanLehn[1] · Greg Chung[2] · Sachin Grover[1] · Ayesha Madni[2] · Jon Wetzel[1]

Published online: 21 January 2016

Abstract A common hypothesis is that students will more deeply understand dynamic systems and other complex phenomena if they construct computational models of them. Attempts to demonstrate the advantages of model construction have been stymied by the long time required for students to acquire skill in model construction. In order to make model construction a feasible vehicle for science instruction, the Dragoon system combined three simplifications: (1) a simple notation for models of dynamic systems, (2) a step-based tutoring system, and (3) problems that described the model to be constructed as well as the system represented by the model. In order to test whether these simplifications reduced the time for learning how to construct models while preserving the benefits of model construction over baseline instruction, three classroom studies were conducted. All studies were experiments, in that they compared classes using Dragoon to classes learning the same material without Dragoon. However, as classroom studies, they could not tightly control all sources of variation. The first study produced null results, but it compared learning across just one class period. The second study in 4 high school science classes showed that instruction based on Dragoon cost only one extra class period (about 50 min) out of 4 class periods and was more effective than the same content taught without Dragoon. A third study in 3 more high school science classes, where 2 Dragoon classes and 1 non-Dragoon class met for the same number of class periods, showed that Dragoon was more effective than the same content taught without Dragoon. The effect sizes were moderately large on both an open response test ($d = 1.00$) and a concept mapping task ($d = 0.49$). Thus, it

✉ Kurt VanLehn
Kurt.Vanlehn@asu.edu

1 Computing, Informatics and Decision Science Engineering, Arizona State University, Tempe, AZ, USA

2 Center for Research on Evaluation, Standards and Student Testing, University of California at Los Angeles, Los Angeles, CA, USA

appears that our efforts have simplified model construction to the point that it can be used in science instruction with no additional class time needed, and yet it still seems to be more effective than the same instruction done without model construction.

Keywords Intelligent tutoring system · Model construction · High school science instruction

Introduction

The main focus of this research is finding out if teaching high school students to construct models of dynamic systems can be done in a feasibly short period of class time and if so, whether such instruction helps them learn more about the dynamic systems than they would otherwise have learned during that time. The research focuses on modeling because modeling has become a central, valued practice in the emerging national standards of the USA. In both the Common Core State Standards for Mathematics (CCSSO 2011) and the Next Generation Science Standards (NGSS 2013; NRC 2012), modeling is one of 7 practices that are considered essential for high school students to master. However, "model" and "modeling" can denote many things. The first subsection below discusses types of models and why we chose to focus on just one type. The second subsection discusses types of educational activities that involve modeling and why we chose to focus on just one type. Subsequent subsections review the relevant literature and finally pose the research question in full detail.

Types of Models

In education, a "model" can denote many things. Although Collins and Ferguson (1993) present an impressively complete list of types of models, their list can be simplified into three basic categories:

- Models that are expressed in an informal notation and their implications are derived informally. Examples are compare-and-contrast and taxonomies.
- Models that are expressed in a formal notation, and their implications are derived manually. Examples are a concept map or a set of mathematical equations written on paper.
- Models that are expressed in a formal notation, and their implications are derived by a computer. Examples are spreadsheets, NetLogo programs, and equations entered into a graphing calculator.

This research is focused exclusively on the third type of model, which uses computers to calculate the model's predictions. Computer-based modeling is increasing in the professions, so this particular form of modeling is arguably an increasingly important practice to learn. Moreover, the standards place special emphasis on computational modeling. The Next Generation Science Standards discussion of modeling often mentions computational models and ends up emphasizing the need for computational modelling tools (pg. 59):

> Curricula will need to stress the role of models explicitly and provide students with modeling tools (e.g., Model-It, agent-based modeling such as NetLogo, spreadsheet models), so that students come to value this core practice and develop a level of facility in constructing and applying appropriate models.

Both the Next Generation Science Standards and the Common Core State Standards for Mathematics distinguish between learning (1) skills and practices, such as computational modeling, and (2) domain concepts, principles and facts. Although the Dragoon project is addressing both instructional objectives, the work on teaching students how to construct models is reported in another publication (VanLehn et al. 2016). This paper covers only the work on using Dragoon to teach science.

Compared to other types of models, there are both advantages and disadvantages to using computational models for teaching science. On the one hand, an advantage of expressing the model in a formal language like mathematics or directed graphs is that the language is designed to make certain kinds of inferences easy and mechanical. Thus, if we follow Chi et al. (1994) in assessing deep understanding by asking students to make inferences that construct information that is not presented in the instruction, then formal models should facilitate such inference-making by the student. For example, telling a student that "the current through a two-terminal device is determined by the device's resistance and the voltage across it" does not allow the student to make as many inferences as telling the student that "V = I*R, where V is voltage, I is current and R is resistance, all relative to the two-terminal device." On the other hand, a disadvantage of formal models, regardless of whether they are interpreted computationally or by hand, is that they are expressed in a formal language that may be difficult for students to understand. Although the operators in the language might pose difficulties (e.g., in a concept map, what does the arrow between two concepts mean?), a more insidious factor is the tendency to use short names for components and quantities in the system, which makes it all too easy for students to lose track of what the names denote and leads them to manipulate the model in nonsensical ways (VanLehn 2013, section 7.2). For example, when interpreting V = I*R, students may lose track of which voltage V refers to. Lastly, when comparing the two types of formal models (human vs. computer calculation), a key advantage of the computational models is that much more complex systems can be analyzed feasibly if the computer does the calculations rather than having students do them. These observations suggest that computational models are most advantageous for learning science when the systems to be understood are complex and when the students already have fluency in the formal language of the model or can attain fluency easily.

Perhaps the most common types of computational modeling systems used in education are graphing calculators, spreadsheets, agent-based models and system dynamics models. The latter two require some explanation. An agent-based model is essentially a set of interacting computer programs, one for each type of agent. For instance, if one is modeling wolf-sheep population dynamics, one might have one program for the wolves and one for the sheep. NetLogo is currently the major language for agent-based modeling (ccl.northwestern.edu/netlogo/).

System dynamics models appear in two forms. When used in university engineering courses, a system dynamics model is expressed as coupled differential equations that are solved using MATLAB, Mathematica or similar systems. When used in social science, ecology, business and other university courses, a system dynamics model is expressed in a graphical notation called a stock-and-flow diagram. The notation, which appeared first in 1983 in Stella (http://www.iseesystems.com/), represents quantities as nodes in a directed graph. For instance, a wolf-sheep population model would have one node representing the number of wolves, a second node representing the number of sheep and other nodes representing other quantities.

For high school science instruction, graphical system dynamics languages seem to offer the best tradeoff along a scale of computational abstraction. Agent-based models are the most concrete in that they have visible objects that represent individuals in the system, such as a wolf, a sheep or a patch of grass. Such a model has a subprogram that represents the event of a wolf-sheep meeting that results in the wolf killing the sheep. This concreteness makes it easy to see what a given model represents and to envision or design a model for a given system. On the other hand, agent-based models require learning how to program NetLogo or some other language so that all those concrete details can be expressed. Thus, learning how to create an agent-based model is hard, but seeing what it represents is easy. This is just as one would expect of a more concrete model.

On the other end of the scale, the equations entered into graphing calculators are abstract but also familiar, because algebraic notation is taught in high school algebra classes. Once one has envisioned or designed an equation, entering it into a graphing calculator is easy. However, it is often difficult for students to see what the terms in an equation represent (Corbett et al. 2006). For instance, the frequency of fatal meetings between wolves and sheep would be expressed by a product of the number of wolves, the number of sheep and a parameter. It is not obvious why that product represents the number of predation events. Thus, constructing equations is easy, but it is hard to see what a given equation and its parts represent and to design equations to model a given system.

Graphical system dynamics models are midway between concrete, agent-based models and abstract, equation-based models. For instance, instead of an anonymous subexpression to represent predation, like the product mentioned above, a well-built graphical system dynamics model would have a node with a long name, such as “number of fatal wolf-sheep meetings” or “number of predation events.” These and other practices make it easier for students to see what a given model represents and to design a model for a given system (Löhner et al. 2003). However, the graphical notation is less familiar than the algebraic one, so it may take longer for students to learn how to enter graphical models than equation-based models.

Although we chose to focus on graphical system dynamics models because we believe they offer a good tradeoff between two sources of difficulty, an ideal curriculum would probably include several types of modeling languages because each has its own advantages, and students preparing to enter our computationally rich world would probably benefit from mastering all of them. However, teaching multiple modeling languages addresses computational modeling *as a practice*, which is a different instructional objective than the one addressed here. This paper is concerned only with using computational modeling to teach scientific principles, systems and facts. For that instructional objective, it may suffice to have students attain fluency in just one modelling language.

Types of Modeling Activities

Model construction and model exploration are the two most common educational activities done with computational models (Alessi 2000b; Stratford 1997). *Model construction* involves writing out a model in a formal notation, running the model on the computer, checking whether the predictions make sense and match any data on hand, and perhaps repeating this process in order to improve the model or its predictions. Like programming a computer, model construction requires fluency in the formal notation was well as good problem solving skills. Figure 1, from the Common Core State Standards for Mathematics, is a diagram of the model construction process.

In contrast, *model exploration* refers to manipulating aspects of a given computational model and observing the changes in its predictions. For instance, computational modeling systems typically let the user manipulate a slider in order to change the value of a model's parameter, and this causes an instantaneous change in the gauges or plots that display the model's predictions. For model exploration, it is not necessary for the student to see the model or to understand the modeling language. When the model is hidden from the user, and all the user sees are the controls (e.g., sliders) and prediction displays (e.g., graphs), the model is typically referred to as a *simulation*. Model exploration is the second major type of model-based learning activity.

According to the ICAP framework (Chi 2009; Chi and Wylie 2014), model construction should be more effective than model exploration, as the former is constructive (the "C" of ICAP) and the model exploration is often merely active (the "A" of ICAP). Although we know of no studies testing this hypothesis with graphical system dynamics models, Hashem and Midouser (2010, 2011) compared model construction and model exploration with NetLogo. More specifically, they focused on two NetLogo models exemplifying emergence, an important concept in complex systems. They compared a 3 h lesson where students *explored* the two models to 48 h of instruction in NetLogo programming culminating in students *constructing* the two models. The model construction group learned more about emergence than the model exploration group. The difference in mean post-test scores was reliable and moderately large. This is consistent with the common-sense hypothesis that model construction takes longer than model exploration, but fosters deeper learning.

We decided to focus on model construction in the belief that it generally affords deeper understanding than model exploration. However, the key is reducing the time required for students to achieve enough model construction skill so that science lessons based on model construction become feasible. In the near future, we would like to run an experiment similar to the Hashem and Midouser one but using Dragoon instead of NetLogo. We hypothesize that constructing Dragoon models produces more learning gains than model exploration and it costs only 1 or 2 h more instructional time instead

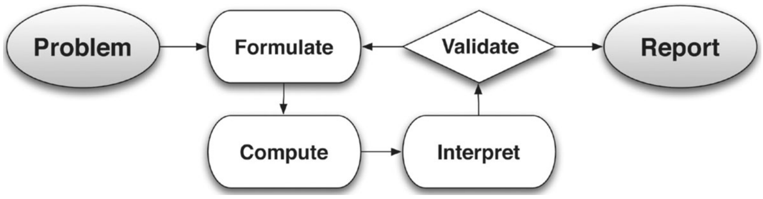


Fig. 1 Modeling is the process of formulating and debugging a model. From (CCSSO 2011)

of 48 h more. However, in our current studies, we compare Dragoon-based model construction only to baseline instruction that does not involve computational model construction. If Dragoon cannot beat baseline instruction, then there is no sense comparing it to model exploration.

However, "constructing a model of a given system" can mean many things depending on how the system is "given" (VanLehn 2013). For easy reference, let us define three points along a continuum of model construction tasks.

The most difficult model construction tasks merely identify the system but do not provide any information about it. Such tasks require the students to seek information empirically or in the literature, so they are a kind of inquiry activity. For example, such a task is: "Create a model of the elephant population of the Serengeti ecosystem covering the next 20 years." This brief text is all that the students are given, so they must search for the current size of the elephant population, its historical sizes, factors that influence the population, etc. and eventually formulate all this information as a model.

At an intermediate level of difficulty are model construction tasks where the system is described completely and concisely. For instance, an example of such a description is:

> As of August, 2014, there were 7535 elephants in the Serengeti ecosystem and an annual growth rate of 5 %. Construct a model of the population covering the next 20 years.

