Leveraging Machine-Learned Detectors of Systematic Inquiry Behavior to Estimate and Predict Transfer of Inquiry Skill

Michael A. Sao Pedro, Ryan S.J.d. Baker, Janice D. Gobert,
Orlando Montalvo, and Adam Nakama
{mikesp, rsbaker, jgobert, amontalvo, nakama}@wpi.edu
Learning Sciences & Technologies Program
Worcester Polytechnic Institute
100 Institute Rd. Worcester, MA 01609

Abstract. We present work toward automatically assessing and estimating science inquiry skills as middle school students engage in inquiry within a physical science microworld. Towards accomplishing this goal, we generated machine-learned models that can detect when students test their articulated hypotheses, design controlled experiments, and engage in planning behaviors using two inquiry support tools. Models were trained using labels generated through a new method of manually hand-coding log files, "text replay tagging". This approach led to detectors that can automatically and accurately identify these inquiry skills under student-level cross-validation. The resulting detectors can be applied at run-time to drive scaffolding intervention. They can also be leveraged to automatically score all practice attempts, rather than hand-classifying them, and build models of latent skill proficiency. As part of this work, we also compared two approaches for doing so, Bayesian Knowledge-Tracing and an averaging approach that assumes static inquiry skill level. These approaches were compared on their efficacy at predicting skill before a student engages in an inquiry activity, predicting performance on a paper-style multiple choice test of inquiry, and predicting performance on a transfer task requiring data collection skills. Overall, we found that both approaches were effective at estimating student skills within the environment. Additionally, the models' skill estimates were significant predictors of the two types of inquiry transfer tests.

Keywords: scientific inquiry, exploratory learning environment assessment, skill prediction, machine-learned models, microworlds, behavior detection, designing and conducting experiments, Bayesian Knowledge-Tracing

This paper or a similar version is not currently under review by a journal or conference, nor will it be submitted to such within the next three months. This paper is void of plagiarism or self-plagiarism as defined in Section 1 of ACM's Policy and Procedures on Plagiarism. Portions of the work discussed in this paper were previously published in two conference papers at EDM2010 (Sao Pedro et al, 2010; Montalvo et al, 2010).

1 Introduction

Assessment and prediction of skill within Intelligent Tutoring Systems and Interactive Learning Environments has been successful for well-defined domains and problems such as problem solving in mathematics (e.g. Corbett & Anderson, 1995; Feng, Heffernan & Koedinger, 2009) and physics (e.g. Gertner & VanLehn, 2000). However, for more ill-defined domains, even though progress has been made (e.g. Mitrovic, 2003; Graesser, Chipman, Haynes & Olney, 2005; Dragon, Woolf, Marshall & Murray, 2006; Roll, Aleven & Koedinger, 2010), significant challenges still remain. Relevant to our work, assessment in the ill-defined domain of science inquiry, a domain embodying the skills at combining scientific processes and using reasoning to develop understanding in a science discipline (National Research Council, 1996, p.105), is difficult because scientific processes are both complex and multi-faceted. Though there are inquiry strategies known to be effective (cf. Chen & Klahr, 1999), there is no one right (or wrong) way to engage in inquiry (de Jong, 2006; Buckley, Gobert, Horwitz & O'Dwyer, 2010). Furthermore, without an adequate way of assessing skill, estimating overall proficiency becomes impossible. To combat these difficulties and promote learning, we are developing a web-based learning environment, Science Assistments, that lets students engage in inquiry within microworlds. This system aims to automatically assess and track authentic inquiry skills defined in the National Science Education Standards (National Research Council, 1996) over several science domains while adaptively scaffolding students' inquiry in real-time (Gobert, Heffernan, Ruiz & Ryung, 2007; Gobert, Heffernan, Koedinger & Beck, 2009).

We present here our work towards assessing and estimating proficiency on a subset of inquiry skills associated with designing and conducting experiments. These skills are demonstrated as students engage in inquiry using a middle school-level physical science microworld within Science Assistments. To support assessment, we developed detectors (models) of systematic data collection behavior based on

student log files using methods from the machine learning/educational data mining literature (cf. Baker & Yacef, 2009; Romero & Ventura, 2010). Training instances were generated by manually inspecting and coding a proportion of student activity sequences using "text replay tagging" of log files, an extension to the text replay approach developed in Baker, Corbett, and Wagner (2006). Similar to a video replay or screen replay, a text replay is a pre-specified chunk of student actions presented in text that includes information such as each student action's time, type, widget selection, and exact input.

In turn, the detector classifications, assessments of inquiry skill demonstrated in a practice attempt, can be aggregated into overall assessments of student proficiency. We compared two methods of estimating skill proficiency, an average-based approach which assumes no learning within the environment, and Bayesian Knowledge-Tracing (Corbett & Anderson, 1995), a more complex model which assumes learning between practice attempts. We compared the efficacy of these proficiency models in two ways. First, we compared them on predicting performance within the learning environment, providing a measure of the internal reliability of these estimates. Second, we compared these proficiency models on predicting performance on transfer tasks requiring inquiry skill. These tasks included a paper-style test of inquiry and a "hands-on" assessment in another domain. This enabled us to get a benchmark on the skill estimates' external validity. It also enabled us to study the relationship between standardized-test style questions and more hands-on inquiry. Though it has been argued that performance assessments are better suited than standardized-test style questions to assess inquiry skills (cf. Black, 1999; Pellegrino, 2001), rote paper tests are still typically used for assessing inquiry skills (cf. Alonzo & Aschbacher, 2004; Gotwals & Songer, 2006). Hence, the relationship between the two forms of assessment must be understood if our environment is to be used, in part, as an assessment tool.

The remainder of this paper is organized as follows. First, we present background work on the inquiry skills and systematic behaviors we are studying and previous research on assessing and

predicting inquiry skills, other skills, and behaviors using machine-learned models. Next, we present our methodology for gathering student data, including an in-depth description of the environment. Then, we describe the process by which we used low-level student data to build and validate machine-learned models of systematic inquiry behavior. Next, we present results on comparing our two approaches on predicting skill within the environment and predicting transfer of inquiry skill. Finally, we present a discussion and conclusions of the paper.

2 Background and Related Research

In our environment, students conducted inquiry using a phase change microworld and inquiry support tools. A microworld (Papert, 1980), as related to science learning, is a runnable, computerized model of real-world phenomena whose properties can be inspected and changed (Pea & Kurland, 1984; Resnick, 1997). Microworlds can support students' scientific inquiry because they share many features with real apparatuses and thus capitalize on perceptual affordances (Gobert , 2005). Using our phase change microworld, students explore how ice changes phases from solid to liquid to gas as it is heated. Its purpose is to foster understanding about the invariant properties of a substance's melting and boiling point through experimentation. Knowledge about melting and boiling points are content requirements outlined in the Massachusetts Education Standards for science (Massachusetts Department of Education, 2006). More details about the phase change environment are presented in Section 3.1.

We assess and estimate skills outlined under the "designing and conducting experiments" strand of the U.S. National Science Education Standards (National Research Council, 1996). Our efforts are focused here because learning to correctly plan and execute controlled experiments enables valid inference making based on data, an important component in generating knowledge within a domain through inquiry (Kuhn, 2005; de Jong et al., 2005). Moreover, we aim to develop models that support

prediction of whether students possess latent inquiry skills that can be tracked to performance outside of the phase change environment.

2.1 Data Collection Behaviors of Interest

In total, we assessed and estimated four systematic data collection behaviors representative of inquiry skills. Two reflected the conceptual and procedural knowledge necessary for conducting data collection. The first is designing controlled experiments, which typically associated with mastery of the Control of Variables Strategy (CVS) (cf. Chen & Klahr, 1999). This strategy states that one should change only a single variable to be tested, the target variable, while keeping all extraneous variables constant, to test the effects of that target variable on an outcome. Though several studies analyzed acquisition of CVS within environments meant to teach just that skill, in isolation from other inquiry skills (e.g. Sao Pedro, Gobert, Heffernan & Beck, 2009; Sao Pedro, Gobert & Raziuddin, 2010; Siler, Klahr, Magaro, Willows & Mowery, 2010), few have analyzed performance at designing controlled experiments as students engage in more open-ended inquiry tasks, as in our environment. A second, related behavior is testing stated hypotheses. This is demonstrated when students run experiments that could be used to support or refute any of their stated hypotheses. We separated this behavior from the designing controlled experiments since students may attempt to test their hypotheses with confounded designs, or may design controlled experiments for a hypothesis not explicitly stated.

The other two data collection behaviors of interest involve, at least in part, self-regulatory metacognitive processes (cf. Winne & Hadwin, 1998). These are using a data table tool and using a hypothesis viewer to plan which experiments to run next. Briefly, the data table tool is an inquiry support tool where students can see the results of all trials they ran during experimentation, and the hypothesis viewer enables students to keep track of all their stated hypotheses. More details about

these tools are given in Section 3.1. Planning is required when deciding how much data are necessary to support or refute a particular hypothesis, and what data are required to test all stated hypotheses.

We included these behaviors since planning is an important skill within scientific inquiry (de Jong, 2006) and metacognition is recognized as an important aspect of learning (Veenman, Van Hout-Worters & Afflerback, 2006; Dignath & Buttner, 2008; Azevedo, 2009). Furthermore, while several studies on self-regulation and metacognition within computer-based learning environments have been conducted (Aleven, McLaren, Roll, & Koedinger, 2006; Winne, et al., 2006; Manlove, Lazonder, & de Jong, 2007; Schraw, 2007, 2009), there is no consensus about how to automatically measure self-regulation (Hadwin, Nesbit, Jamieson-Noel, Code & Winne, 2007; Azevedo, 2009; Scraw, 2009) and few studies have addressed planning within the context of scientific inquiry (e.g. Manlove & Lazonder, 2004).

2.2 Assessment and Prediction of Inquiry Skills and Behaviors

In our approach, we aim to develop machine-learned assessment and proficiency estimate models of the four systematic data collection behaviors previously mentioned. Previous research has successfully used machine learning techniques to distinguish students' problem solving strategies within exploratory learning environments. For example, Bernardini and Conati (2010) used clustering and Class Association Rules to capture learner models of effective and ineffective learning strategies within an environment for learning about a constraint satisfaction algorithm. Ghazarian and Noorhosseini (2010) constructed task-dependent and task-independent machine-learned models to predict skill proficiency in computer desktop applications. Research has also been conducted on using machine learning techniques to model competency and knowledge within inquiry environments. Stevens, Soller, Cooper and Sprang (2004) used self-organizing artificial neural networks to build models of novice and expert performance using transition logs within the HAZMAT high school chemistry learning environment. They then leveraged

those models to construct a Hidden Markov Model for identifying learner trajectories through a series of activities. Rowe and Lester (2010) developed Dynamic Bayesian Network models of middle school students' narrative, strategic and curricular knowledge as students they explored within a 3D immersive environment on microbiology, Crystal Island. Finally, Shores, Rowe and Lester (2010) compared machine learning algorithms' efficacy at predicting whether students would utilize a particular inquiry support tool shown to improve learning within that same environment. The work presented here differs from this earlier work in one key fashion. Whereas previous work has looked for general indicators of problem solving skill in inquiry environments (Stevens, et al., 2004; Rowe & Lester, 2010), or predictors of whether students will use cognitive support tools (Shores, Rowe & Lester, 2010), the work in this paper develops models of specific inquiry subskills (cf. National Research Council, 1996) and tracks them over a series of activities.