Although such problems often provide relatively complete information about the system and students are not expected to go to the literature to find out more about the system, even a complete system description may require making key assumptions, such as knowing that "growth rate" includes all sources of change in the elephant population, including births, deaths, emigration and immigration. A concise system description may also include extra, distractor information. Such model construction problems are like the word problems that are widely used in arithmetic and algebra classes.

Perhaps the simplest model construction tasks occur when students are first learning how to use the tools; these tasks provide a complete and concise description of the *model* as well as the system to be modeled. For instance, if the modeling language is algebraic equations, then such a description would be:

> Construct a model of the Serengeti elephant population covering 2014 to 2034. Let E(t) be the herd population in year t. Assume $E(2014) = 7535$ and $E(t+1) = E(t) + 0.05*E(t)$.

If the students' task is to construct a model using a graphing calculator or a spreadsheet, then they still have some work to do despite the concise description of the model. Similarly, a graphical system dynamics model can be described in terms of the nodes that need to be defined, and an agent-based model can be described in terms of the procedures each agent should follow.

In summary, three types of model construction tasks have been defined: One identifies the system but does not describe it; the second describes the system completely and concisely; the third describes both the model and the system completely and concisely. This research focuses on the latter two types of model construction tasks, so let's adopt some terminology:

- An *analytic* model construction task provides a complete and concise description of the system to be modeled. Solving such problems requires *analytic* skill, and a student who can solve any such problem has acquired *analytic* mastery.
- A *notational* model construction task provides a complete and concise description of both the model and the system. Solving such problems requires *notational* skill, because students must translate the model description into the notation of the model. A student who can solve any such problems has acquired *notational* mastery.

Although analytic model construction tasks are often used in instruction (e.g., Mandinach and Cline 1994), notational model construction tasks appear to be less common, and the distinction between them has not been noticed or defined before.

Prior work on Educational Use of Graphical System Dynamics Modeling

There has already been considerable work on using graphical system dynamics modeling languages in education. It began in 1983 when Stella was first released. Educators were strongly impressed by the potential pedagogical benefits of Stella, and several large projects were conducted wherein teachers worked with researchers to co-design Stella-based instruction in a variety of disciplines (Doerr 1996). However, after a decade of studies, participants reached a discouraging conclusion (Alessi 2000a; Mandinach and Cline 1994, 1996; Zaraza and Fisher 1999). They first noted that there were three major ways to use Stella:

1. Model exploration
2. Model construction “word problems,” which are called analytic tasks in the terminology introduced above.
3. Model construction of a non-trivial natural or engineered system. These are the first kind of model construction task mentioned early, which involve some type of inquiry.

Activity 1 (model exploration) was by far the most common activity, even among participants who had tried hard for many years to get their classes to do activity 3 (model-based inquiry). Activity 2 (analytic model construction tasks) was mostly done in math classes, whereas our focus is on science instruction. After several years, researchers and reviewers concluded that using model construction to learn science was rare because it required analytic mastery, which was not taught in math classes so it had to be taught in science classes. Although a few students attained sufficient mastery quickly, many students struggled to construct models even after many hours of instruction.

This suggested that the modeling language should be simplified in the hope that analytic mastery would be more quickly attained. This approach was pursued by two successful projects, Model-It (Crawford and Cullin 2004; Lee et al. 2011; Metcalf et al. 2000) and Betty’s Brain (Biswas et al. 2010; Chin et al. 2013; Leelawong and Biswas 2008; Schwartz et al. 2008, 2009; Segedy et al. 2012a, b). Both used a graphical modeling language that was similar to a concept map, in that nodes represented quantities and links represented how one quantity influenced another. Links were labelled with icons (in Model-It) or symbols (in Betty's Brain) that indicated whether the influence was positive, negative, strongly positive or strongly negative. A key distinction in system dynamics modeling (between regular functions and integrals, i.e., stocks) was either absent (Betty's Brain) or turned off for most students (Model-It). Thus, students primarily worked with qualitative descriptions of direct relationships between quantities. Lee et al. (2011) showed that inquiry instruction based on Model-It was more effective than inquiry instruction based on answering questions. Chin et al. (2010) showed that Betty's Brain was more effective than instruction based on either concepts maps or science kits.

While these studies of Model-It and Betty's Brain showed benefits for model construction compared to instruction without models, they did require many hours of training from the students. When shorter studies compared model construction to baseline instruction, they usually did not show uniformly reliable benefits for model construction (VanLehn 2013). For instance, a particularly well-controlled study (van Borkulo et al. 2012) compared students working on Co-Lab to students reading texts and answering essay questions. There were differences in the expected direction in only 2 of 8 measures. However, the Co-Lab instruction was short: 150 min.

Although these simplified modeling languages make it possible to use model construction to teach science, their representational powers are quite limited compared to Stella and other graphical system dynamics languages. Whereas Stella and similar languages are roughly equivalent to differential equations, the augmented concept maps of Model-It and Betty's Brain are roughly equivalent to algebraic equations. They lack integrals (stocks), so they can only model *static* systems and not *dynamic* ones. Perhaps most importantly, they do not allow feedback loops, so they cannot model many important system behaviors such as homeostasis, oscillation or exponential decay. The positive empirical results with the simplified languages are encouraging in that they showed model construction was more effective than instruction based on baseline instruction. However, the problem remains: How can students use models to learn about *dynamic* systems?

Meanwhile, early efforts to make object-oriented programming easy enough for children to do (i.e., Logo (Papert 1980), Smalltalk (Goldberg and Tenenbaum 1975)) eventuated in a new computational modeling paradigm, now called *agent-based modeling*. After several early developments (Boohan 1995; Neumann et al. 1999; Repenning et al. 2000), Uri Wilensky's NetLogo (ccl.northwestern.edu/netlogo) emerged as the most popular agent-based modeling tool, although competitors are beginning to appear such as Molecular Workbench (mw.concord.org) and SimSketch (Bollen and Van Joolingen 2013).

Case studies with NetLogo showed that there exist K12 students who can construct models of non-trivial systems (Centrola et al. 2000; Levy and Wilensky 2005; Wilensky 2003; Wilensky and Reisman 2006; Wilensky and Resnick 1999). Unfortunately, it proved to be difficult to replicate the success of case study participants in classrooms of students with variable background. For instance, Hashem and Midiouser (2010, 2011) found that it took 48 h of instruction in NetLogo programming to prepare students for a science lesson.

Because achieving analytic mastery takes too long for some students for both graphical system dynamics models and agent-base models, model exploration has become the dominant method for incorporating computational modeling into science instruction. Model exploration is the centerpiece of many interventions such as:

- *Biology*: (Buckley et al. 2004; Hickey et al. 2003; Horwitz et al. 2010; Wilensky and Novak 2010)
- *Chemistry* (Chiu and Linn 2014; Levy and Wilensky 2009a, b)
- *Earth science*: (Gobert and Pallant 2004; Svihla and Linn 2012)
- *Physics*: (Sengupta and Wilensky 2009; Shen and Linn 2011)

In summary, the current state of art in educational modeling uses model exploration almost exclusively. Although the standards are clearly calling for students to practice model construction (see Fig. 1), it is absent from the curriculum because analytic mastery takes too long to acquire for some students. Nonetheless, there are always a few students who master model construction quickly and deftly apply it to construct a deep understanding of systems, which keeps the hope alive that all students can somehow be taught model construction quickly enough that it can be a useful tool for understanding natural and engineered systems. This is the problem addressed by the project reported here. For the reasons discussed above, we are focusing on graphical system dynamics models.

Basic Technical Approach

One approach to making model construction feasible in schools is to reduce the time required to achieve analytic mastery enough that science classes can afford to teach analytic mastery instead of relying on math classes to do so. This project started by pursuing that approach and achieved some success. By simplifying the notation and having students practice solving modeling problems on a step-based tutoring system, it appears that most students who have had high school algebra can achieve analytic mastery with Dragoon in a maximum of 7 h of instruction, with a mean of about 5 h (VanLehn et al. 2016).

However, for most science instruction applications, spending 5 to 7 h on a prerequisite skill isn't feasible. If teachers can only spend three or four class periods on teaching a particular dynamic system, then they can't afford to spend most of that time getting students up to speed on analytic mastery. During the course of the studies described here, we discovered a more efficient instructional method: After a brief introduction to the modeling notation and the computer user interface, students are given *notational* model construction tasks for the target dynamics systems. Thus, they learn the target science concepts while they attain notational mastery. They do not attain analytic mastery, nor do they need it in order to learn the science.

The rest of this document presents first the model construction system, Dragoon, and then the studies where it was used for science instruction.

Dragoon

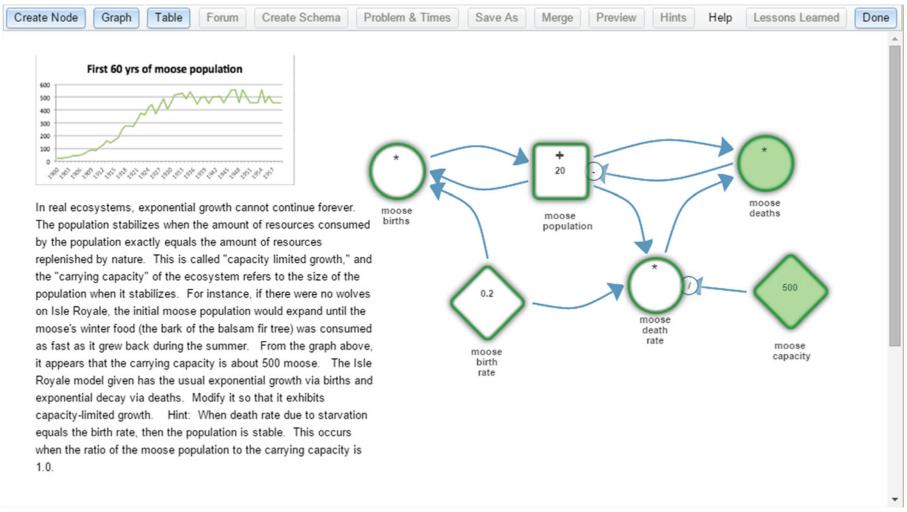
First, some terminology needs to be underscored. A *system* is just a part of the real world, and a *dynamic system* is a part of the real world that changes over time. A record of the system's changes is called its *behavior*. A *model* is an expression in a modeling language that can be interpreted by computer. Executing a model generates its *predictions*. An accurate model will generate predictions that match the system's behavior.

Dragoon uses a graphical notation similar to the one pioneered by Stella (Doerr 1996). Figure 2 shows a simple model in Dragoon. Each node in the directed graph represents both a quantity and how its value is calculated. For every input to a node's calculation, there is a link coming into the node from a node representing that input quantity.

- A *rectangle* represents a quantity whose value is the integral of its inputs over time.
- A *circle* represents a quantity whose value is an ordinary function (i.e., no integrals or differentials) of its inputs.
- A *diamond* represents a parameter, which is a quantity with a constant value. Parameters have sliders so that the user can easily modify their value and observe the effect on the model's predictions.

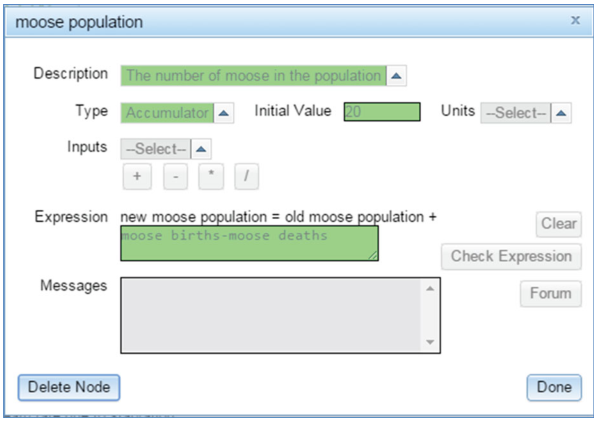
It is important to understand the difference between function nodes and accumulator nodes. Suppose a quantity X is determined by Y + Z; then X is represented by either a function node or an accumulator node with both Y and Z as input links. However, it is often unclear to students which type of node to use. They should use an accumulator if Y + Z represents how much X *changes* with each unit of time. In mathematical language, $dX/dt = Y + Z$ or $X = \int (Y + Z)\, dt$. On the other hand, if X's value at any time tick is equal to the value of Y+ Z at that same time tick, then X should be represented by a function node. That is, $X = Y + Z$ at all times. Using the model of Fig. 2 as an example, "moose population" is an accumulator because it is the integral over time of its inputs, "moose births" and "moose deaths." On the other hand, "moose births" is a function because it is not an integral over time of its inputs, "moose population" and "moose birth rate", but instead is a function (i.e., the product) of their current values.

Authors construct a model by clicking on the "Create Node" button, and then filling in a form (Fig. 3) with a name for the node, its type (parameter, accumulator or function), its inputs and how its value is calculated. When the student clicks on the Graph or Table button, Dragoon pops up plots or tables of the quantities as a function of



In real ecosystems, exponential growth cannot continue forever. The population stabilizes when the amount of resources consumed by the population exactly equals the amount of resources replenished by nature. This is called "capacity limited growth," and the "carrying capacity" of the ecosystem refers to the size of the population when it stabilizes. For instance, if there were no wolves on Isle Royale, the initial moose population would expand until the moose's winter food (the bark of the balsam fir tree) was consumed as fast as it grew back during the summer. From the graph above, it appears that the carrying capacity is about 500 moose. The Isle Royale model given has the usual exponential growth via births and exponential decay via deaths. Modify it so that it exhibits capacity-limited growth. Hint: When death rate due to starvation equals the birth rate, then the population is stable. This occurs when the ratio of the moose population to the carrying capacity is 1.0.

Fig. 2 The Dragoon model construction system, showing a simple problem and a correct solution

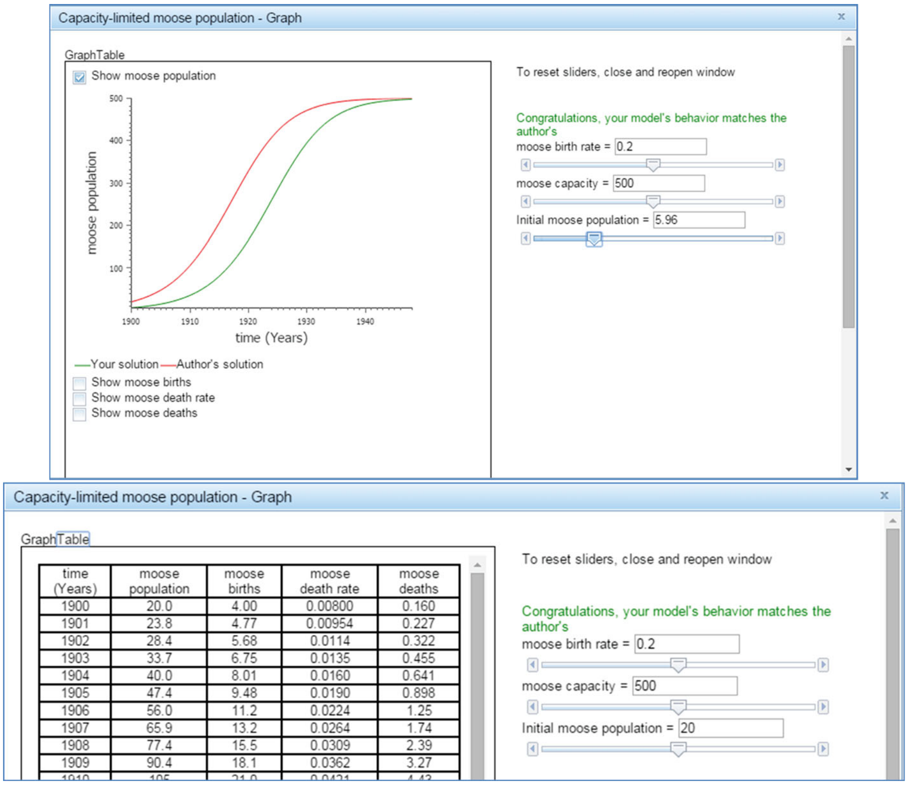


Description	The number of moose in the population
Type	Accumulator
Initial Value	20
Units	--Select--
Inputs	--Select--
Expression	new moose population = old moose population + moose births-moose deaths
Messages	

Fig. 3 The Dragoon node editor is a form. Filling in a field correctly causes it to turn green

time (see Fig. 4). The sliders allow the user to temporarily change the value of the parameters and observe the resulting change in the plots and tables.

What has been described so far is just the typical model construction system: an editor for constructing a model and displaying its predictions. When Dragoon is in author mode, this is all that the user has available. (Actually, there are a few more



To reset sliders, close and reopen window

Congratulations, your model's behavior matches the author's

moose birth rate = 0.2

moose capacity = 500

Initial moose population = 5.96

To reset sliders, close and reopen window

Congratulations, your model's behavior matches the author's

moose birth rate = 0.2

moose capacity = 500

Initial moose population = 20

time (Years)	moose population	moose births	moose death rate	moose deaths
1900	20.0	4.00	0.00800	0.160
1901	23.8	4.77	0.00954	0.227
1902	28.4	5.68	0.0114	0.322
1903	33.7	6.75	0.0135	0.455
1904	40.0	8.01	0.0160	0.641
1905	47.4	9.48	0.0190	0.898
1906	56.0	11.2	0.0224	1.25
1907	65.9	13.2	0.0264	1.74
1908	77.4	15.5	0.0309	2.39
1909	90.4	18.1	0.0362	3.27
1910	105	21.0	0.0421	4.43

Fig. 4 Graphs and table of the model's predictions

features in author mode. See (Wetzel et al. 2016) for details.) On the other hand, when Dragoon is in one of its student modes, it can give helpful feedback. This study used only *immediate feedback mode*, wherein Dragoon provides feedback on each step in the model-construction process. It colors an entry in the node editor green if its value matches the corresponding value in the author's model, and red otherwise (Fig. 3). When too many mismatches have been made on an entry, Dragoon fills in the entry with the author's value and colors it yellow. When the student closes the node editor, the colors of its entries are reflected in the color of its boundary (see Fig. 1).

Dragoon has other features than the ones discussed here (see (Wetzel et al. 2016) for descriptions). However, they were not used in the studies discussed below. In the studies below, students merely accessed Dragoon in order to solve an assigned modeling problem. They logged into Dragoon, creating an account if they had not done so already, found the assigned modeling problem in the menus, solved it, and clicked on Done.

Formative Evaluations

Our initial attempts to use Dragoon to teach science were conducted in late 2013. We first developed a short sequence of model construction problems that were intended to introduce students to Dragoon and modeling and give them sufficient skill in analytic model construction. The sequence involved only familiar systems, such as gaining and losing weight while dieting, so students did not need to learn any science in order to work through the introductory sequence. Pilot tests with 4 high school students indicated that everyone could work through the sequence in less than an hour.

In order to detect changes in students' skill and understanding of system dynamics modeling, a pre-test and post-test were developed. They consisted of a series of questions about a familiar system: the accumulation and removal of litter on the school grounds.

Next, four science teachers met with researchers for a 2 day workshop. The teachers were all from the same high school. The four teachers taught biology, chemistry, physics and earth science, respectively. They learned how to use Dragoon and then worked with researchers to develop modules for their classes. All the modules began by having students work through the introductory sequence. The modules then split, focusing on topics chosen by the teachers.

When the modules were enacted, the instruction lasted from 1 to 3 h, spread over a small number of days. The pre-test and post-test took about 35 to 45 min each. Teachers did all the teaching; researchers only observed and conducted a few interviews after selected classes.

The formative evaluations succeeded in uncovering a large number of flaws in the Dragoon software, the instruction, the professional development, and many other things. This led to a complete redesign of the notation in order to simplify it. The Dragoon software was completely rewritten. Dragoon was rewritten in JavaScript to run in a web browser and thus avoid Java installation problems. We learned to organize the instructional development differently. Using a single workshop to train teachers in using Dragoon and then expecting them to author problems and instruction was too challenging for most of the teachers. In subsequent work, we first taught teachers how to use Dragoon, and then had them work closely with researchers who did the actual authoring.

Most importantly, we found that students varied considerably in how fast they learned to use Dragoon. Some students required only a few minutes to understand the notation and learn how to use the interface. Others were still struggling after several hours of usage. For instance, students in the AP Physics class learned how to construct Dragoon models much faster than students in the biology class. This led us to develop different instructional approaches for different classes. The subsequent studies, presented next, used the rewritten Dragoon software and began implementing new practices.

Study 1: Physics

The purpose of study 1 was to evaluate Dragoon-based instruction in domain principles and concepts. The context was an AP high school physics course. Students who had already learned a little about the dynamics of falling bodies solved problems using either Dragoon or paper. This study occurred before we understood the distinction between notational mastery and analytic mastery. All the problems were presented using complete, concise descriptions of systems and not models. For instance, the descriptions did not use modeling terminology such as “accumulator” or “parameter.”

Design

The study had two instructional treatments which were different enough that they had to be run in different classes. Thus, all the students in one class used Dragoon-based instruction while all the students in a second class used paper-based instruction. The same instructor taught both classes. The students’ learning gains were measured using a pre-test and post-test with some shared items.

Subjects

Most of the students (75 %) were in 12th grade, most (67 %) were taking calculus, and most (76 %) had received an A grade on their most recent math class. In other words, these students had very strong mathematical backgrounds compared to most high school students.

There were 26 students in the Dragoon group and 29 students in the Control group. Although there were no statistically reliable differences between the groups in their math, computer science and physics background, the Control group had slightly more students (31 %) who programmed outside of class compared to the Dragoon group (19 %). The Control group also self-reported more familiarity with modeling than the Dragoon group.

Materials

The instructor developed the instructional materials, which consisted of five kinematics problems (see Fig. 5). The same kinematics principles and concepts were used in all the

1. An object starts at rest and falls from a building that is 56 meters tall. Graph its velocity and position from 0 seconds to 4 seconds. Use 9.81 m/s^2 for the acceleration due to gravity. Choose coordinates so that the ground is zero.
2. An object starts at rest and falls from a building that is 56 meters tall. Graph its speed and distance fallen between 0 seconds and 4 seconds. Use 9.81 m/s^2 for the acceleration due to gravity.
3. An object starts at rest and falls from the top of a silo that is 120 meters tall. For an object falling freely from rest, show that the distance traveled during each successive second increases in the ratio of successive odd integers (1, 3, 5, etc.). Assume that acceleration due to gravity is 9.81 m/s^2. Choose coordinates so that the top of the building is at zero and acceleration is positive.
4. A weight is thrown vertically upwards with a velocity of 8.5 m/s from an initial height of 1.2 m. Graph its velocity and height from 0 seconds to 3 seconds. How fast is it moving when it reaches a height of 4.0 m? How long is required to reach this height? Explain why there are two answers. Use -9.81 m/s^2 as the gravitational acceleration.
5. A stone is thrown vertically upward with a velocity of 12.0 m/s from the edge of a cliff 162 m high. Graph its velocity and height from 0 seconds to 8 seconds. How much later does it reach the bottom of the cliff? What is its velocity just before hitting? What total distance did it travel? Because the stone is moving against gravity we will set acceleration due to gravity at -9.81 m/s^2. Choose coordinates so that the bottom of the cliff is at zero.

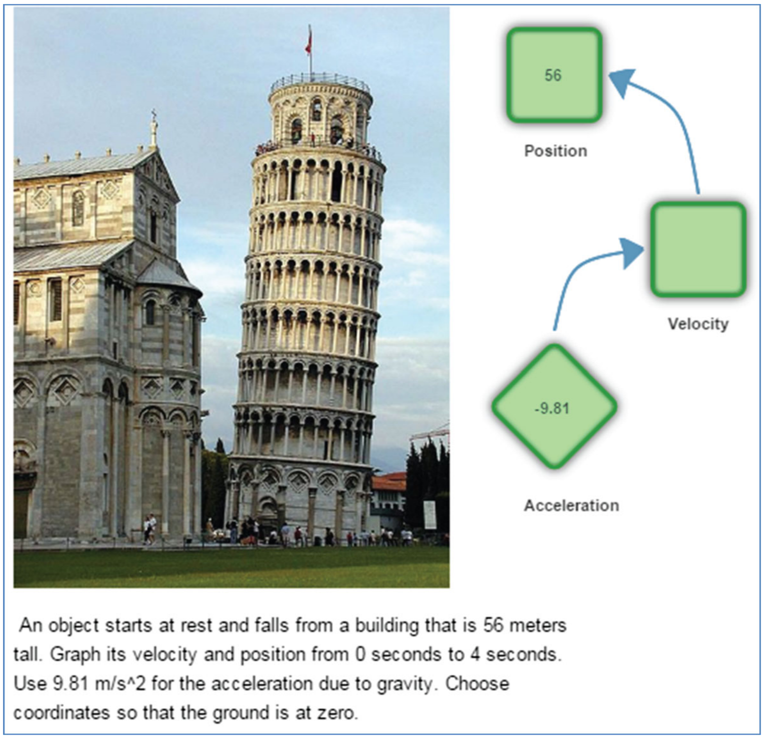
Fig. 5 The 5 problems used in Study 1

problems. The problems did not form a model progression. That is, each could be solved without having solved the earlier ones. The presentations were system descriptions rather than model descriptions. That is, the descriptions did not identify nodes nor indicate their types and inputs. Although the first two problems only asked the student to construct a model that would graph particular quantities, the remaining three problems required first constructing a model then answering questions about its behavior. Nonetheless, all the problems required the same model structure, which is shown in Fig. 6. Only the values of the parameters differed across problems.