In addition, our work differs from previous work that developed models of specific inquiry skills using a knowledge engineering approach, also known as cognitive task analysis. In these approaches, rules were defined that encapsulate specific behaviors (Koedinger, Suthers & Forbus, 1998; McElhaney & Linn, 2010) or differing levels of systematic experimentation skill (Buckley, Gobert & Horwitz, 2006, Buckley et al, 2010). Similarly, Schunn and Anderson (1998) engineered a rule-based ACT-R model of scientific inquiry based on an assessment of skill differences in formulating hypotheses, exploring, analyzing data, and generating conclusions between novices and experts.

Knowledge engineered models have also been used in several analyses of the relationship between scientific inquiry behavior and learning. For example, Buckley, Gobert and Horwitz (2006) and Buckley, et al. (2010) showed that systematic inquiry demonstrated within microworld-based activities positively affects students' acquisition of content knowledge, as measured by pre- and post-test gains. Specifically, they found that systematic performance on certain inquiry tasks within BioLogica, one of their microworld activities, predicted about 10% of the variance in students' post-test gain scores,

irrespective of whether they actually succeeded at the inquiry task during learning. In Dynamica, a software tool for Newtonian Mechanics, Gobert, Buckley, Levy and Wilensky (2007) also identified strategic approaches to inquiry tasks that had significant positive correlations with post-test conceptual gains. In Connected Chemistry, Levy and Wilensky (2006) found that model exploration during inquiry led to greater conceptual gains.

Like these past knowledge engineering approaches, we use student interactions with the learning software as a basis for creating our models. As with Schunn and Anderson (1998), we are interested in evaluating and quantifying students' skills as well as determining how well our detectors predict systematic behavior. Our approach, however, is different in that we do not prescribe rules for systematicity a-priori. Instead, given student data, human-classified labels, and a feature set derived from student data, we use machine learning techniques to build models of various inquiry behaviors. This technique has several advantages. First, the resulting models capture relationships that humans cannot easily codify rationally, while leveraging the human ability to recognize demonstration of skill. The models also represent boundary conditions – and the fuzziness at the edges of boundary conditions – more appropriately than knowledge engineering approaches. Finally, the accuracy and generalizability of machine learning approaches are easier to verify than for knowledge engineering, since machine learning is amenable to cross-validation, a standard method for predicting how well models will generalize to new data (cf. Efron & Gong, 1983).

2.3 Machine-Learned Models for Predicting Complex Learner Behaviors

It is worth noting that several others have successfully utilized machine learning techniques to model and detect other complex learner behaviors within learning environments. Beck (2005), for example, developed an IRT-based model incorporating response times and correctness to predict disengagement

in an approach called “engagement tracing”. Cocea and Weibelzahl (2009) labeled raw log files, distilled features, and then built and compared several models of disengagement yielded by different machine learning algorithms. Cetintas, Si, Xin and Hord (2009) used a combination of timing features, mouse movements unique to each student to build off-task behavior detectors. Walonoski and Heffernan (2006) and Baker, Corbett, Roll and Koedinger (2008) successfully built and validated gaming the system detectors by triangulating qualitative field observations with features gleaned from log files. And finally, Baker and de Carvalho (2008) and Baker, Mitrovic and Mathews (2010) labeled the gaming the system behavior using text replays, which also led to successful detectors under cross-validation. Our work is similar to these projects in that we follow a similar paradigm to construct our detectors of inquiry behaviors. Specifically, we leverage the success of Baker and de Carvalho’s (2008) and Baker, Mitrovic and Mathews’ (2010) use of text replays as our method of classifying training instances. Our research is the first to utilize this technique to generate models of specific systematic inquiry behaviors.

3 Methodology

3.1 Participants

Participants were 148 eighth grade students ranging in age from 12-14 years from a public middle school in suburban Central Massachusetts. Students belonged to one of six class sections and had one of two science teachers. They had no previous experience using microworlds within Science Assistments.

3.2 Materials

3.2.1 Phase Change Environment and Activities

The phase change environment (Figures 1 and 2), developed using OpenLaszlo (www.openlaszlo.org), had students engage in authentic inquiry using a microworld and inquiry support tools. A typical task provided students with an explicit goal to determine if a particular independent variable (container size, heat level, substance amount, and cover status) affected various outcomes (melting point, boiling point, time to melt, and time to boil). Thus, for a given independent variable, proficiency was demonstrated by hypothesizing, collecting data, reasoning with tables and graphs, analyzing data, and communicating findings about how that variable affected the outcomes.

These inquiry processes were supported by arranging them into different inquiry phases: "observe", "hypothesize", "experiment", and "analyze data". Students began in the "hypothesize" phase and were allowed some flexibility to navigate between phases as shown in Figure 1.

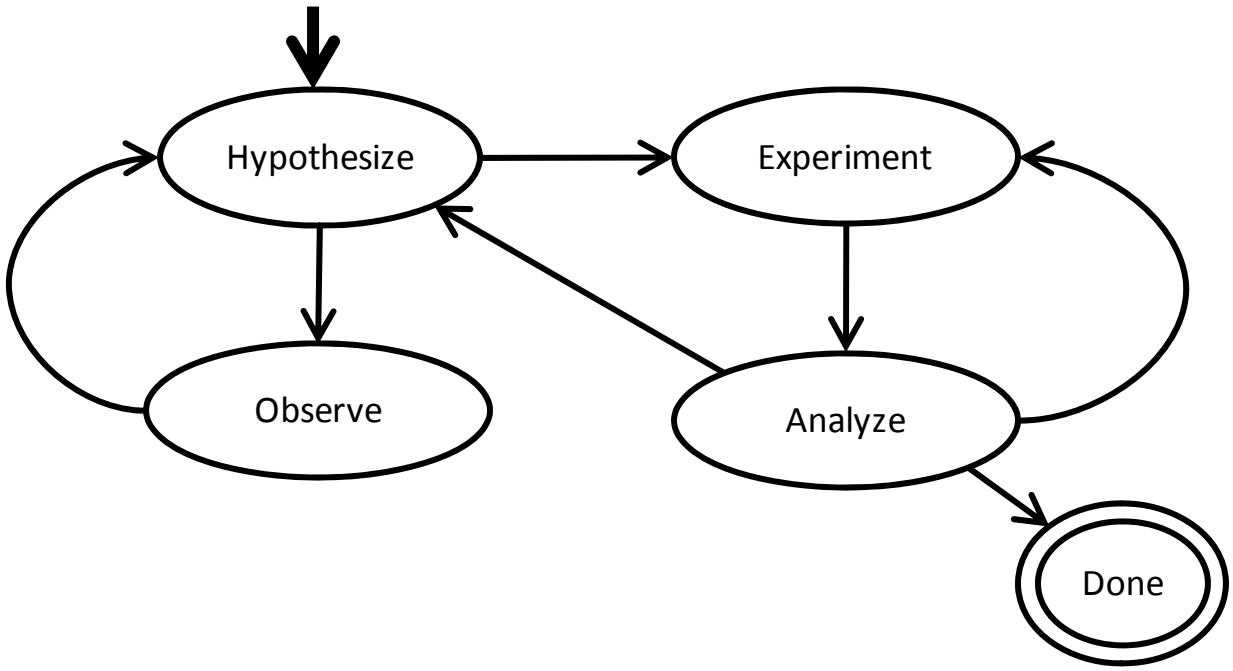

Figure 1. Paths through inquiry phases

In the “hypothesize” phase, students used the hypothesis constructing tool (Figure 2) to generate testable hypotheses. The “observe” phase and “experiment” phase (Figure 3) were similar. In the “experiment” phase, the student designed and ran experiments, and had access to two inquiry support tools, a data table summarizing previously run trials and a hypothesis list. These tools aimed to help students plan which experiments to run next. The “observe” phase, in contrast, hid the inquiry support tools so that students could focus specifically on the simulation. This gave students the opportunity to explore the microworld if they were not yet ready to formulate a hypothesis. Finally, in the “analyze” phase, students were shown the data they collected and used the data analysis tool to construct an argument based on their data to support or refute their hypotheses. In the current version of the phase change environment, no feedback was given to the students about the quality or correctness of their inquiry processes. One goal of this work is to develop automatic detectors for evaluating when students are haphazard in their inquiry (Buckley, Gobert & Horwitz, 2006; Buckley, et al., 2010) to enable the system to provide helpful feedback.

We designed this environment to enable a moderate degree of student control, less than in purely exploratory learning environments (Amershi & Conati, 2009), but more than in classic model-tracing tutors (Koedinger & Corbett, 2006) or constraint-based tutors (Mitrovic, Mayo, Suraweera, & Martin, 2001). In particular, as already mentioned, students had some freedom to navigate between inquiry phases (Figure 1) and had flexibility within each phase to conduct many actions. For example, while in the hypothesizing phase (Figure 2), students could elect to explore the simulation more before formulating any hypotheses by moving to the “observe” phase. Alternatively, they could choose to specify one or more hypotheses like, “If I change the container size so that it increases, the melting point stays the same” before collecting data. Within the “experiment” phase (Figure 3), students could run as

many experiments as they wished to collect data for any one or all of their hypotheses. Within the "analysis" phase students also had several options. As they constructed their claims, students could decide to go back and collect more data or, after constructing claims based on their data, they could decide to create additional hypotheses, thus starting a new inquiry loop. Students conducted these inquiry activities in various patterns, engaging in inquiry in many different ways.

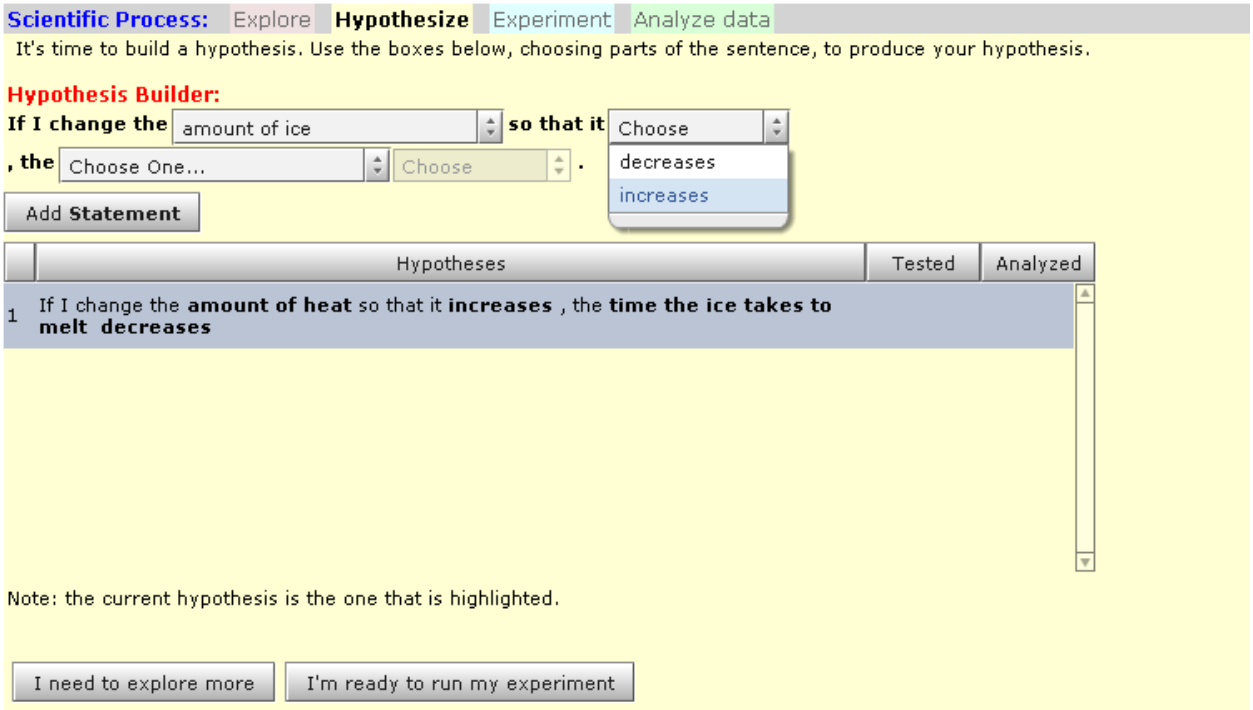

Figure 2. Hypothesizing tool for the Phase Change microworld.