The assessments were composed of four subtests. The first subtest assessed the student's skill at understanding dynamic systems and constructing tabular models of them. The second subtest consisted of solving traditional quantitative physics questions. The third subtest had students draw acceleration, velocity and displacement of falling objects over time. The fourth subtest asked students to draw a concept map that used a given set of node and link names. Although subtest 3 (drawing graphs) was given only on the post-test, the other three subtests were given during both the pre-test and post-test.

Procedure

The study spanned three consecutive days, with each class period lasting 50 min. Both the Treatment and Control classes were in the late morning. On the first day, the pre-test was given. On the second day, students solved the five instructional problems working as they normally did, wherein some students worked independently and some worked in pairs. The instructor gave the Dragoon class a brief demonstration of how to use Dragoon, and then told the students to solve the 5 problems. The Control students solved the same problems on paper while using their textbook as a resource. Although the instructor did not apply a uniform procedure for giving Control students' feedback



An object starts at rest and falls from a building that is 56 meters tall. Graph its velocity and position from 0 seconds to 4 seconds. Use 9.81 m/s^2 for the acceleration due to gravity. Choose coordinates so that the ground is at zero.

Fig. 6 A solution to the first problem used in Study 1

on their work, he did roam the classroom answering questions and inspecting students' progress. Moreover, these students tended to ask for feedback if they had any doubts about correctness. On the third day, the post-test was given.

Scoring of the Pre-Tests and Post-Tests

The content of the pre-test and post-test will be described in the results section. Here the methods for scoring are described for items that could not be scored by an exact match to an answer on a coding sheet.

For open response items, researchers constructed rubrics and trained a coder to apply them. Unfortunately, a second coder was not used so we do not have inter-coder reliability measures.

For the concept maps, two experts generated "correct" concept maps. Student concept maps were compared to the two expert maps, to produce four sets of scores per expert:

- *Exact* scoring counts propositions in the student map that exactly match propositions in that expert's map.
- *Ignore direction* scoring counts propositions in the student map that match propositions in the expert map when disregarding the arrow direction but taking into account the link label.

- *Ignore label* scoring counts propositions in the student map that match propositions in the expert map when disregarding the relationship label but taking into account the arrow direction.
- *Loose* scoring counts propositions in the student map that match propositions in the expert map if both the arrow direction and the relationship label are disregarded.

Thus, each student concept map received eight scores. The inter-rater reliability was assessed using generalizability theory. The amount of variability attributed to rater was high, and ranged from 26 % for the exact scoring to 49 % for the loose scoring. This means that scores from one expert map should only be compared to scores from the same expert map. Similarly, scores should not be compared across score types.

Results: Pre-Tests and Post-Tests

The first subtest assessed the students' skill at modeling system dynamics. On the pre-test, the two conditions hardly differed in their mean scores (Dragoon 17.42 (sd 2.98); Control 17.86 (sd 2.52); $p = 0.56$). On the post-test, the mean of the Dragoon students (18.24, sd 2.47) and the mean of the Control students (17.79, sd 2.74) were not significantly different ($p = 0.54$). Thus, it appears that on this measure, none of the students gained much, and the Dragoon students were no better at learning than the Control students.

The performance of the students on the second subtest, which measured their ability to answer traditional physics questions, turned out to be too difficult for these students. Out of a possible 3 points, one Dragoon student scored 1 point on the pre-test; all other scores, on both pre-test and post-test, were zero. The Control students were only slightly better, with a mean score of 0.21 (sd 0.56) out of 3 on the pre-test and a mean of 0.31 (sd 0.54) on the post-test. None of the differences between means were significant. Thus, on this measure, there was a floor effect.

The third subtest, which measured students' ability to draw the kinematic values of acceleration, velocity and displacement of a falling object over time, was given only during the post-test. Out of a maximum score of 6, the mean score of the Dragoon students (2.60, sd 1.29) was slightly lower than the mean score of the Control students (2.72, sd 1.16), but the means were not reliably different ($p = 0.71$).

The fourth subtest had students draw a concept map using a given list of physics terms. As mentioned earlier, there were eight scores per concept map (2 expert maps × 4 methods of matching). On all eight score types, the Control students scored higher than the Dragoon students on both the pre-test and the post-test. The gain scores (post – pre) averaged −0.66 for the Dragoon students and −2.46 for the Control students. This could be regression to the mean, perhaps caused by students putting less effort into the post-test concept maps.

Overall, the results suggest that neither group learned much physics nor much system dynamics modeling during the study. This may have been due to the short duration of the treatment. Moreover, these students had already been studying kinematics for 2 weeks. Clearly, they had much left to learn, as their scores were not high. Nonetheless, adding one more day of instruction onto the preceding 2 weeks appears not to have made much difference in their scores.

Results: Log Data

Because the Control students did not use a computer system for their problem solving, log data were available only for the Dragoon students. Thus, the analyses in this section comprise a brief description of the Dragoon students' work processes.

Although the observers reported that most student were on-task for most of the time, as one would expect from such academically successful students, the mean time spent solving problems on Dragoon was low. Of the 55 min period, students had a Dragoon problem open for only 21 min on average. Although the initial 15 min (approximately) of the period were spend on classroom logistics and a demonstration of Dragoon, the students were not given any readings or workbooks, so one would expect that they would spend almost all the remaining 40 min working on Dragoon. One possible interpretation is that although every student had a laptop running Dragoon, some worked in pairs. Thus, they may have collaborated on solving a problem using one student's computer, then opened Dragoon on the second computer only to copy the model from the first computer. Because the observer did not record the arrangement of students into pairs, this conjecture cannot be confirmed with the log data.

Even though students were given minimal instruction in how to use Dragoon, most figured it out. Figure 7 is a histogram of the number of problems solved. Only five students solved no problems at all. Although the mean number of problems solved was also low, 1.6 problems, many of the students (55 %) at least opened all the problems, even though they didn't always complete them. Overall, these findings suggest that their difficulty was in solving the problems rather than in using Dragoon.

Interpretation of Study 1

The bad news from this study is that our materials seemed to be somewhat advanced for these students, as neither group appear to have learned much new physics during the single day of instruction that they received. The good news is that some students seemed to have attained at least some proficiency in analytic model construction in that most (24 of 29) were able to solve at least one Dragoon problem without much instruction on how to use Dragoon. However, we do not know how much help students

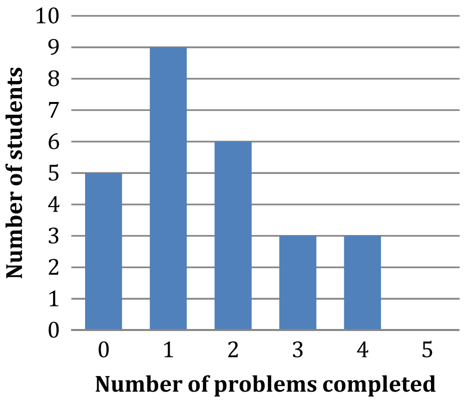


Fig. 7 Study 1, Dragoon students only, histogram of number of problems solved

were giving each other, so there is considerable uncertainty about this interpretation. On the other hand, observers of both Study 1 and the formative evaluation that occurred the preceding year in the same classrooms reported informally that the students seemed to have no trouble using Dragoon. This would be plausible given the strong backgrounds of these AP physics students. However, if the formative evaluation of the other classes is any guide, then rapid skill acquisition is not likely to occur in more typical high school science classes. Thus, the next study used a more typical class, notational tasks instead of analytic ones, and a longer, more structured instructional sequence than the one used in Study one.

Study 2: Physiology

The purpose of study 2 was to compare Dragoon to baseline instruction over a longer period of instruction using a highly structured workbook of notational tasks. The context was four high school physiology classes in a school in California. This study pioneered the use of concise model descriptions for the target systems.

Design

Two of the four classes used Dragoon instruction and the other two used baseline instruction. Two teachers were involved. One taught a baseline class; the other taught both Dragoon classes and a baseline class. Both classes used paper workbooks. The workbooks contained the same expository material (text and images) and they had the same multiple-choice and open response questions. The main difference was that the Dragoon classes constructed models in Dragoon whereas the baseline class constructed models by filling out tables and drawing graphs.

The classes proceeded in parallel. That is, all took the pre-test on the same day, had instruction on the same days, and took the post-test on the same day.

Participants

The two teacher participants were veteran physiology teachers, but had seldom used technology in their classrooms. They were interested in seeing if Dragoon could help them meet the Next Generation Science Standards for modeling, but were in general neutral about the benefits of technology for students.

There were 95 total participating students: 50 were in control classes and 45 were in Dragoon classes. Most of the students (72 %) were in 10th grade. The remainder were evenly split between 11th and 12th grade. The majority of the students (60 %) were currently taking geometry and had taken algebra I during the preceding year. Only one student was in Calculus. The average grade point for Dragoon students on their math class in the preceding year was 3.0 vs. 2.9 for the control students. Few of the students (5 %) had taken programming classes. In short, these students were more representative of the whole high school population than the AP Physics students of Study 1.

Materials

The materials consisted of two instructional workbooks, a pre-test and a post-test. The main workbook taught students about digestive energy balance, whereas the other workbook addressed blood glucose regulation. The second workbook was to be used only by students who had finished the first workbook. The post-tests did not address knowledge of blood glucose regulation.

The Dragoon workbook for digestive energy balance was authored by researchers with input from the teachers. Of its 27 pages, the first 14 pages were heavily illustrated, detailed, step-by-step instructions for logging into Dragoon and constructing a model. The remaining 13 pages were a progression of 8 levels. At each level, the students constructed a model in Dragoon or extended a model that they built in an earlier level, and then answered questions about its behavior. Models or model extensions were described in terms of nodes, because students were expected to have only notational mastery and not analytic mastery. The Appendix shows the Dragoon problems for the first 4 levels. Level 8 was a 30-node model that included the impact of exercise, height, gender, age, digestive activity expenditures and proportion of fat to lean body mass on energy expenditure and storage. Energy was expressed in calories. All parameter values and model relationships were accurate with respect to the physiology literature. The model exhibited and explained the surprising (to the students) fact that once you gained fatty tissue, reducing your caloric intake to its former values is not sufficient to burn off the excess fat.

The non-Dragoon instructional workbook for digestive energy balance was adapted from the Dragoon one. It had students fill in tables and draw graphs instead of constructing models.

The blood glucose workbooks were similar to the energy balance ones. It eventuated in the widely cited "minimal model" of glucose-insulin homeostasis (Bergman et al. 1979).

In short, although these workbooks were simple enough that high school sophomores could use them, they enabled students to construct models similar to those used by professionals. For instance, wrestlers and jockeys who monitor weight closely use versions of the energy balance model built by the students.

The pre-test consisted of five questions about energy balance and homeostasis (and a questionnaire about the students' background). The five questions required either short essays or mathematical derivations. Two questions involved interpretation of graphs.

The post-test consisted of 11 questions about digestive energy balance (and 7 questions seeking their opinions about Dragoon and their instructional experience). Of the 11 physiology questions, 6 asked the student about information that was explicitly presented in both worksheets:

- One question asked students to draw a concept map using a set of given terms.
- Another asked students to define or write a description of system; it listed a few concepts to include and asked students to be as specific as possible and to use examples of systems.
- Four multiple-choice questions were of the form "what factors directly affect…" and listed some factors.

These questions asked students to recall the model (i.e., energy balance and the variables and relationships affecting it) rather than draw inferences from it.

The remaining questions on the post-test had students apply the energy balance model in different ways:

- One multi-part question provided part of the model as a set of equations and asked for numerical answers given numerical values.
- Another multi-part question provided data for some model values and asked students to apply the model to predict weight gain/loss.
- Two questions asked students to draw qualitative graphs (i.e., the axes did not have numbers on them) showing the energy balance over time under two different conditions.
- Another question presented graphs of two individual's weights over time, and asked the student to compare and contrast them, using the model.

These questions, which asked students to apply the model, could be considered moderately deep in that the answers were not be in the instructional material, so students would need to make inferences in order to answer the questions. Moreover, some of the required inferences had been done routinely by the students in the baseline condition as they constructed tables, whereas making such inferences was probably infrequent in the Dragoon condition, because Dragoon did the arithmetic and drew the graphs. Although the post-test could be considered somewhat biased toward the baseline condition, even the questions that asked for application of the model did not strike us or the instructors as particularly deep, using the Chi et al. (1994) definition of "deep." They were typical of questions normally used for assessment in this class.

Procedure

The implementation window lasted 6 days with the first and last day devoted to pre-test and post-test respectively. The classes met for 54 min each. The Control classes met at roughly the same time of day as the Treatment classes.

For the Dragoon classes, the teacher introduced students to the idea of systems modeling and a researcher introduced students to Dragoon on Day 2. On Day 3, the researcher continued the introduction on Dragoon by working through an example Dragoon problem with students as a whole class. This took about 10 min. Students then started working through the workbooks. The first half of the Dragoon workbook was devoted to teaching students how to understand the notation and to use Dragoon. Although lengthy due to the many screenshots, students seem to proceed relatively quickly through this introductory material, and all finished it by the end of Day 3. On Days 4 and 5, students continued to work through the workbooks and solve energy balance problems with Dragoon.

For the Control groups, the teachers introduced students to the idea of systems modeling on Day 2. Students worked through the workbook problems in groups on Days 3 and 5. Because the Dragoon classes had most of a day devoted to learning Dragoon (Day 3), the Control classes had an open day (Day 4) devoted to topics that were irrelevant to energy balance. Thus, the Control and the Dragoon classes had approximately the same amount of class time, 2 days, for solving problems, and they took the pre-tests and post-tests on the same days.

In both the Dragoon and Control classes, students worked in small, self-selected groups at their own pace. Students were encouraged to collaborate and ask each other for help. The teacher circulated among the students, keeping them on task and providing help on the physiology when asked. Although workbooks were collected at the end of each day and handed back at the beginning of the next day, they were not scored or marked between days. The teachers told students that they would be examining the student's work and test results, but they were vague about when and whether the tests would contribute to the students' grades.

Although the Dragoon students received feedback immediately from Dragoon as they solved problems, feedback was delayed for the Control students. Teachers checked the Control students' work as the roamed the classroom, and insured that by the time a workbook was handed in, all the work on it was correct. After the experiment was concluded, researchers confirmed that all Control workbooks were filled out correctly with at most minor unintentional errors such as arithmetic mistakes.

In short, the procedure tried to control for some important variables, including time available for problem solving, amount of feedback, time of day, and amount of collaboration. However, this was a small scale, real-world study, so control was weak.

Results: Pre-Test and Post-Tests

Scoring of the pre-tests and post-test was done in the same way as scoring was done in study 1 (see section 4.5). Once again, we collected inter-coder reliability measures for the concept maps but not for the other test items.

On the pre-tests, the mean score of the Dragoon students was 1.69 (SD = 1.58) out of a maximum score of 6. The mean score of the Control students was 1.16 (SD = 1.04), which was reliably lower ($p < .01$; two-tailed).

On the post-tests, excluding the concept-mapping question, the mean score of the Dragoon students was 4.53 (SD = 1.71) out of a maximum of 10 points possible. The mean score of the Control students was 3.59 (SD = 1.52), which was reliably lower ($p = .006$) with a moderately large effect size (d = 0.62; where d = difference in post-test scores / standard deviation of control post-test score). However, because the pre-test scores between the two groups were reliably different, an ANCOVA was run with pre-test score as the covariate. This showed that the Dragoon group performed reliably better than the Control group ($p = .029$) with a medium effect size (d = 0.47, where d = difference in adjusted post-test scores / standard deviation of pooled adjusted post-test scores).

On the post-test question that asked students to draw a concept map using a set of given physiological terms, the results were marred by missing data. Concept maps were not drawn by 9 of the Dragoon students and 7 of the Control students. As in study 1, the maps were scored by comparing them to an expert's map using four different rubrics. Dragoon students scored higher than the Control students on all four rubrics, but none of the differences were statistically reliable.

One question appeared on both the pre-test and post-test. It displayed two graphs, one showing a linear increase in weight over time and the other showing a linear decrease in weight over time. The question asked, "Compare and contrast the two graphs and explain what you think might be going on with each person's weight based on your understanding of energy balance including energy storage, energy ingestion, energy expenditure, and the respective individuals' starting weights." This open-response question was intended to provide an assessment of students' overall understanding of the system. The mean scores on the post-test were higher than the mean scores on the pre-test for both groups (Dragoon: 49 → 80 %; Control 42 → 50 %), which suggests that both groups learned from their instruction. The Dragoon group's gain was reliably larger than the Control group's gain ($p < .01$ using an ANCOVA with pre-test score as the covariate).

There was a small positive relationship between student math grade/final math grade composite and both their pretest score (r = 0.12) and their posttest score (r = 0.20). This suggests that students who excelled previously in math may have performed slightly better on the pre- and posttest.

Results: Log Data Analyses

Log data were available only for students in the Dragoon group, as the Control students worked only on paper. Because there were 45 Dragoon students and problem solving occurred on days 3, 4 and 5, there would ideally be 3*45 = 135 log files, but 11 were missing. It can assumed that these represent students who were absent from class that day (8 %).

Although the class periods were nominally 54 min each, the Dragoon students had to walk to the computer lab, start the computers and then later shut down the computers and walk back to their classroom, which reduced the amount of time they had available for solving problems. On the other hand, part of day 3 was devoted to solving problems. Observers estimated that the total available problem solving time was about approximately 105 min for the Dragoon students. The log files showed that students spent an average of 91 min with a Dragoon problem open, which seems consistent with the estimated 105 min that they had available since they also had to read and write in the workbook between Dragoon problems. In contrast, the Control students solved problems on days 3 and 5 (and not day 4), and they spend almost the whole 54 min class period doing so on both days because they did not go to the computer lab. Observers estimated that they had about 96 min of available problem solving time. The observers reported that almost all the students in all classes were on-task almost all the time. These figures suggest, albeit approximately, that the time-on-task was about the same for the two groups. However, the Dragoon group had to spend about 45 min learning about Dragoon. This extra "cost" was incurred during Day 3.

There were correlations between the students' score on the post-test and the measures of the work on Dragoon. However, this might be partially due to the students' prior knowledge and diligence. In order at least partially factor out incoming competence, adjusted post-test scores were used. A students' adjusted post-test score is their post-test score minus the score they would get based on the pre-test score alone. Figure 8 shows a scatterplot of the Dragoon students' adjusted post-test score versus the amount of time they spent problem solving over the 3 days. Even with pre-test factored out, there is still a moderate correlation (R = 0.47; $p < .05$). Similarly, students who completed more problems also scored higher on the post-test. Figure 9 shows a moderate correlation (R = 0.39; $p < .05$). Although these are correlations and not a

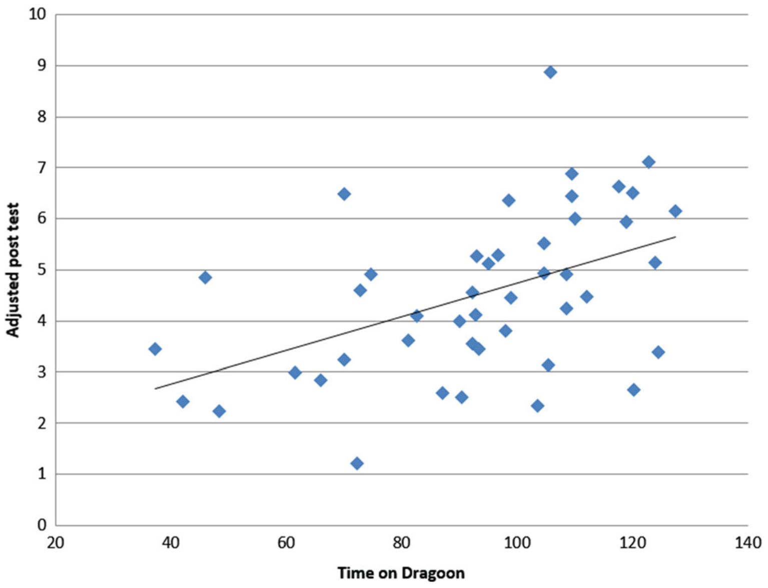


Fig. 8 Study 2, Dragoon students, adjusted post-test versus time

demonstration of a causal relationship, they suggests that the more work one does on Dragoon, the more one learns about physiology.

The energy balance instruction in both groups was comprised of 8 problems. The log files show that the Dragoon students completed a mean of 4.7 problems. Moreover, only 8 % of the Dragoon students completed all the problems. In contrast, 100 % of the Control students completed all their problems. Clearly, the Dragoon students were

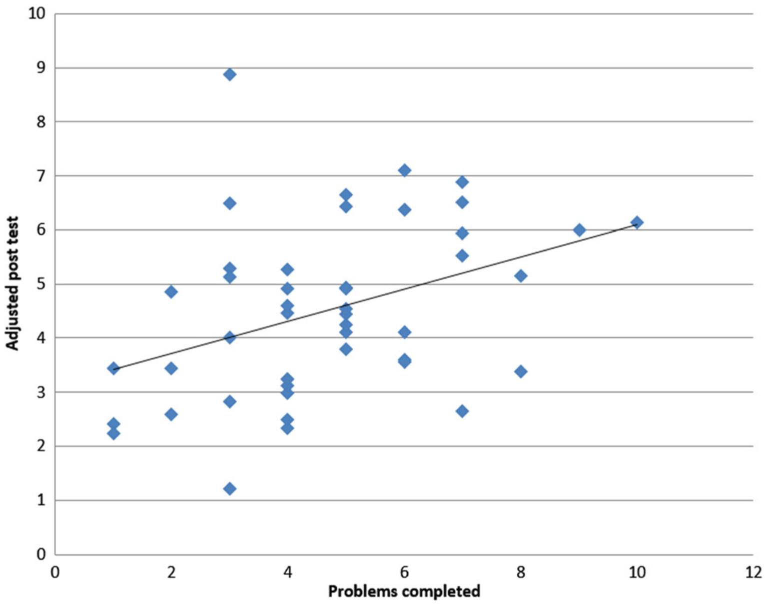


Fig. 9 Study 2, Dragoon students, adjusted post-test versus problems completed

moving more slowly through the curriculum. Without further study, we cannot determine why.

Interpretation of Study 2 Results

As usual when doing small-scale classroom studies, many important variables were not adequately controlled. In study 2, these included the initial competence of the two groups (measured with pre-test scores), the teacher, the amount of feedback given to students and the amount of time available for solving problems. Nonetheless, the experiment was strong enough to eliminate a few possibilities.

The experiment suggests that students do not need to have analytic mastery before they can learn science by constructing models. As our companion paper shows (VanLehn et al. 2016), even with the tutoring and scaffolding of Dragoon, it still takes students many hours to achieve analytic mastery. The Dragoon students in study 2 were given only notational problems (i.e., the text described both the system and the model). We do not know how many attained analytic mastery, but it would be a welcome surprise if any did. And yet the Dragoon students still managed to learn about the science of physiological energy balance.

The experiment also suggests that the extra cost of using model construction can be measured in minutes instead of hours. Unlike the Hashem and Mioduser (2010, 2011) study, where model construction students received an extra 48 h of instruction, the Dragoon students in this study received only about 45 min of extra instruction on how to construct Dragoon models. This was comprised of a demonstration by a researcher, and working through the first part of a workbook with lots of screen shots and low-level instruction on usage and notation.

The biggest mystery in the study 2 data is why did the Dragoon students complete so few problem and yet learn more physiology then the Control students? While the Control students completed all 8 problems, the Dragoon students completed on average only 4.7 problems. They attempted 6.8 problems. They may have read ahead in the workbook without doing the problems. Further study is needed in order to explain how they learned as much physiology as they did.

The main result of Study 2 is that instruction that includes solving Dragoon problems appears to be more effective for learning about a complicated physiological system than the same instruction with problems done on paper instead of Dragoon. However, given the lack of control over variables, this result must be considered encouraging but not at all definitive. Thus, we conducted another study that also compared Dragoon to baseline instruction.

Study 3: Population Dynamics

The purpose of study 3 was to see if the positive results of study 2 could be found in a new task domain, population ecology, with new teachers and students. The new instruction also featured a new instructional activity adapted from university studies of Dragoon (Iwaniec et al. 2014).

Design

Two classes used Dragoon instruction and one class used baseline instruction. Both treatments involved students working through paper workbooks in their normal classrooms. However, the Dragoon students also worked on laptops and constructed Dragoon models, while the baseline/control students filled out tables and drew graphs instead of constructing Dragoon models. Both groups took a pre-test, mid-test and post-test.

The study occupied three consecutive class meetings, and all classes started the study within 1 day of each other.