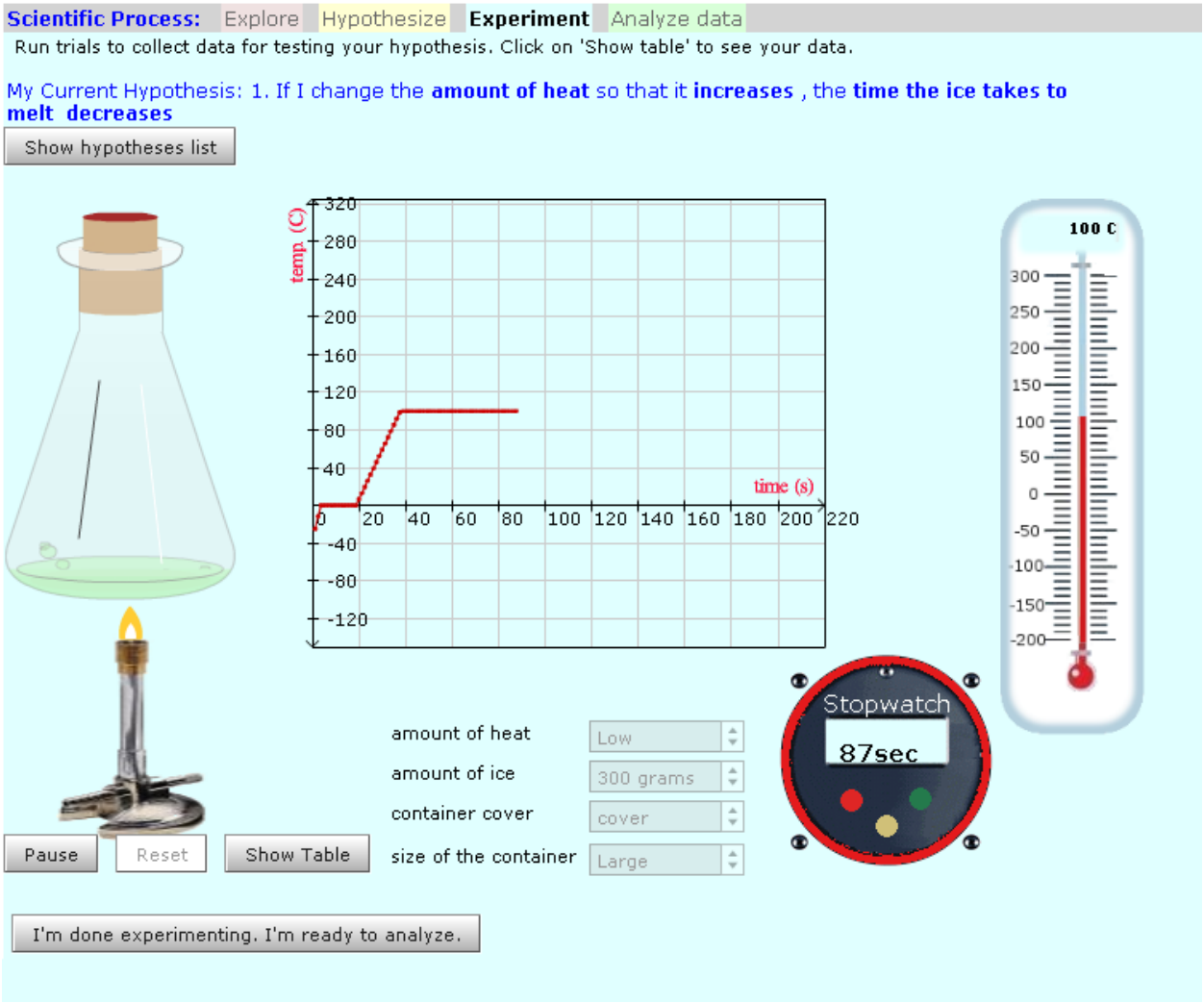

Figure 3. Data collection example for the Phase Change microworld.

Since the environment currently does not provide feedback on students' inquiry processes, students could engage in either systematic or haphazard inquiry behavior. Specific to the "hypothesize" and "experiment" phases, students acting in a systematic manner (Buckley, et al., 2010) collect data by designing and running controlled experiments that test their hypotheses. They also may use the table tool and hypothesis viewer in order to reflect and plan for additional experiments. These systematic behaviors are representative of the "designing and conducting experiments" skills (National Research Council, 1996) we aim to assess with machine-learned detectors. In contrast, students acting

haphazardly in our environment may construct experiments that do not test their hypotheses, not collect enough data to support or refute their hypotheses, design confounded experiments, fail to use the inquiry support tools to analyze their results and plan additional trials (cf. de Jong, 2006), or collect data for the same experimental setup multiple times (Buckley, Gobert & Horwitz, 2006; Buckley, et al., 2010). Within this approach, we focus on detecting appropriate systematic inquiry behaviors, rather than specific haphazard inquiry behaviors, to enable assessment of skills students possess, rather than how they fail.

3.2.2 Transfer Assessments

To investigate the degree to which our automated detectors capture knowledge that transfers outside of the phase change environment studied, we developed three transfer assessment batteries. These instruments measure students' understanding of hypotheses and designing controlled experiments (cf. National Research Council, 1996), skills aligned with the inquiry behaviors modeled in this paper. These assessments provided a way to validate our proficiency estimate models of skill within the phase change environment (cf. Corbett & Anderson, 1995; Beck & Mostow, 2005). They also allowed us to study the relationships between our measures of authentic inquiry performance and more traditional measures of inquiry knowledge (cf. Black, 1999; Pellegrino, 2001).

Two assessments utilized multiple choice items, an approach involving items similar to those seen in standardized paper tests of inquiry (cf. Alonzo & Aschbacher, 2004; Gotwals & Songer, 2006). These items came from several sources: our team, an inquiry battery developed by a middle school science teacher, and an assessment battery on designing controlled experiments developed by Strand-Cary and Klahr (2009). Items were chosen to be as domain-neutral as possible. The first multiple-choice assessment contained 6 items and measured understanding of hypotheses. These items required

students to identify independent (manipulated) variables, dependent (outcome) variables, and a testable hypothesis for different cover stories. The second multiple-choice assessment contained 4 items and measured understanding of controlled experiments. The first item required students to identify the Control of Variables Strategy procedure (cf. Chen & Klahr, 1999). The remaining three items required students to identify the appropriate controlled experiment that makes it possible to test a specific variable's effects on an outcome.

Our third assessment, the ramp transfer test, was designed specifically to measure students' authentic skill at designing controlled experiments. This assessment required students to construct four unconfounded experiments within a different domain, a ramp microworld on determining which factors (ramp surface, ball type, ramp steepness, and ball start position) would make a ball roll further down a ramp (Sao Pedro, Gobert, Heffernan & Beck, 2009; Sao Pedro, Gobert & Raziuddin, 2010). For each item, two ramp apparatuses in an initially confounded setup were presented. Students were asked to change the ramp setups in order to test the effects of a given target variable (e.g. ramp surface). A setup was evaluated as correct if they correctly contrasted the target variable while keeping all other extraneous variables the same.

Though the ramp transfer test and multiple choice tests on designing controlled experiments attempt to measure the same skill, their formats are quite different. Therefore, we did not combine scores from the two tests to form a single measure of the skill. This choice also enabled us to analyze if authentic inquiry skill in one domain predicts skill in another domain (the ramp environment) separately from our analysis predicting performance at answering multiple choice questions involving that same skill.

3.3 Procedure

All students used the Science Assistments system (Gobert, et al., 2007; Gobert, et al., 2009) to engage in the phase change learning activities and transfer assessments. The entire procedure occurred over two class periods, about 1.5 hours in total. The order of activities was as follows:

- *Orientation Activities:* These activities prepared students for inquiry within the phase change environment. They included a primer on the relevant scientific vocabulary, exploration time with the microworld that let them run as many experiments as they liked without the inquiry support tools, and a planning step where they explained in their own words how to use the simulation to conduct experiments.
- *Phase Change Activities (Section 3.2.1):* Four inquiry activities were administered using the phase change environment with inquiry support tools. Each activity asked students to test how a particular independent variable (e.g. container size) affected all the dependent measures. Students then engaged in several inquiry loops to address the goal. A different goal was given in each of the four activities, one for each independent variable. However, students could choose to ignore the current goal and test any hypothesis and conduct as many different kinds of experiments as desired.
- *Transfer Assessments (Section 3.2.2)*: Finally, students took the three transfer assessments in which their skills at designing controlled experiments and testing hypotheses were measured.

During this time, Science Assistments logged all students' interactions within the phase change environment as the student engaged in inquiry. In the following sections, we show how we used the low-level interaction data to construct and validate machine-learned detectors of systematic data collection behavior. Then, we describe how to leverage the detectors to estimate skill within the phase change environment, and predict performance on transfer tests requiring these skills.

4 Building and Validating Behavior Detectors using Text Replay Tagging

Within this section, we discuss our work to build machine-learned detectors that determine if students are systematic in their data collection. In particular, we built detectors of four specific data collection behaviors: testing hypotheses, designing controlled experiments, and planning further experiments using the table tool or hypothesis viewer. In section 5, we discuss our work to use these detectors to infer the latent inquiry skills associated with these behaviors.

In this approach, we utilized "text replay tagging" to enable human coders to classify clips, textual sequences of low-level student actions gleaned from log files, within the phase change environment. Text replay tagging is an extension of text replay approach originally developed in Baker, Corbett and Wagner (2006). These authors showed that text replays can achieve good inter-rater reliability, and that text replays agree well with predictions made by models generated using data from quantitative field observations. This approach is also similar to that of Cocea and Weibelzahl (2009) in that behavior was labeled over unprocessed log files. In text replays and text replay tagging, human coders are presented "pretty-printed" versions of log files with important data features emphasized to simplify the coding process. Text replay tagging also differs from these other approaches in that they only permit labeling a clip with a single category. Text replay tagging, on the other hand, allows multiple tags to be associated with one clip. For example, within our domain, a clip may be tagged as involving designing controlled experiments, involving testing the correct hypothesis testing, both, or neither.

After producing these classifications, each student's activity sequences were summarized by creating a feature set from the data. Classification methods were then used to find models over the feature set that predict the labels from the data. In accordance with our data, we consider features

aggregated over significant portions of students' inquiry, such as the experimental setups students designed within an activity, rather than step or transaction-level data, unlike in many prior models of student behavior (e.g. Beck, 2005; Walonoski & Heffernan, 2006; Baker & de Carvalho, 2008; Amershi & Conati, 2009; Cetintas, Si, Xin, & Hord, 2010; Baker, Mitrovic, & Mathews, 2010).

There are several steps in generating and validating detectors using text replay tagging. The key steps are: defining low-level user interface actions, deciding how to generate clips for those actions, defining suitable behavior tags, deciding how to display clips for a human coder to classify the clips, and determining an appropriate feature set over which the machine learning algorithm will generate a model. We discuss next how we applied this process to generate and assess the goodness of our detectors.

4.1 Fine-Grained Student Logs

All students' fine-grained actions were recorded as students engaged in inquiry within the Phase Change microworld activities. An example of an unprocessed log file for a student is shown in Table 1. In developing our models, we looked specifically at student actions from the "hypothesize" and "experiment" phases of inquiry because this is where systematic data collection behaviors occur. Logged actions included low-level widget interactions from creating hypotheses, designing experiments, showing or hiding support tools (the data table or hypothesis list), running experiments, and transitioning between inquiry activities (i.e. moving from hypothesizing to experimenting). Looking more deeply, the following data were recorded for each action:

- *Action:* A unique action ID
- *Time:* The action's timestamp, in milliseconds

- *Activity:* The unique ID of the activity in which the action took place
- *Student:* The ID of the student working on the activity
- *Widget:* A unique name associated with a graphical widget / system component associated with the activity
- *Who:* The entity who initiated the action, the student or the system
- *Variable:* The unique aspect of the inquiry problem that the widget / system component changes. Examples include individual components of the hypothesis and values that can be changed for the Phase Change simulation.
- *Value:* current value for the variable, if applicable.
- *Old Value*: previous value for the variable, if applicable.
- *Step Name*: A unique marker describing the action taken by the user. This is akin to a problem solving step in Cognitive Tutors (cf. Corbett & Anderson, 1995; Koedinger & Corbett, 2006). In particular, this information helped simplify and standardize the development of our clip generation and feature distillation software.

To give a concrete example, action 62955 in Table 1 indicates that the student changed the value of the "Level of heat" variable from "Low" to "Medium". In all, 27,257 unique student actions for phase change were logged. These served as the basis for generating clips for our domain, contiguous sequences of actions specific to experimenting and data collection.

Table 1. Unprocessed log file segment for a student engaging in inquiry within a single activity.

Action	Time	Activity	Student	Widget	Who	Variable	Value	Old Value	Step Name
62934	...5669	147212	85240	variable_containerSize	system	Container Size	Large	null	INIT_SET_IV
62935	...5669	147212	85240	variable_coverStatus	system	Cover Status	Cover	null	INIT_SET_IV
62936	...5669	147212	85240	variable_substanceAmount	system	Amount of Substance	300 grams	null	INIT_SET_IV
62937	...5669	147212	85240	variable_heatLevel	system	Level of heat	Low	null	INIT_SET_IV
62938	...5684	147212	85240	hypothesis.iv	student	iv	Level of heat	null	SPECIFY_IV_HYPOTHESIS
62939	...5691	147212	85240	hypothesis.iv.dir	student	iv.heatLevel.direction	increases	null	SPECIFY_IV_DIRECTION_HYPOTHESIS
62940	...5704	147212	85240	hypothesis.iv.dir	student	iv.heatLevel.direction	decreases	increases	SPECIFY_IV_DIRECTION_HYPOTHESIS
62941	...5707	147212	85240	hypothesis.dv	student	heatLevel.dv	time to melt	null	SPECIFY_DV_HYPOTHESIS
62942	...5722	147212	85240	hypothesis.iv.dir	student	iv.heatLevel.direction	increases	decreases	SPECIFY_IV_DIRECTION_HYPOTHESIS
62943	...5731	147212	85240	hypothesis.dv.dir	student	heatLevel.dv.timeMelting.direction	decreases	null	SPECIFY_DV_DIRECTION_HYPOTHESIS
62944	...5737	147212	85240	hypothesis.add	student				ADD_HYPOTHESIS
62945	...5740	147212	85240	stage:hypothesize->experiment	student				CHANGE_STAGE_HYPOTHESIZE_EXPERIMENT
62946	...5757	147212	85240	variable_heatLevel	student	Level of heat	High	Low	CHANGE_IV
62947	...5760	147212	85240	variable_heatLevel	student	Level of heat	Low	High	CHANGE_IV
62948	...5763	147212	85240	run	student				RUN
62949	...5763	147212	85240	PhaseTable.cvs.column	student				SELECT_TRIALS
62950	...5764	147212	85240	simulation	system	state	run		
62951	...5777	147212	85240	simulation	system	state	complete		SIM_COMPLETE
62952	...5781	147212	85240	reset	student				REVERT_IVS
62953	...5781	147212	85240	simulation	system	state	reset		
62954	...5781	147212	85240	simulation	system	state	ready		
62955	...5784	147212	85240	variable_heatLevel	student	Level of heat	Medium	Low	CHANGE_IV
62956	...5786	147212	85240	run	student				RUN
62957	...5786	147212	85240	PhaseTable.cvs.column	student				SELECT_TRIALS
62958	...5787	147212	85240	simulation	system	state	run		
62959	...5793	147212	85240	simulation	system	state	complete		SIM_COMPLETE
62960	...5797	147212	85240	reset	student				REVERT_IVS
62961	...5797	147212	85240	AppManager	system	state	reset		
62962	...5797	147212	85240	AppManager	system	state	ready		
62963	...5807	147212	85240	variable_heatLevel	student	Level of heat	High	Medium	CHANGE_IV
62964	...5809	147212	85240	run	student				RUN
62965	...5809	147212	85240	simulation	system	state	run		
62966	...5814	147212	85240	simulation	system	state	complete		SIM_COMPLETE
62967	...5818	147212	85240	showDataTable	student				DATA_TABLE_DISPLAY
62968	...5854	147212	85240	showHypotheses	student				HYPOTHESES_LIST_DISPLAY
62969	...5880	147212	85240	showHypotheses	student				HYPOTHESES_LIST_DISPLAY
62970	...5884	147212	85240	stage:experiment->analyze	student				CHANGE_STAGE_EXPERIMENT_ANALYZE
					… Actions for Analysis Phase of Inquiry…				
62982	...5979	147212	85240	stage:analyze->hypothesize	student				CHANGE_STAGE_ANALYZE_HYPOTHESIZE
62983	...5979	147212	85240	HypTable.analyzed.column	student	row:1	true	null	
62984	...5996	147212	85240	stage:hypothesize->experiment	student				CHANGE_STAGE_HYPOTHESIZE_EXPERIMENT
62985	...6004	147212	85240	stage:experiment->analyze	student				CHANGE_STAGE_EXPERIMENT_ANALYZE
					… Actions for Analysis Phase of Inquiry…				
62988	...6007	147212	85240	submit	student				

4.2 Generating Clips and Text Replays

Clips were composed of contiguous sequences of fine-grained actions from the "hypothesize" and "experiment" phases. Clips could begin at different points, depending on how students navigated through inquiry phases (see Figure 3 for allowable phase transitions). First, a clip could begin at the start of a full inquiry loop when a student enters the "hypothesize" phase. This phase could be entered in two ways, either by starting the activity, or by choosing to create more hypotheses in the "analyze" phase, thereby starting a new inquiry loop. A clip could also begin in the middle of a full inquiry loop if a student chose to go back to the "experiment" phase to collect more data while in the "analyze" phase. A clip always ended when the "experiment" phase was exited. As an example, the action sequence for the student's activity shown in Table 1 yielded two clips, one clip containing actions 62934 through 62970, and another containing actions 62982 through 62985 when the student navigated back to the "hypothesize" phase from the "analyze" phase. This clip generation procedure yielded 1,503 clips from the database of all student actions.

In designing text replays to display clips for tagging, it was necessary to use coarser grain-sizes than prior versions of this method that delineated clips on a pre-specified length of time (e.g. Baker & de Carvalho, 2008; Baker, Mitrovic & Mathews, 2010). In particular, we found it necessary to show significant periods of experimentation so that human coders could precisely evaluate experimentation behavior relative to students' hypotheses. For example, it would be difficult to accurately tag a clip in which a student transitioned from "analyze" back to "experiment" without seeing students' associated actions from the previous "hypothesize" phase. As another example, data collected in a previous inquiry cycle were sometimes utilized by students to test a hypothesis in a later inquiry cycle; not seeing these previous could lead to incorrect tagging. Without showing the full history for coding, it would not be possible for coders to recognize the student's sophisticated inquiry behavior. To compensate, our text

replays contained clips representing the actions for testing the current hypothesis, and cumulative data including actions performed when testing previous hypotheses or collecting previous data. In other words, we coded clips while taking into account actions from earlier clips from the same activity.

4.3 Text Replay Tag Definitions

The next step was to define possible tags, or classification labels, that could be applied to clips. Nine tags were identified, corresponding to systematic and haphazard data collection behaviors of interest. In line with the text replay tagging approach, any or all of these tags could be used to classify a clip. These tags were: “Designed Controlled Experiments”, “Tested Stated Hypothesis”, “Used Data Table to Plan”, “Used Hypothesis List to Plan”, “Never Changed Variables”, “Repeat Trials”, “Non-Interpretable Action Sequence”, “Indecisiveness”, and “No Activity”. These were chosen based on systematic and haphazard behaviors identified in previous work on inquiry-based learning (cf. Gobert et al., 2006; de Jong, 2006; Buckley et al., 2010). We also added one extra category for unclassifiable clips, “Bad Data”, for a total of 10 tags.

As previously mentioned, our analyses focused on four behaviors associated with data collection skill of particular theoretical importance. The four corresponding tags are: “Designed Controlled Experiments”, “Tested Stated Hypothesis”, “Used Data Table to Plan”, and “Used Hypothesis List to Plan”. We tagged a clip as “Designed Controlled Experiments” if the clip contained actions indicative of students trying to isolate the effects of one variable. “Tested Stated Hypothesis” was chosen if the clip had actions indicating attempts to test one or more of the hypotheses stated by the student, regardless of whether or not the experiments were controlled. We tagged a clip as “Used Data Table to Plan” if the clip contained actions indicative that the student viewed the trial run data table in a way consistent with

planning for subsequent trials. Finally, “Used Hypothesis List to Plan” was chosen if the clip had actions indicating that the student viewed the hypotheses list in a way consistent with planning for subsequent trials.