Participants

Two veteran biology teachers participated. They had not participated in Study 2, but enthusiastically volunteered when told about Study 2. Neither teacher routinely used technology in their classrooms.

The students were AP Biology students. A total of 59 students participated: 41 in the two Dragoon classes and 18 in the control class. Most of the students (58 %) were in 10th grade. The remainder were in 11th grade (35 %) and 12th grade (7 %). Some of the students (34 %) had taken programming classes.

The majority of the students (73 %) were currently taking trig/pre-calculus and 27 % were in Calculus. The grade point average for Dragoon students on their math class in the preceding year was 3.17 vs. 3.00 for the control students.

Materials

The materials consisted of two instructional workbooks, a pre-test, a mid-test, a post-test and a background questionnaire. The workbooks will be described first.

The first workbook explained the difference between linear, exponential and logistic growth of populations. It then had students do exercises, one set for each type of growth. Each set of exercises started by presenting information about the population and asking students fill-in-the-blank questions about it. Then students built the model either in Dragoon or by filling in tables and drawing graphs. The exercise set ended with more questions about the population.

The second workbook reviewed exponential growth, then covered predator–prey models. It started with the Lotka-Volterra model, which assumes both predators and prey would have exponential growth if they were not interacting. This model is unstable and always causes the predators to die off and sometimes the prey as well. The second model assumes that the prey would have logistic growth if it were not for the predators. This model always exhibits damped oscillation that converges on steady levels of both populations.

Besides the difference in the way models were constructed, the main difference between the Dragoon workbook and the control workbook was that the Dragoon workbook had 2 pages at the beginning of the first workbook devoted to explain the notation of Dragoon. Although the Dragoon workbook from the preceding study had a 14-page tutorial showing students how to manipulate the Dragoon interface, the tutorial material was reduced in this study because the observers felt that students learned the user interface better from the demonstration and from trial-and-error. The two-page explanation of Dragoon consisted of a 1-page explanation of the notation and two simple model-construction exercises done on paper that focused on discriminating between functions and accumulators.

All the model construction activities were presented exactly the same way to both groups. They described the system with one bullet per node, as in Fig. 10. Because the bullets did not explicitly tell the Dragoon students which type of node to define, this presentation could be considered slightly more difficult than the notational tasks used in Study 2. However, this bulleted presentation still relieves students of considerable amount of analytic work. It identifies the quantities and the formulas for calculating each quantity. Dragoon students must still supply a node type, but otherwise this bulleted presentation is close to the notational presentations of Study 2.

Because all three classes used a jigsaw activity (described next), there were two versions of the first workbook. One used lions as the population, and the other used zebras as the population. Otherwise, they were identical.

The pre-test and post-test were very similar to each other. They differed only in the "cover stories" of the problems (e.g., a horse population on the pretest was replaced by a fox population on the post-test). The test consisted of 5 open-response items, with a maximum score of 45 points. The first asked the students to provide a definition of a system, to invent and describe two population growth systems and to compare them. The second item asked students to sketch graphs for exponential growth and logistic growth, then compare their underlying populations. The third item asked students to complete the construction of a mathematical model of the logistic growth of horse population using a fill-in-the-blank format. The fourth item presented two graphs with labelled points, and asked students to explain what might be going on with each population, especially at the labelled points; both graphs were for the prey population of a predator–prey system but one was for the original Lotka-Volterra model and the other is for the modified Lotka-Volterra model. The fifth item presents two graphs with dampened oscillations; one labeled "rabbit population" and the other labelled "wolf populations". Students are asked to explain what's going on with this system.

Assumptions:
- The lion population is 4 in the first year (1950).
- Births are the only change in the population. Deaths, immigration, and emigration do not occur.
- The number of **lion births** each year is the **lion population** times **the probability of a lion giving birth (birth rate)** during the year.
- The probability of a lion giving birth each year is 50%.

Fig. 10 Presentation of a model construction problem in Study 3

The mid-test asked students to draw a concept map using a set of concepts (noun phrases such as "predator deaths" and "prey birth probability") and a set of links such as "affects" or "part of." The same concept mapping task was also included in the pre-test and post-test. The background questionnaire asked students about their grade level, programming classes, math classes, grades in the most recent math class, and whether they felt it was important to do well on the pre-test.

Procedure

The implementation window lasted 3 class sessions of 100 min each. Each class worked at its own pace through the instructional activities except that 40 min of the first and last day were devoted to pre-testing and post-testing respectively. The background questionnaire was given as part of the pre-test.

Three activities occurred between the pre-test and post-test. First, the students were first asked to work independently through the initial workbook, but before that, the experimenter demonstrated Dragoon to the Dragoon students. Second, the students completed the concept-mapping mid-test. Third, the teachers placed students in pairs and had them work on the second workbook.

All the classes used a simple jigsaw activity. When working individually, half of each class worked through the lion workbook, and the other half worked through the zebra workbook. When the students were placed in pairs, one student had studied lions and the other had studied zebras. When the pairs started the second workbook, the first task was to help each other construct a model of the species that they had not studied. The models were structurally isomorphic, but the values of the parameters were different. The second task in the workbook was to construct the Lotka-Volterra model together, collaboratively. The third task was to collaboratively construct the modified Lotka-volterra model.

As in Study 2, the teachers roamed both the control and Dragoon classrooms checking student's work, answering questions and giving feedback. The Dragoon students also got immediate feedback on their work from Dragoon itself. We did not attempt to equate the amount of feedback given to the two groups, because we view Dragoon's feedback as an integral part of the treatment and a possible source of its benefits.

Results

In the control condition, three pairs of students finished early. In the Dragoon condition, two pairs finished early. Unfortunately, the log data were lost, so we cannot provide a more detailed analysis of timing data than that.

Non-Concept Map Results

This subsection reports results of the pre-test and post-test, excluding the concept maps. All test items were scored using rubrics developed in consultation with the instructors. All items on all tests were scored by two raters. Inter-rater reliabilities were measured with Cronbach's alpha. The minimum reliability was 0.660 and the average was 0.822, where 0.7 is considered acceptable for low stakes testing.

Both the pre-test and post-test had 5 questions, but some students did not answer them all. Rather than estimate the missing data, we excluded tests that did not have all 5 questions answered. Thus, different tests have different N.

The pre-test scores of the two groups were not significantly different (two-tailed *t*-test, equal variances assumed, $p = 0.376$). The score for the Dragoon group (mean 26.6, standard deviation 5.8, $N = 39$) was slightly higher than the score for the control group (mean 24.9, standard deviation 7.7, $N = 18$).

Using an ANCOVA with pre-test score as the covariate, the adjusted post-test scores of the Dragoon group (mean 31.00, standard deviation 6.00, $N = 36$) were higher than those of the control group (mean 24.00, standard deviation 6.96, $N = 18$). The difference was reliable ($p < .000$) and large (d = 1.00).

Concept Map Results

This section reports results from the concept mapping tasks, which were given during the pre-, mid- and post-test. As in Study 2, concept maps were scored with 4 criteria. Since the scoring procedures were the same as Study 2, we assumed that the generalizability analysis would again advise against aggregating scores, so we used only one rater and compared only scores from the same criterion. We present here the scores from the Loose criterion. The pattern of results is the same with the other scores.

For concept maps drawn during the pre-test, the control group scored slightly higher (mean 26.3, standard deviation 12.4, $N = 21$) than the Dragoon group (mean 24.7, standard deviation 12.8, $N = 44$). The difference was not reliable ($p = 0.64$).

For concept maps drawn at the mid-test, the control group again scored slightly higher (mean 29.9, standard deviation 14.4, $N = 18$) than the Dragoon group (mean 27.5, standard deviation 13.1, $N = 41$). When the mid-test scores were compared in an ANCOVA with pre-test scores as a covariate (control adjusted mean 27.7, Dragoon adjusted mean 28.5), they were not reliably different ($p = 0.62$).

For concept maps drawn during the post-test, the Dragoon group now scored higher (mean 32.5, standard deviation 12.5, $N = 38$) than the control group (mean 28.9, standard deviation 16.0, $N = 18$). In an ANCOVA with pre-test as a covariate, the Dragoon adjusted score was higher (mean 33.0, standard deviation 10.7) than the control adjusted score (mean 27.3, standard deviation 11.7), and the difference was marginally reliable ($p = 0.08$) and moderately large (d = 0.49).

Aptitude-Treatment Interaction

Although the small number of control subjects makes it impossible to check whether there is an aptitude treatment interaction, Fig. 11 presents scatter plots that can give one a qualitative sense of which students benefited the most.

The scores on the non-concept map tests are shown in the scatterplot on the left. There might be a trend for students with higher pre-test scores to benefit more from Dragoon. However, in the concept map scores shown in the scatterplot on the right, it appears that the students with lower pre-test scores may have benefited more from Dragoon. Thus, it appears that future study will

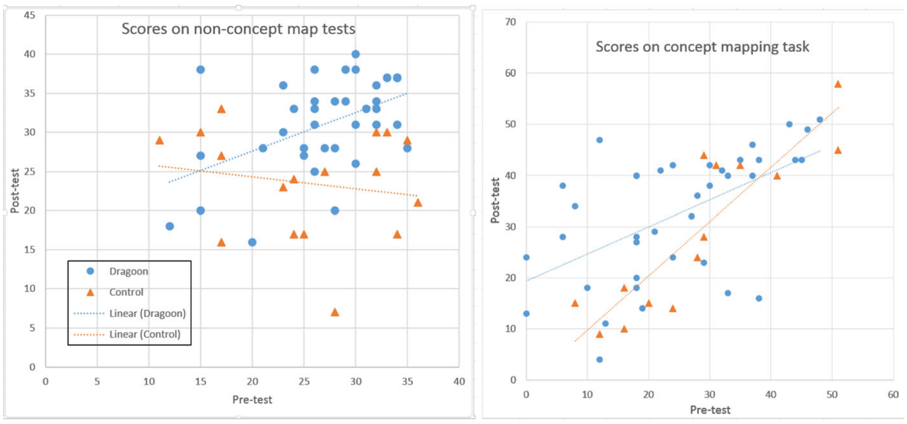


Fig. 11 Scatter plots of pre- vs. post-test scores; concept maps on right

be needed before we can tell which students benefit the most from model construction activities using Dragoon.

Interpretation of Study 3

Study 3's results suggest that the Dragoon students learned more than the control students. That is certainly good news, but what was the time cost? Although we only have rough observations of time-on-task, it appears that both groups of students spent about the same amount of time—they had the same number of class periods, and roughly the same number of students apparently finished all the materials. This is good news, because the Dragoon group in Study 2 needed more time (one extra class period) and solved fewer problems. The greater speed of the Dragoon group in study 3 may be due to the reduction of the tutorial content from 14 pages in Study 2 to 2 pages in Study 3.

Another interesting difference between study 2 and study 3 was the use of a jigsaw activity in study 3. That is, in both the control and Dragoon groups, students worked individually during the first half of the instruction on modeling the population dynamics of either zebras or lions. During the second half, a zebra student and a lion student worked together to model predator–prey interactions. The major benefit of Dragoon over the non-Dragoon materials may have occurred during the second half, because both the pre-test and mid-test scores were not significantly different.

Conclusions

Computational model construction activities can be used for many pedagogical purposes, such as acquiring skill in model construction, conducting an advanced type of inquiry, understanding the epistemology of science or developing ownership of knowledge (VanLehn 2013). The research presented here focuses exclusively on two instructional objectives: understanding and applying domain principles (Study 1) and understanding the function of a specific system (Studies 2 and 3). In general terms, the research asks: how can model construction be used to teach *science concepts*, *principles and facts*?

Although the goal is to teach science using computational model construction, the major impediment has always been this: it takes too long for students to acquire enough competence in computational model construction such that constructing models helps deepen their understanding of the systems and principles that they are modeling. So ironically, skill acquisition stands in the way of concept/principle acquisition.

This situation, which was reviewed earlier, suggests using step-based tutoring to teach the cognitive skill of model construction because such systems are often effective at teaching cognitive skills (Kulik and Fletcher 2015; Ma et al. 2014; Steenbergen-Hu and Cooper 2013, 2014; VanLehn 2006, 2011). Thus, we built Dragoon as a step-based tutoring system with the goal of reducing the time to master model construction to a few hours so that students could then use their skill to learn science. We also used a graphical language for expressing models that was simpler than the traditional stock-and-flow language and yet had more expressive power than earlier work on simplifying the language (Chin et al. 2010; Lee et al. 2011). We then used pilot studies with college students in lab settings to iteratively refine Dragoon and a sequence of training problems.