4.4 Clip Tagging Procedure

To support coding in this fashion, we developed a new tool for text replay tagging (Figure 4). This tool, implemented in Ruby, enabled the classification of clips using any combination of the tags defined in Section 4.3. The tool displays one text replay at a time, consisting of the current clip and all relevant predecessor clips. Within our approach, a human coder chooses at least one but possibly several tags to classify the clip.

Two human coders, the first and fifth authors, tagged a subset of the data collection clips to generate a corpus of hand-coded clips for training and validating our detectors. The subset contained one randomly chosen clip (e.g. first clip, second clip, etc.) for each student-activity pair, resulting in 571 clips. This ensured a representative range of student clips were coded. The human coders tagged the same first 50 clips to test for agreement; the remaining clips were split for each to code separately. Each coder independently tagged about 260 clips each in three to four hours.

Agreement for the 50 clips tagged by both coders was high overall. Since each clip could be tagged with one or several tags, agreement was determined by computing separate Cohen’s Kappa values for each tag. Over all ten tags, there was an average agreement of $\kappa = 0.86$. Of specific importance to this work, there was good agreement on the designing controlled experiments, $\kappa = .69$. There was perfect agreement between coders for testing the stated hypothesis ($\kappa = 1.00$), planning using the data table ($\kappa = 1.00$), and planning using the hypothesis list ($\kappa = 1.00$). The high degree of

agreement was achieved in part through extensive discussion and joint labeling prior to the inter-rater reliability session. Even though the agreement on designing controlled experiments was lower than the other behaviors, all Kappas were at least as good as the Kappas seen in previous text replay approaches leading to successful behavior detectors. For example, Baker, Corbett, & Wagner (2006) reported a Kappa of .58, and Baker, Mitrovic and Mathews (2010) reported a Kappa of .80 when labeling "gaming the system" behavior in clips from two different learning environments.

The human coders tagged 31.2% of the clips as showing evidence of designing controlled experiments; 34.4% were tagged as showing evidence of collecting data to test specified hypotheses. Planning behaviors involving the data table and hypothesis list were relatively rarer. Only 8.2% and 3.5% of the clips were tagged as exhibiting planning using the data table tool and hypothesis list viewer, respectively.

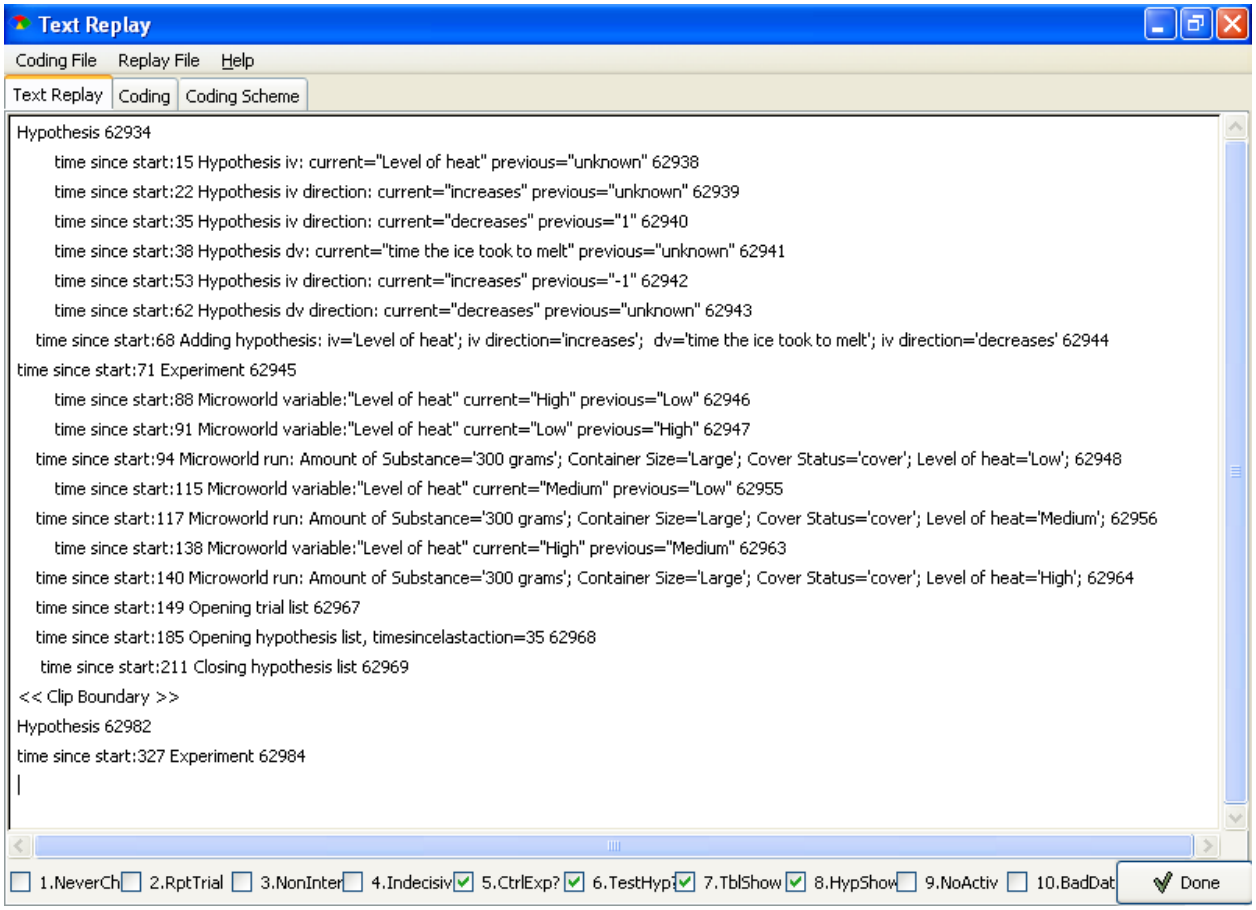

Figure 4. Text Replay Tagging Tool with an example text replay corresponding to the action sequence displayed in Table 1. This clip, the second clip generated for the activity, was tagged as involving designing controlled experiments, testing stated hypotheses, and using the data table and hypothesis list to plan which experiments to run next.

4.5 Feature Distillation

Next, a set of features were distilled that, when combined with the classification labels from text replay tagging, yielded data instances for machine learning. Seventy-three features were selected based on features used in previous detectors of other constructs (e.g. Walonoski & Heffernan, 2006; Baker, et al., 2008), and previous work that identified indicators of systematic and haphazard inquiry behavior (e.g. de Jong, 2006; Buckley, Gobert & Horwitz, 2006; Gobert, et al., 2010). The classes of features,

summarized in Table 2, included: variables changed when making hypotheses, hypotheses made, total trials run, incomplete trials run (runs in which the student paused and reset the simulation), complete trials run (runs in which the student let the simulation run until completion), simulation pauses, data table displays, hypothesis list displays, variable changes made when designing experiments, and the total number of all actions (any action performed by a student). For each feature class, we computed a count each time the action was taken as well as timing values, similar to approaches taken by Walonoski and Heffernan (2006) and Baker et al. (2008). Timing features included the minimum, maximum, standard deviation, mean and median. We also included a feature for the activity number associated with the clip since students may exhibit different behaviors for each of the activities.

Two additional feature counts specifically related to systematic data collection were also computed. The first was a unique pairwise controlled trials count using the Control of Variables Strategy (CVS), a count of the number of unique trials in which only one factor differed between them. This was similar to the approach taken by McElhaney and Linn (2010) to assess CVS, except they computed a pairwise CVS count for *adjacent* trials (i.e. trial *n* and trial *n+1* demonstrate CVS). Our count, on the other hand, tallied any pairwise CVS trials. This choice was made because students had the opportunity to view their previous trials in the data table, and could judge if they had adequate data or needed to run more trials. The second feature was a repeat trial count (Buckley, Gobert & Horwitz, 2006; Gobert, et al., 2010), the total number of trials with the same independent variable selections. These authors hypothesized that repeating trials is indicative of haphazard inquiry. It is worth noting that repeat trials were not included in the CVS count; that count only considered unique trials.

We computed feature values at two different levels of granularity. Recall that within a phase change activity, a student could make and test several hypotheses, thus generating multiple clips for that activity. Feature values could thus be computed *locally*, considering only the actions within a single

clip, or *cumulatively*, taking into account all actions within predecessor clips for an activity. For each data instance, we computed two values for each feature, one local and one cumulative, with the aim of comparing the effectiveness of each feature set in predicting data collection behavior. An excerpt of the dataset used to generate and validate the detectors is shown in Table 3.

Table 2. Summary of all 73 distilled features used to build machine-learned detectors.

		Time information					
Feature Classes	Count	Total	Min	Max	Mean	SD	Median
All actions	X	X	X	X	X	X	X
Hypothesis variable changes	X	X	X	X	X	X	X
Hypotheses added	X	X	X	X	X	X	X
Data table use	X	X	X	X	X	X	X
Hypothesis list use	X	X	X	X	X	X	X
Simulation variable changes	X	X	X	X	X	X	X
Simulation pauses	X	X	X	X	X	X	X
Total trials run	X	X	X	X	X	X	X
Incomplete trials run	X	X	X	X	X	X	X
Complete trials run	X	X	X	X	X	X	X
Repeated trials	X						
Unique pairwise CVS trials	X						
Activity number	X						

Table 3. Example instances used for machine learning with values for local (Loc) attribute and cumulative (Cu) attributes. The row in boldface and italics corresponds with the clip coded via text replay tagging in Figure 4.

Student	Clip	Activity	Loc: Action Count	Loc: Action Total Time	…	Loc: Repeat Trial Count	Cu: Action Count	Cu: Action Total Time	…	Ctrlled Exps?	Test Hyps?	Plan Data	Plan Hyp List?
…													
81850	2	3	7	120		0	15	155		N	Y	N	N
81843	2	1	0	0		0	7	69		N	N	N	N
85240	***2***	***1***	***0***	***0***	…	***0***	***22***	***208***	…	***Y***	***Y***	***Y***	***Y***
78382	2	3	13	79		0	33	224		Y	Y	N	N
85238	2	3	8	145		0	16	219		N	Y	N	N
…													

4.6 Detector Generation and Validation Approach

Machine-learned detectors were generated and validated using the corpus of hand-coded clips and summary features. They were developed within RapidMiner 4.6 (Mierswa, Wurst, Klinkenberg, Scholz, & Euler, 2006) using the following procedure. First, redundant features correlated to other features at or above 0.6 were removed. Then, detectors were constructed using J48 decision trees with automated pruning to control for over-fitting. More specifically, two algorithm parameters were set to control for over-fitting; the minimum number of instances per leaf (*M*) was set to 2, and the confidence threshold for pruning (*C*) was set to 0.25.

This technique was chosen for three reasons. First, J48 decision trees have led to successful behavior detectors in previous research (Walonoski & Heffernan, 2006; Baker & de Carvalho, 2008). Second, decision trees produce relatively human-interpretable rules. Finally, such rules can be easily integrated into our environment to assess student behavior, and accordingly inquiry skills, in real-time. Six-fold cross-validation was conducted at the student level, meaning that detectors were trained on five randomly selected groups of students and tested on a sixth group of students. By cross-validating at this level, we increase confidence that detectors will be accurate for new groups of students.

Detectors were assessed using two metrics, A’ (Hanley & McNeil, 1982) and Kappa. A' is the probability that if the detector is comparing two clips, one involving the category of interest (designing controlled experiments, for instance) and one not involving that category, it will correctly identify which clip is which. A' is equivalent to both the area under the ROC curve in signal detection theory, and to W, the Wilcoxon statistic (Hanley & McNeil, 1982). A model with an A' of 0.5 performs at chance, and a model with an A' of 1.0 performs perfectly. In these analyses, A’ was computed at the level of clips, rather than students, using the AUC (area under the curve) approximation. Statistical tests for A’ are not presented in this paper. An appropriate statistical test for A’ in data across students would be to

calculate A' and standard error for each student for each model, compare using *Z* tests, and then aggregate across students using Stouffer's method (cf. Baker, Corbett, & Aleven, 2008). However, the standard error formula for A' (Hanley & McNeil, 1982) requires multiple examples from each category for each student, which is infeasible in the small samples obtained for each student (a maximum of four) in our text replay tagging. Another possible method, ignoring student-level differences to increase example counts, biases undesirably in favor of statistical significance.

Second, we used Cohen's Kappa (κ), which assesses whether the detector is better than chance at identifying the correct action sequences as involving the category of interest. A Kappa of 0 indicates that the detector performs at chance, and a Kappa of 1 indicates that the detector performs perfectly.

A' and Kappa were chosen because, unlike accuracy, they attempt to compensate for successful classifications occurring by chance (cf. Ben-David, 2008). Thus, we can achieve a better sense of how well our detectors can classify given our corpus' unbalanced labels, with between 4% and 34% of instances labeled as positively demonstrating one of the behaviors. We note that A' can be more sensitive to uncertainty in classification than Kappa, because Kappa looks only at the final label whereas A' looks at the classifier's degree of confidence in classifying an instance.

4.7 Analysis of Machine-Learned Classifiers

In our analyses, we determined if machine-learned detectors could successfully identify the four data collection behaviors of interest, testing hypotheses, designing controlled experiments, planning with the hypothesis list and planning with the data table. As part of this goal, we compared whether detectors built with features computed using the current and all predecessor clips (cumulative features) achieved better prediction than those built with features looking solely at the current clip (local features), as

measured by A' and Kappa. We hypothesized that cumulative features would yield better detectors, because the additional information from previous clips may help more properly identify systematic behavior. For example, since students could re-use previous trials to test new hypotheses, actions in subsequent clips may be more abbreviated. Thus, taking into account previous actions could provide a richer context to identify and disambiguate behavior.

Separate detectors for each behavior were generated from each feature set, resulting in eight different detectors. Separate detectors were built for each behavior, as opposed to one detector to classify all behaviors, for two reasons. First and most important, the different behaviors were not necessarily mutually exclusive; they could be demonstrated simultaneously in each clip. Building a single classifier only would allow for finding a *single* behavior within a clip. Second, the number of instances available for training and testing were slightly different for each behavior. This occurred, because for a small set of clips in the corpus that both human coders tagged, there was disagreement as evidenced by the imperfect inter-rater reliability measures (see Section 4.4). Within that set, we only used clips where the two coders were in agreement for a specific behavior.

The confusion matrices capturing raw agreement between each detector's prediction and the human coders' tagging under student-level cross-validation are shown in Table 4. Overall, we found that detectors of three of the four behaviors were quite good overall and that there were no major differences between detectors built using cumulative versus noncumulative attributes as measured by A' and Kappa. The designing controlled experiments detector using cumulative attributes (A' = .85, κ = .47) performed slightly better than the detector built with non-cumulative attributes (A' = .81, κ = .42). The hypothesis testing detector built with cumulative attributes (A' = .85, κ = .40) had a slightly higher A', but slightly lower Kappa than the non-cumulative detector (A' = .84, κ = .44). The planning using the table tool detector using cumulative attributes (A' = .94, κ = .46) had higher Kappa and A' than its non-

cumulative counterpart (A' = .83, κ = .31). We do note, however, that these detectors appear to bias towards inferring that a student is not demonstrating skill. This is indicated by recall values ranging from 51% to 63% for cumulative attribute-based detectors and 30% to 53% for local attribute-based detectors (shown in Table 4). Thus, these detectors are most appropriate for use in fail-soft interventions, where students assessed with low confidence (in either direction) can receive interventions that are not costly if misapplied. Overall, the performance of these detectors as measured by A' and Kappa, is comparable to detectors of gaming the system refined over several years (e.g., Baker & de Carvalho, 2008; Baker, Mitrovic & Mathews, 2010). Therefore, we can also use the detectors to automatically classify the remaining clips to obtain a full profile of students' data collection behavior.

Detectors of planning using the hypothesis list did not perform as well, achieving A' = 0.93, κ = 0.14 for the non-cumulative attributes and A'=.97, κ = 0.02 for the cumulative attributes. The substantial difference between A' and Kappa is unusual. It appears that what happened in this case is that the model, on cross-validation, classified many clips incorrectly with low confidence. In other words, A' catches the overall rank-ordered correctness of the detector across confidence values by considering pairwise comparisons, even though many clips were mis-categorized at the specific threshold chosen by the algorithm. One possibility is that the low number of positive tags for this behavior made the detectors more prone to over-fitting. Though generally not satisfactory, the value of A' suggests that these detectors may still be acceptable for fail-soft interventions.

Table 4. Confusion matrices for each behavior's cumulative and non-cumulative attribute-based detector tested under six-fold student-level cross-validation.

		Controlled Exps?		Test Hyps?		Plan Data Table?		Plan Hyp List?	
		True N	True Y	True N	True Y	True N	True Y	True N	True Y
Cumulative Features	Pred N	325	65	303	79	503	23	539	19
	Pred Y	63	111	71	117	20	24	11	1
		A'=.85, K=.47		A'=.85, K=.40		A'=.94, K=.46		A'=.97, K=.02	
		Pc=.64, Rc=.63		Pc=.62, Rc=.60		Pc=.55, Rc=.51		Pc=.08, Rc=.05	
		True N	True Y	True N	True Y	True N	True Y	True N	True Y
Local Features	Pred N	343	87	331	93	511	33	545	18
	Pred Y	45	89	43	103	12	14	5	2
		A'=.81, K=.42		A'=.84, K=.44		A'=.83, K=.31		A'=.93, K=.14	
		Pc=.66, Rc=.51		Pc=.71, Rc=.53		Pc=.54, Rc=.30		Pc=.29, Rc=.10	

Note: Pc = Precision, Rc = Recall

4.8 Final Models of Data Collection Behavior

In this section, we show the final models of each behavior, generated using all hand-coded clips. We focus on the three detectors which performed well under cross-validation (designing controlled experiments, testing hypotheses, and planning using the data table), as these detectors are good enough to use in estimating skill and predicting transfer (section 5). These detectors are built cumulative features, due to the slightly better performance obtained using this method.

Prior to constructing these final detectors, we again removed correlated features at the 0.6 level, reducing the number of features from 73 to 25. Most features were time-based, though some count-based features remained: repeat trials, incomplete runs, hypotheses added, data table and hypothesis list uses, and the total number of actions. Additionally, how many activities the student had

completed so far remained. These features were used to construct the three decision trees for each behavior. The resulting trees for the designing controlled experiments and testing hypotheses detectors were wider and more complex than the tree for planning using the table tool detector. Portions of the decision trees for the designing controlled experiments detector and planning using the table tool detector are shown in Figures 5 and 6, respectively.

The designing controlled experiments tree had 46 leaves and 91 nodes and used nearly all of the features remaining after correlation filtering. The root note, as shown in Figure 5, was “median time spent changing simulation variables” feature, and indicated that if variables were not changed (i.e. median time = 0), then the clip did not exhibit behavior in line with designing controlled experiments (in this case, no experiments were designed at all). The next level down branched on “minimum time running trials <= 1”, a proxy for the number of runs. If this value were zero, it would mean that the clip contained no simulation runs, an indicator that the clip did exhibit the behavior. However, if this value were one, it would mean that simulation runs occurred, but in at least one case was very quick, one second or less. This potentially indicates a student realizing the trial is unnecessary, in turn a potential indicator of inquiry skill. As a result of this complexity, hierarchy under this branch is complex and tries to disentangle if the clip exhibits systematic behavior using several features primarily associated with simulation runs and microworld simulation variable changes. On the other hand, if the minimum time running trials was greater than 1, the lower branches involved time features associated with running simulations, the number of hypotheses made, the number of times the data table was displayed, and the median time for all actions. Overall, the tree is predominantly based on time features associated with number of runs, number of actions, and simulation variable changes to distinguish behavior. It is worth noting, incidentally, that this tree did not contain the pairwise unique CVS count feature since it was pruned during the “remove correlated features” step. It could be that the tree would be more compact had this feature been considered. However, the features included captured the designing

controlled experiments behavior successfully, given the good cross-validated A' and Kappa achieved by this detector.

The decision tree for the planning using the table tool detector was narrow, but deep and utilized fewer features than the designing controlled experiments detector. It had 18 leaves and 35 nodes. The root note was the "mean time viewing the data table" feature. At this level, if the mean time viewing the table tool was less than 1.4 seconds, the detector labeled the clip as not exhibiting the behavior which is in line with our expectation of this behavior. In fact, the mean time and standard deviation of time viewing the table tool appeared several times in the decision tree hierarchy, including at leaves. In these cases, larger values for these features were associated with positive demonstration of the behavior in a clip.

These decision trees were used to classify the remaining unseen clips to assess each student over all activities. In the next section, we describe how we leveraged these assessments to build estimates of each skill and used those estimates to predict performance on our transfer tests.