When the formative classroom evaluation reported here began, we believed the Dragoon training sequence would take high school students about an hour to complete and it would adequately prepare them to learn science via model construction. However, the formative evaluation showed that there was a huge variation in students ability to acquire skill in model construction and hence in their success at using model construction to learn science. Although students in the AP physics classes acquired model construction skill quickly, students in the other three classes did not.

This informal observation was confirmed in Study 1. Students in AP physics classes didn't even need the training sequence. Their teacher demonstrated how to use Dragoon, and then the student learned how to use it by themselves while the teacher circulated among them.

This huge variation in speed in acquiring model construction skill is consistent with the literature. Case studies indicate that selected students can quickly learn how to construct models and then use their skill to gain impressive scientific insights (Blikstein and Wilensky 2010; Centrola et al. 2000; Sengupta and Wilensky 2009; Wilensky 2003; Wilensky and Reisman 2006; Wilkerson-Jerde et al. 2014). On the other hand, there are studies with a representative sample of students that showed little benefit from model construction and/or little acquisition of model construction skill (Levy and Wilensky 2009b; van Borkulo et al. 2012). Finally, there are studies with longer training times that did show benefits for representative samples of students (Hashem and Mioduser 2011; Lee et al. 2011).

This led us to develop a third simplification. (The modeling language and the tutoring system count as the first two.) In earlier work, including Study 1, model construction problems were presented as typical word problems—a concise description of the *system* was given. Starting with Study 2, Dragoon students received a concise description of the *model* as well as a concise description of the system. This was done intentionally, so that the students could concentrate on understanding the system and not on the challenge of translating words into model relationships. They still had to understand the formal notation, so they presumably acquired notational mastery as they translated text descriptions of the model into the formal, graphical language of the model. It seems that notational mastery suffices for learning science via model construction, at least in Studies 2 and 3.

It seems that problems that provide students with a concise description of the model lie midway in difficulty between model exploration and traditional analytic model construction problems. Model exploration is still easier for instructors to use, because students do not need to understand the notation of the models nor the user interface for constructing them. On the other hand, students who must construct a model of the system are required to pay some attention to the quantities and relationships in the model. Some of these quantities and relationships may not be visible in a model exploration exercises. Thus, one would expect model construction students to learn more about the internal works of the model and hence the system.

There were hints in the data that Dragoon may be more effective when used by pairs of students than when used by individuals. In the formative evaluations, students worked mostly individually, whereas in Study 2, they worked in pairs. In Study 3, they worked initially alone and then in pairs, and may have learned more when working in pairs. This hypothesis would need to be tested; these observations are too informal to be trusted. However, if there is high variation in speed of acquisition of model acquisition, as argued earlier, then working in pairs would mean that at least one of the students may have above-average skill in model construction, and that may suffice for both students to learn the science content.

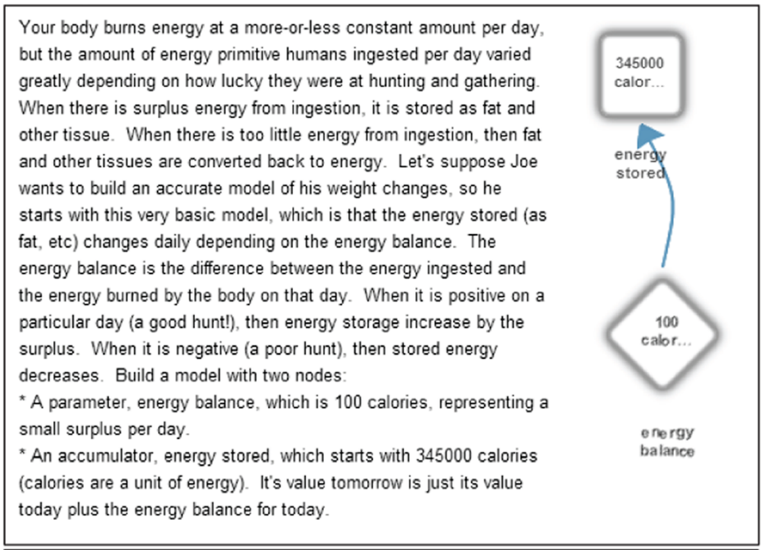
Our main goal was to demonstrate that model construction could become a feasible method for teaching science, and the experiments showed without a doubt that this goal was met. In all three studies, the amount of time required for Dragoon instruction was similar to the amount of time required for the control instruction. Moreover, Studies 2 and 3 suggest that Dragoon may have provided better knowledge of the target system, as one would expect of model construction when compared to instruction without model construction. Although Studies 2 and 3 were conducted without the controls that rigorous laboratory experiments have, and thus would need replication for the results on learning gains in order to be trusted, the prime goal of establishing feasibility requires that such studies be conducted in the messiness of real world classrooms. In that context, the combination of notational mastery, a step-based tutoring system, a simplified notation for graphical system dynamics models and perhaps working in pairs seems to have enabled teachers to feasibly use model construction, a method that had been abandoned as infeasible by many earlier advocates.

A good next step in this line of research would be to compare model construction to model exploration. In addition, it would be interesting and even necessary to try this form of instruction in more science classes in order to find out where it works best.

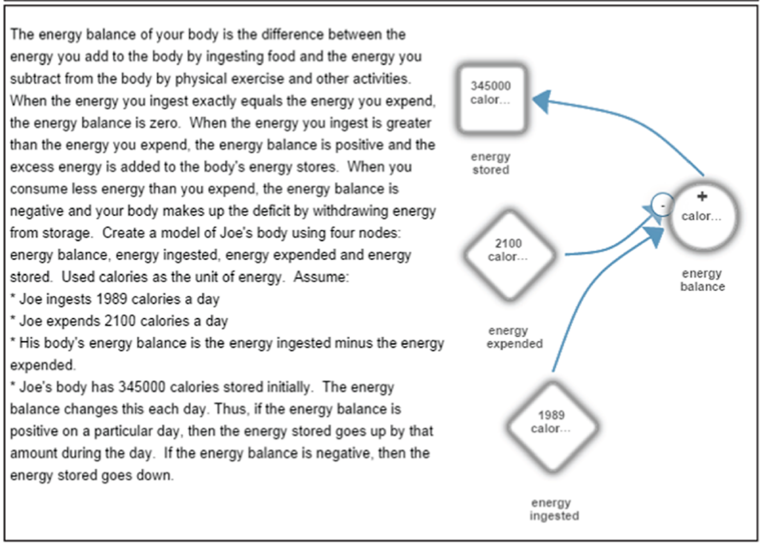
Acknowledgments This research is supported the Office of Naval Research contract N00014-13-C-0029 and by NSF award 1123823.

Appendix: The first four Dragoon problems of Study 2

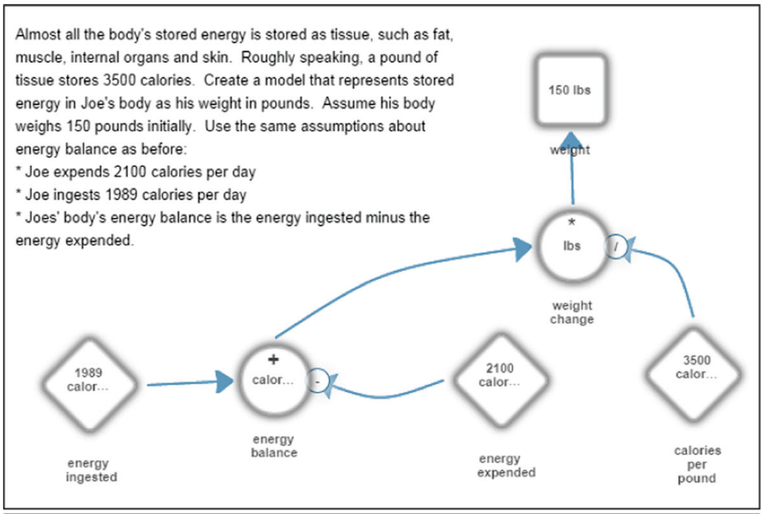
These figures show both the problem (text) and the correct solution (model). Of course students only saw the problem.



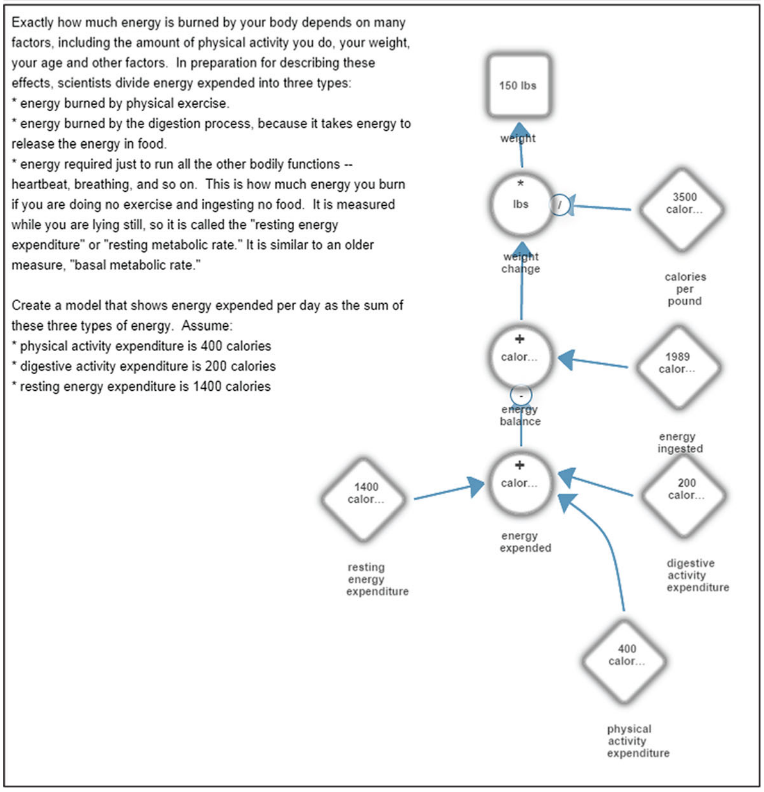
Your body burns energy at a more-or-less constant amount per day, but the amount of energy primitive humans ingested per day varied greatly depending on how lucky they were at hunting and gathering. When there is surplus energy from ingestion, it is stored as fat and other tissue. When there is too little energy from ingestion, then fat and other tissues are converted back to energy. Let's suppose Joe wants to build an accurate model of his weight changes, so he starts with this very basic model, which is that the energy stored (as fat, etc) changes daily depending on the energy balance. The energy balance is the difference between the energy ingested and the energy burned by the body on that day. When it is positive on a particular day (a good hunt!), then energy storage increase by the surplus. When it is negative (a poor hunt), then stored energy decreases. Build a model with two nodes:
* A parameter, energy balance, which is 100 calories, representing a small surplus per day.
* An accumulator, energy stored, which starts with 345000 calories (calories are a unit of energy). It's value tomorrow is just its value today plus the energy balance for today.



The energy balance of your body is the difference between the energy you add to the body by ingesting food and the energy you subtract from the body by physical exercise and other activities. When the energy you ingest exactly equals the energy you expend, the energy balance is zero. When the energy you ingest is greater than the energy you expend, the energy balance is positive and the excess energy is added to the body's energy stores. When you consume less energy than you expend, the energy balance is negative and your body makes up the deficit by withdrawing energy from storage. Create a model of Joe's body using four nodes: energy balance, energy ingested, energy expended and energy stored. Used calories as the unit of energy. Assume:
* Joe ingests 1989 calories a day
* Joe expends 2100 calories a day
* His body's energy balance is the energy ingested minus the energy expended.
* Joe's body has 345000 calories stored initially. The energy balance changes this each day. Thus, if the energy balance is positive on a particular day, then the energy stored goes up by that amount during the day. If the energy balance is negative, then the energy stored goes down.



Almost all the body's stored energy is stored as tissue, such as fat, muscle, internal organs and skin. Roughly speaking, a pound of tissue stores 3500 calories. Create a model that represents stored energy in Joe's body as his weight in pounds. Assume his body weighs 150 pounds initially. Use the same assumptions about energy balance as before:
* Joe expends 2100 calories per day
* Joe ingests 1989 calories per day
* Joes' body's energy balance is the energy ingested minus the energy expended.



Exactly how much energy is burned by your body depends on many factors, including the amount of physical activity you do, your weight, your age and other factors. In preparation for describing these effects, scientists divide energy expended into three types:
* energy burned by physical exercise.
* energy burned by the digestion process, because it takes energy to release the energy in food.
* energy required just to run all the other bodily functions -- heartbeat, breathing, and so on. This is how much energy you burn if you are doing no exercise and ingesting no food. It is measured while you are lying still, so it is called the "resting energy expenditure" or "resting metabolic rate." It is similar to an older measure, "basal metabolic rate."