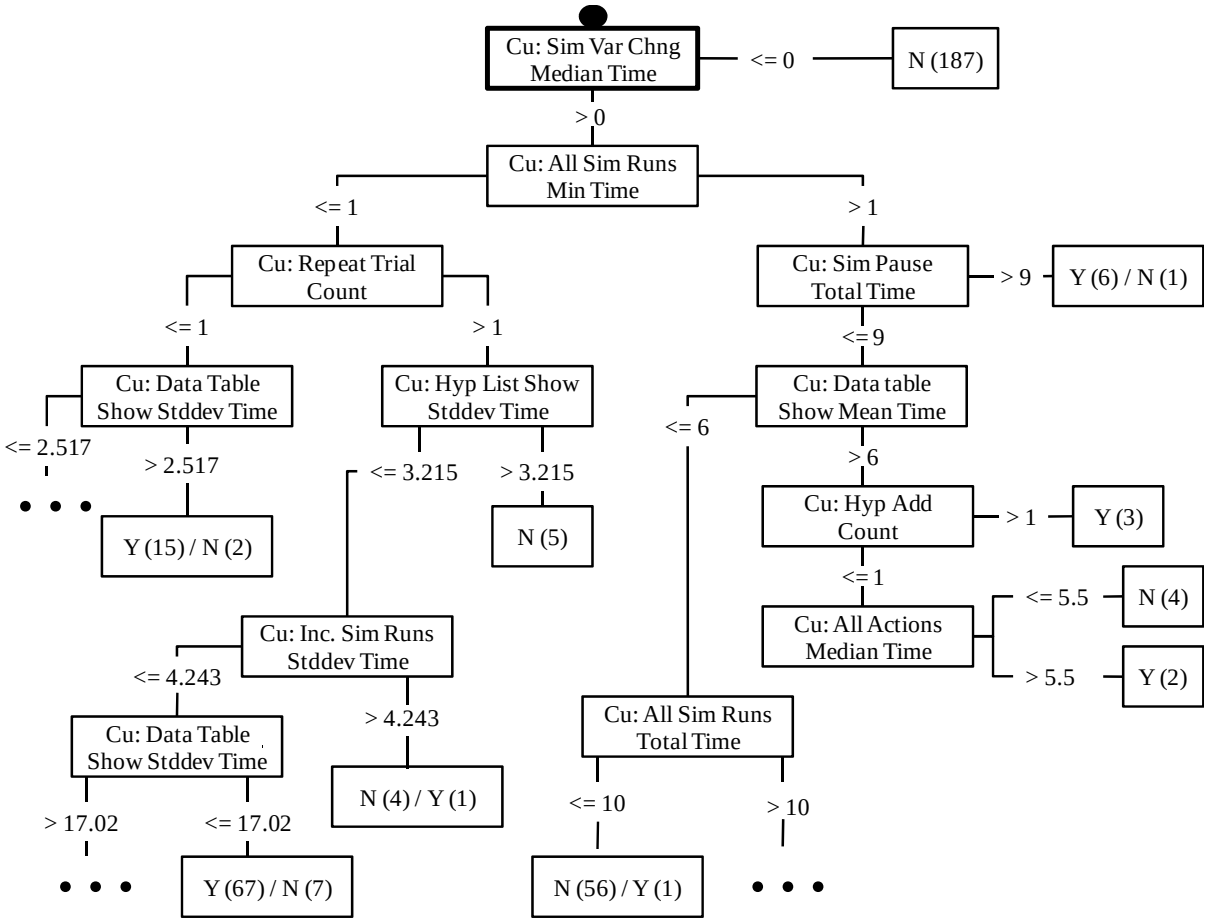

Figure 5. Portion of the decision tree for the designing controlled experiments behavior.

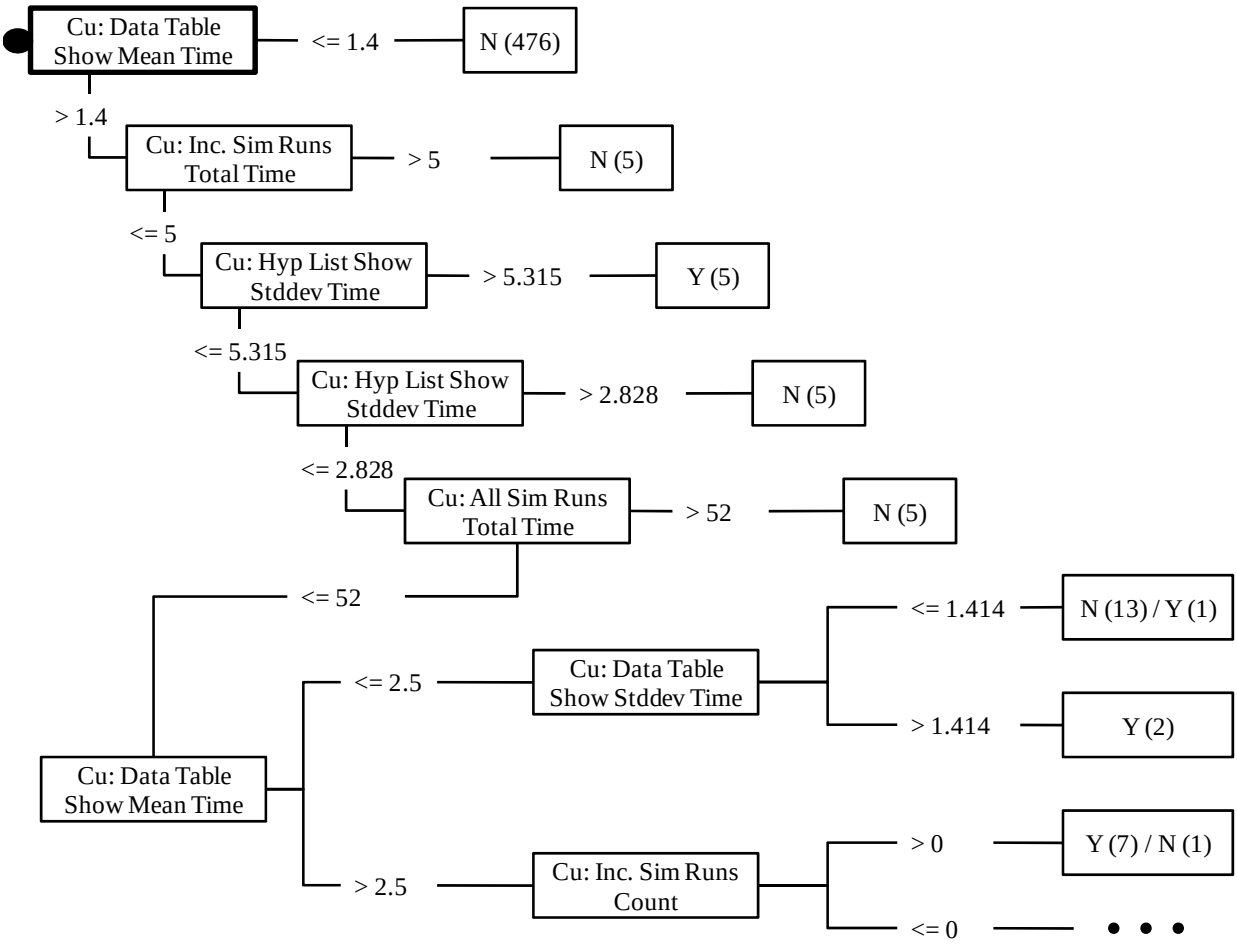

Figure 6. Portion of the decision tree for planning using the table tool.

5 Estimating Skill and Predicting Posttest Performance using Detectors

Given the ability to classify systematic behavior using our detectors, to a reasonable degree, it is possible to predict students' latent proficiency at each skill. Being able to predict proficiency has two key benefits. First, skill predictions can be used to determine when and how to adaptively scaffold students to support their learning. Second, by building these estimates, we can investigate whether authentic inquiry skill demonstrated within the microworld predicts performance on the transfer tests.

We constructed estimates of proficiency for the three systematic data collection behaviors for which detectors were successfully developed: designing controlled experiments, testing stated hypotheses, and planning using the table tool. To produce these estimates, we labeled all student data with the detectors, including data not labeled by human coders (approximately 62% of the data). We then aggregated students' performance across the four phase change activities. We compared two approaches for doing this in terms of their efficacy at predicting future performance and at predicting performance on the inquiry transfer tests (these tests are discussed in detail in Section 3.2.2). These two approaches for modeling student performance were an average-based approach, and Bayesian Knowledge-Tracing, a classic approach for modeling learning within intelligent tutoring systems (cf. Corbett & Anderson, 1995). In this section, we describe the procedure for constructing and evaluating these predictive models. We then present comparisons on how well each model predicted behavior within the phase change environment, and predicted performance on the transfer tests.

5.1 Generating Composite Measures of Systematic Student Behavior

Building skill proficiency estimates required a full collection of assessments for all students over all activities. Thus, we leveraged the behavior detectors to determine if each skill was exhibited within each clip. The resulting collection of assessments contained different numbers of clips per student, because students could engage in data collection varying numbers of times within each activity (see Figure 1). In some cases though, students transitioned between inquiry phases, causing new clips to be generated, but performed no actions within those phases. Clips of this nature were pruned from the collection of assessments since these events did not reflect students' skills or lack thereof. Each of the remaining clips was treated as a practice opportunity. These assessed practice opportunities served as the basis for

producing our two proficiency measures, estimates of how well the student knew each data collection skill. We describe how we computed these estimates below.

5.2 Average-Based Proficiency Estimate

The first proficiency estimate, an average-based approach, assumed no learning occurred between practice attempts. In this approach, we averaged the number of clips in the practice opportunity corpus that positively demonstrated each skill. For example, if a student engaged in 12 data collection activities (clips) and was labeled as designing controlled experiments in 5 out of 12 clips and testing their hypotheses in 8 out of 12 clips, they received .42 and .75, for each respective skill proficiency estimates. This approach serves as a reasonable baseline because it can be expected that students who possess the inquiry skills of interest will demonstrate them at each practice opportunity.

5.3 Bayesian Knowledge-Tracing Proficiency Estimate

The second proficiency estimate employed Bayesian Knowledge-Tracing (BKT) (Corbett & Anderson, 1995). BKT is a two-state Hidden Markov Model, mathematically equivalent to a simple Bayes Net (Reye, 2004), that tries to measure the latent trait of knowledge based on observable student performance over a series of time-slices. In our case, the observable student performance is the demonstration of systematic data collection skill.

The BKT model assumes that at any given opportunity to demonstrate a skill, a student either knows the skill or does not know the skill, and may either give a correct or incorrect response (i.e. demonstrate or fail to demonstrate the inquiry skill). Further, the model assumes that all skills are independent. A student who does not know a skill generally will give an incorrect response, or in our

case, fail to demonstrate a systematic inquiry skill. But, there is a certain probability (P(G), the Guess parameter) that the student will demonstrate appear to demonstrate the skill despite not knowing it. Correspondingly, a student who knows a skill generally will demonstrate it, but there is a certain probability (P(S), the Slip parameter) that they will not succeed in demonstrating the skill. At the beginning, each student has an initial probability P(L_0) of knowing each skill, and at each practice opportunity, the student has a certain probability P(T) of learning the skill irrespective of whether or not they demonstrated it. Thus, there are four parameters, P(G), P(S), P(L_0), and P(T), which must be learned to form a BKT model for a skill. Within the classic implementation of Bayesian Knowledge-Tracing, these four parameters are assumed to be the same for all students.

Using these four parameters, we can incrementally compute the probability that a student knows the skill and the estimate that a student will demonstrate that skill after completing a practice opportunity. The equations for these calculations are:

$$P(L_{n-1}|Clip_n) = \begin{cases} \dfrac{P(L_{n-1}) * \big(1 - P(S)\big)}{P(L_{n-1}) * \big(1 - P(S)\big) + \big(1 - P(L_{n-1})\big) * P(G)}, Behavior\_in\_Clip_n \\ \dfrac{P(L_{n-1}) * P(S)}{P(L_{n-1}) * P(S) + \big(1 - P(L_{n-1})\big) * (1 - P(G))}, \sim Behavior\_in\_Clip_n \end{cases}$$

$$P(L_n) = P(L_{n-1}|Clip_n) + \Big(\big(1 - P(L_{n-1}|Clip_n)\big) * P(T)\Big)$$

$$P(Behavior\_in\_Clip_n) = P(L_{n-1}) * \big(1 - P(S)\big) + (1 - P(L_{n-1})) * P(G).$$

In the equations above, we compute the estimate that the student knows the skill after a practice attempt at time *n*, P(L_n), by first applying Bayes' Theorem to calculate the probability that the student knew the skill beforehand using the evidence from the current clip, $P(L_{n-1}|Clip_n)$. Then, we take into account the possibility that the student learned it in during this practice opportunity, *P*(*T*). Note that in the P(L_n) equation, $P(L_{n-1}|Clip_n)$is replaced by one of the two equations depending on whether or the

behavior was demonstrated in clip *n*. In addition, we can also estimate the likelihood that a student will demonstrate the behavior $P(Behavior\_in\_Clip_n)$ at a practice opportunity *n,* based on the probability of previously knowing the skill, i.e. $P(L_{n-1})$, and how likely a guess or slip is for this skill.