Create a model that shows energy expended per day as the sum of these three types of energy. Assume:
* physical activity expenditure is 400 calories
* digestive activity expenditure is 200 calories
* resting energy expenditure is 1400 calories

References

Alessi, S. M. (2000). *The application of system dynamics modeling in elementary and secondary school curricula.* Paper presented at the RIBIE 2000 - The Fifth Iberoamerican Conference on Informatics in Education, Viña del Mar, Chile.

Alessi, S. M. (2000b). Building versus using simulations. In J. M. Spector & T. M. Anderson (Eds.), *Integrated and holistic perspectives on learning, instruction and technology* (pp. 175–196). Dordrecht: Kluwer.

Bergman, R. N., Ider, Y. Z., Bowden, C. R., & Cobelli, C. (1979). Quantitative estimation of insulin sensitivity. *American Journal of Physiology-Endocrinology and Metabolism, 236*(6), E667–E677.

Biswas, G., Jeong, H., Kinnebrew, J. S., Sulcer, B., & Roscoe, R. D. (2010). Measuring self-regulated learning skills through social interactions in a teachable agent environment. *Research and Practice in Technology Enhanced Learning, 5*(2), 123–152.

Blikstein, P., & Wilensky, U. (2010). MaterialSim: A constructionist agent-based modeling approach to engineering education. In M. J. Jacobson & P. Reimann (Eds.), *Designs for Learning Environments of the Future: International Perspectives from the Learning Sciences* (pp. 17–60). Berlin: Springer.

Bollen, L., & Van Joolingen, W. R. (2013). SimSketch: Multiagent simulations based on learner-created sketches for early science education. *IEEE Transactions on Learning Technologies, 6*(3), 208–216.

Boohan, R. (1995). Children and computer modelling: Making worlds with WorldMaker. In J. D. Tinsley & T. J. van Weert (Eds.), *Proceedings of the Sixth World Conference on Computers in Education* (pp. 975–985). London: Chapman and hall.

Buckley, B. C., Gobert, J. D., Kindfield, A. C. H., Horwitz, P., Tinker, R. F., Gerlits, B., & Willett, J. (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.

CCSSO. (2011). The common core state standards for mathematics: downloaded from www.corestandards.org on October 31, 2011.

Centrola, D., Wilensky, U., & McKenzie, E. (2000). *A hands-on modeling approach to evolution: Learning about the evolution of cooperation and altruism through multi-agent modeling – The EACH project.* Paper presented at the International Conference of the Learning Sciences.

Chi, M. T. H. (2009). Active-Constructive-Interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science, 1*, 73–105.

Chi, M. T. H., & Wylie, R. (2014). ICAP: A hypothesis of differentiated learning effectiveness for four modes of engagement activities. *Educational Psychologist, 49*(4), 219–243.

Chi, M. T. H., de Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*(3), 439–477.

Chin, D., Dohmen, I. M., Cheng, B. H., Oppezzo, M., Chase, C. C., & Schwartz, D. L. (2010). Preparing students for future learning with Teachable Agents. *Educational Technology Research and Development, 58*, 649–669.

Chin, D. B., Dohmen, I. M., & Schwartz, D. L. (2013). Young children can learn scientific reasoning with teachable agents. *IEEE Transactions on Learning Technologies, 6*(3), 248–257.

Chiu, J. L., & Linn, M. C. (2014). Supporting knowledge integration in chemistry with a visualization-enhanced inquiry unit. *Journal of Science Education and Technology, 23*, 37–58.

Collins, A., & Ferguson, W. (1993). Epistemic forms and epistemic games: Structures and strategies to guide inquiry. *Educational Psychologist, 28*(1), 25–42.

Corbett, A., Wagner, A. Z., Lesgold, S., Ulrich, H., & Stevens, S. M. (2006). The impact of learning of generating vs. selecting descriptions in analyzing algebra example solutions. In S. A. Barab, K. E. Hay, & D. T. Hickey (Eds.), *The 7th International Conference of the Learning Sciences* (pp. 99–105). Mahwah, NJ: Erlbaum.

Crawford, B. A., & Cullin, M. (2004). Supporting prospective teachers' conceptions of modelling in science. *International Journal of Science Education, 26*(11), 1370–1401.

Doerr, H. M. (1996). Stella ten-years later: A review of the literature. *International Journal of Computers for Mathematical Learning, 1*, 201–224.

Gobert, J. D., & Pallant, A. (2004). Fostering students' epistemologies of models via authentic model-based tasks. *Journal of Science Education and Technology, 13*(1), 7–22.

Goldberg, A., & Tenenbaum, B. (1975). Classroom communication media. *ACM SIGCUE Bulletin, 9*, 61–68.

Hashem, K., & Mioduser, D. (2010). *Learning by modeling (LbM) – The contribution of computer modeling to students' evolving understanding of complexity.* Paper presented at the Second International Conference on Educational Technology and Computer (ICETC), Shanghai, China.

Hashem, K., & Mioduser, D. (2011). The contribution of learning by modeling (LbM) to students' understanding of complexity concepts. *International Journal of e-Education, e-Business, e-Management and e-Learning, 1*(2), 151–157.

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

Horwitz, P., Gobert, J. D., & Buckley, B. C. (2010). Learning genetics from dragons: From computer-based manipulatives to hypermodels. In M. J. Jacobson & P. Reimann (Eds.), *Designs for Learning Environments of the Future*. New York: Springer.

Iwaniec, D. M., Childers, D. L., VanLehn, K., & Wiek, A. (2014). Studying, teaching and applying sustainabilty visions using systems modeling. *Sustainability, 6*(7), 4452–4469. doi:10.3390/su6074452.

Kulik, J. A., & Fletcher, J. D. (2015). Effectiveness of intelligent tutoring systems: a meta-analytic review. *Review of Educational Research*.

Lee, C. B., Jonassen, D., & Teo, T. (2011). The role of model building in problem solving and conceptual change. *Interactive Learning Environments, 19*(3), 247–265.

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence and Education, 18*(3), 181–208.

Levy, S. T., & Wilensky, U. (2005). *An analysis of students' patterns of exploration with NetLogo models embedded in the Connected Chemistry environment*. Montreal: Paper presented at the American Educational Research Association.

Levy, S. T., & Wilensky, U. (2009a). Crossing levels and representations: The Connected Chemistry (CC1) curriculum. *Journal of Science Education and Technology, 18*(3), 224–242.

Levy, S. T., & Wilensky, U. (2009b). Students' learning with the Connected Chemistry (CC1) curriculum: Navigating the complexities of the particulate world. *Journal of Science Education and Technology, 18*(3), 243–254.

Löhner, S., Van Joolingen, W. R., & Savelsbergh, E. R. (2003). The effect of external representation on constructing computer models of complex phenomena. *Instructional Science, 31*, 395–418.

Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology, 106*(4), 901–918.

Mandinach, E. B., & Cline, H. F. (1994). *Classroom dynamics: Implementing a technology-based learning environment*. Mahwah: Erlbaum.

Mandinach, E. B., & Cline, H. F. (1996). Classroom dynamics: The impact of a technology-based curriculum innovation on teaching and learning. *Journal of Education Computing Research, 14*(1), 83–102.

Metcalf, S. J., Krajcik, J., & Soloway, E. (2000). Model-It: A design retrospective. In M. J. Jacobson & R. B. Kozma (Eds.), *Innovations in Science and Mathematics Education: Advanced Designs for Technologies of Learning* (pp. 77–115).

Neumann, E. K., Feurzeig, W., & Garik, P. (1999). An object-based modelling tool for science inquiry. In W. Feurzeig & N. Roberts (Eds.), *Modelling and Simulation in Science and Mathematics Education* (pp. 138–148). New York: Springer.

NGSS. (2013). *Next Generation Science Standards: For States, By States*: The National Academies.

NRC. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting concepts, and Core Ideas*. Washington: National Academies Press.

Papert, S. (1980). *Mindstorms*. New York: Basic Books.

Repenning, A., Ioannidou, A., & Zola, J. (2000). AgentSheets: End-user programmable simulations. *Journal of Artificial Societies and Social Simulations, 3*(3), no page numbers.

Schwartz, D. L., Blair, K. P., Biswas, G., Leelawong, K., & Davis, J. (2008). Animations of thought: Interactivity in the teachable agent paradigm. In R. Lowe & W. Schnotz (Eds.), *Learning with animations: Research and implications for design* (pp. 114–141). Cambridge: Cambridge University Press.

Schwartz, D. L., Chase, C., Chin, D. B., Oppezzo, M., Kwong, H., Okita, S. Y., & Wagster, J. (2009). Interactive metacognition: Monitoring and regulating a teachable agent. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Handbook of Metacognition in Education* (pp. 340–358). New York: Taylor & Francis.

Segedy, J. R., Kinnebrew, J. S., & Biswas, G. (2012a). Relating student performance to action outcomes and context in a choice-rich learning environment. In S. A. Cerri, W. J. Clancey, & G. Papadourakis (Eds.), *Intelligent Tutoring Systems: 11th International Conference ITS 2012* (pp. 505–510). Berlin: Springer-Verlag.

Segedy, J. R., Kinnebrew, J. S., & Biswas, G. (2012b). *Supporting student learning using converstational agents in a teachable agent environment*. Paper presented at the Proceedings of the 10th International Conference of the Learning Sciences, Sydney, Australia.

Sengupta, P., & Wilensky, U. (2009). Learning electricity with NIELS: Thinking with electrons and thinking in levels. *International Journal of Computers for Mathematical Learning, 14*(1), 21–50.

Shen, J., & Linn, M. C. (2011). A technology-enhanced unit of modeling static electricity: Integrating scientific explanations and everyday observtiions. *International Journal of Science Education, 33*(12), 1597–1623.

Steenbergen-Hu, S., & Cooper, H. M. (2013). A meta-analysis of the effectiveness of Intelligent Tutoring Systems on college students' academic learning. *Journal of Educational Psychology.*

Steenbergen-Hu, S., & Cooper, H. M. (2014). A meta-analysis of the effectiveness of Intelligent Tutoring Systems on college students' academic learning. *Journal of Educational Psychology.* doi:10.1037/a0034752.

Stratford, S. J. (1997). A review of computer-based model research in precollege science classroom. *Journal of Computers in Mathematics and Science Teaching, 16*(1), 3–23.

Svihla, V., & Linn, M. C. (2012). A design-based approach to fostering understanding of global climate change. *International Journal of Science Education, 34*(5), 651–676.

van Borkulo, S. P., van Joolingen, W. R., Savelsbergh, E. R., & de Jong, T. (2012). What can be learned from computer modeling? Comparing expository and modeling approaches to teaching dynamic systems behavior. *Journal of Science Education and Technology, 21*, 267–275.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence and Education, 16*, 227–265.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist, 46*(4), 197–221.

VanLehn, K. (2013). Model construction as a learning activity: A design space and review. *Interactive Learning Environments, 21*(4), 371–413.

VanLehn, K., Wetzel, J., Grover, S., & van de Sande, B. (2016). Learning how to construct models of dynamic systems: The effectiveness of the Dragoon intelligent tutoring system. . *IEEE Transactions on Learning Technologies*, In press.

Wetzel, J., VanLehn, K., Chaudhari, P., Desai, A., Feng, J., Grover, S., . . . van de Sande, B. (2016). The design and development of the Dragoon intelligent tutoring system for model construction: Lessons learned. . *Interactive Learning Environments*, In press.

Wilensky, U. (2003). Statistical mechanics for secondary school: The GasLab multi-agent modeling toolkit. *International Journal of Computers for Mathematical Learning, 8*(1), 1–41.

Wilensky, U., & Novak, M. (2010). Teaching and learning evolution as an emergent process: The BEAGLE project. In R. Taylor & M. Ferrari (Eds.), *Epistemology and Science Education: Understanding the Evolution vs. Intelligent Design Controversy.* New York: Routledge.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories–an embodied modeling approach. *Cognition and Instruction, 24*(2), 171–209.

Wilensky, U., & Resnick, M. (1999). Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and Technology, 8*(1), 3–19.

Wilkerson-Jerde, M. H., Gravel, B. E., & Macrander, C. A. (2014). Exploring shifts in middle school learner's modeling activity while generating drawings, animations and computational simulations of molecular diffusion. *Journal of Science Education and Technology.*

Zaraza, R., & Fisher, D. (1999). Training system modelers: The NSF CC-STADUS and CC-SUSTAIN projects. In W. Feurzeig & N. Roberts (Eds.), *Modeling and simulation in science and mathematics education, Volume 1* (pp. 38–69). New York: Springer.