To derive BKT models for each systematic data collection skill, we determined values for the four parameters ($P(L_0)$, P(T), P(G), P(S)) as follows. As in Corbett and Anderson (1995), the values of Guess (P(G)) and Slip (P(S)) were bounded in order to avoid the "model degeneracy" problems that arise when performance parameter estimates rise above 0.5 (cf. Baker, Corbett & Aleven, 2008). When values of these parameters go above 0.5, it is possible to get paradoxical behavior where, for instance, a student who knows a skill is more likely to get it wrong than to get it right. For our work, both P(G) and P(S) were bounded to be below 0.3. However, unlike Corbett and Anderson (1995), we used brute force search to find the best fitting estimates of the four parameters. In this approach, all potential parameter combinations of values at a grain-size of 0.01 were tried for each skill, i.e. *Params* = ($P(L_0)$, P(T), P(G), P(S)) $\in$ {(0.01, 0.01, 0.01, 0.01), (0.01, 0.01, 0.01, 0.02), (0.01, 0.01, 0.01, 0.03), …, (0.01, 0.01, 0.02, 0.01), …, (0.99, 0.99, 0.30, 0.30)}. We chose the parameter set yielding the lowest sum of squares residual (*SSR*) between the likelihood of showing a behavior $P(Behavior\_in\_Clip_n)$, and the actual data, $Observed\_Behavior\_in\_Clip_n$. In other words, we computed

$$SSR = (Observed\_Behavior\_in\_Clip_n - P(Behavior\_in\_Clip_n))^2$$

for all clips, summed these values, and chose the parameter set yielding the lowest sum. After finding the best parameter set *Params*, we repeated the above procedure to find a tighter fit for the parameters, *Params*', by searching around a radius of .01 for each parameter at a .001 grain-size. For example, if the parameters yielding the lowest SSR on the first pass were *Params* = (.05, .03, .04, .06), we would search for tighter fitting parameters by finding the parameters *Params*' yielding the minimal SSR within the set {(0.041, 0.021, 0.031, 0.051), …, (0.059, 0.039, 0.049, 0.069)}.

We chose the brute force approach since recent investigations (e.g. Ritter et al., 2009; Pardos & Heffernan, 2010) suggested that the Bayesian Knowledge-Tracing parameter space is non-convex (Boyd & Vandenberghe, 2004), meaning that using the Expectation Maximization (EM) algorithm could produce non-optimal parameters. Furthermore, the brute force approach we used has been shown to be computationally tractable for larger datasets than ours (cf. Baker, et al., 2010). However, we note that this approach may still not guarantee optimal parameters; there has been recent debate as to whether Expectation Maximization functions better or worse than Brute-Force Search and other algorithms. Results thus far have been inconclusive across studies (cf. Pavlik, Cen & Koedinger, 2009; Gong, Beck & Heffernan, 2010; Baker et al., 2010, 2011).

It is also worth noting that BKT has been shown to effectively estimate skill even when data on correctness of student responses are noisy (Beck & Sison, 2006). This work showed that BKT models are effective even when observables are biased towards false negatives, as in our case since the skill detectors bias towards labelling students as not demonstrating skill (see Section 4.7).

5.4 Results

5.4.1 Comparison of Models in Predicting Inquiry Behavior within the Microworld

We compared the two estimation models for each skill on how well they could predict if students would demonstrate that skill for a given practice attempt ($Clip_n$) within the phase change environment. To do this, each detector was first used to classify the entire data set, including data not labeled by the human coders. Then, each estimation model was applied to the entire data set to produce proficiency estimates at each practice opportunity. For the average-based model, we estimated skill at the time of $Clip_n$ by computing the average number of times the student demonstrated skills in prior attempts (Avg_{n-1}). For the BKT model, we computed $P(Behavior\_in\_clip_n)$, the probability that the student would demonstrate the skill in $Clip_n$, which was based on having previously learned the skill, and on the guess and slip

parameters. In these comparisons, we omitted all first practice opportunities from model comparison, as the average-based model is undefined in this context.

Model goodness was determined by computing A' because it is an appropriate metric when the predicted value is binary (skill demonstrated in $Clip_n$ or not), and the predictors for each model are numerical (probabilities of demonstrating skill). However, unlike Baker et al. (2010) which computed A' values per student and aggregated them into a meta A', we collapsed over the student parameter. This was done because there was not enough within variance per skill for each student to produce a meaningful within-student A'. We then compared models based on their A' goodness measures.

For each behavior, the average-based and BKT models fit comparably well to student performance. There was at most a 2% difference between the two models for each skill: designing controlled experiments (A'=.74 for both), planning using the table tool (BKT model A'=.70 vs. Avg model A'=.71), and testing hypotheses (BKT model A'=.79 vs. Avg model A'=.77). Typically, the difference between models can be statistically tested by performing a Z-test comparing A' values (cf. Fogarty, Baker, & Hudson, 2005). However, we could not perform this analysis since we collapsed over students to compute A' values, making all observations non-independent. In practice, the differences between the proficiency estimate models for each skill are small. Additionally, the A' values are reasonably high in all cases, suggesting that either approach would be adequate in estimating proficiency.

Thus, both models perform acceptably at predicting the success of a student's current practice attempt within the phase change environment. using aggregated information from previous practice attempts. Furthermore, these findings suggest that the behaviors represent a latent skill which is stable over time. Surprisingly, though, the BKT model did not outperform the average-based model. We discuss possible reasons why the BKT model did no better than the average-based model, in the Discussion and Conclusions section.

5.4.2 Comparison of Models in Predicting Transfer Test Performance

The proficiency estimate models also enabled us to examine if inquiry skills demonstrated in the phase change environment predicted performance on the transfer tests. This also provides a measure of external validity of these models. As described in Section 3.2.2, students completed three transfer tests: a multiple-choice test of inquiry on designing controlled experiments, a hands-on test of designing controlled experiments to determine what factors make a ball roll further down a ramp, and a multiple-choice test on testing hypotheses. In our analyses, we determined the degree to which a particular skill demonstrated within the phase change environment predicted performance on the transfer test measuring the same skill. For example, performance on the designing controlled experiments transfer tests was predicted using the two proficiency models' final estimates of that skill in the phase change environment. We computed these final skill estimates as follows. The final skill estimate for the average-based models was computed as the average number of times the student exhibited the behavior over all clips. For the BKT models, we used the estimate of student knowledge, $P(L_n)$, at the last clip. We used $P(L_n)$ rather than *P(Behavior_in_Clip)* for the BKT estimate because the guess and slip parameters used within the microworld may not apply to our transfer tests.

In these analyses, we considered only students who completed both the phase change microworld activities and the inquiry and ramp transfer post-tests. This resulted in 134 students. Prediction strength was determined by computing Pearson correlations (r), and checking if they were significantly different from zero (no positive or negative correlation) at the $p < .05$ level. Means and standard deviations for each test and each model's final skill estimates are shown in Table 5.

Both the BKT model and average-based model significantly predicted transfer test performance, with the average-based model predicting performance equally well or better than the BKT model for all tests, as shown in Table 6. For the designing controlled experiments skill, both models achieved relatively modest but statistically significant correlations to each of the transfer tests. The two models

each achieved a correlation of r=.26, p=.003 to the multiple-choice test. They also achieved similar correlations with the hands-on activity, the ramp environment, r=.38, p<.001 for the average-based model, and r=.37, p<.001 for the BKT model.

For the skill of testing stated hypotheses, the average-based model achieved better predictive performance on this skill's corresponding multiple-choice transfer test (r=.41, p<.001) than the corresponding BKT model (r=.31, p<.001). The difference between correlations was statistically significant, as determined by a significance test of the difference between two correlation coefficients with correlated samples (Ferguson, 1976, pp.171-172), t(131)=2.50, p=.01. This suggests that the average-based model yielded a more valid estimate of this skill.

Finally, we examined if planning using the table tool significantly predicted performance on any of the transfer tests, which were designed to test the other two skills. This measure was not correlated to a statistically significant degree to either of the designing controlled experiments transfer tests. However, the average-based model achieved a significant if modest correlation to the multiple-choice test for testing stated hypotheses (r=.24, p<.01). Curiously, there was not a statistically significant correlation for the BKT model (r=.05, p=.57). The difference between the two models was marginally significant, t(131)=1.85, p=.07. Because the models did not agree, our results are inconclusive as to whether or not planning skill relates to proficiency at hypothesis testing as measured by our multiple-choice test.

Table 5. Means and Standard Deviations for estimates of inquiry skill and posttest measures, $N = 134$.

	Max	*M*	*SD*
Average-Based Models			
Controlled Experiments Average		0.25	0.26
Testing Hypotheses Average		0.30	0.29
Mindful Planning using Table Tool Average		0.05	0.11
BKT Models			
Controlled Experiments BKT Estimate		0.32	0.38
Testing Hypotheses BKT Estimate		0.43	0.41
Mindful Planning using Table Tool BKT Estimate		0.01	0.08
Dependent Measures			
Inquiry Posttest: Testing Hypotheses	6	2.06	1.53
Inquiry Posttest: Controlled Experiments	4	2.09	1.18
Ramp Transfer: Controlled Experiments	4	1.59	1.70

Table 6. Correlations between posttest measures and each model's estimate of inquiry skill, $N = 134$.

	Avg-Based Model *r*	BKT Model *r*	*t* Difference
Dependent Measures			
Inquiry Posttest: Testing Hypotheses	.41***	.31***	2.50*
Inquiry Posttest: Controlled Experiments	.26**	.26**	0.03
Ramp Transfer: Controlled Experiments	.38***	.37***	0.21

Note: The *t* difference between model correlations was computed using a significance of the difference between two correlation coefficients for correlated samples (Ferguson, 1976, pp.171-172).

*$p < .05$; **$p < .01$; *** $p < .001$

5.4.3 Comparing the Hands-on and Multiple Choice Transfer Assessments

Given that the models significantly predicted performance on both the multiple choice and authentic transfer measures of designing controlled experiments, we examined whether the models were better predictors for one type of test format than the other. This enabled us to address whether a performance assessment better captured inquiry skill than a multiple choice test, a topic of debate within the assessment and science education communities (cf. Black, 1999; Pellegrino, 2001). Recall that both models appeared to have stronger correlations with the ramp transfer test than the multiple-choice measure of designing controlled experiments (Avg-based r=.26 for the multiple-choice measure vs. r=.38 for the authentic measure, and BKT r=.26 for the multiple-choice measure vs. r=.37 for the authentic measure). Though the correlation appeared to be stronger for both models for the ramp task, neither difference was statistically significant, $t(131)=1.38$, p=.17 for the average-based model and $t(131)=1.30$, p=.20 for the BKT model.

6 Discussion and Conclusions

Despite the recognized importance of learning scientific inquiry skills (National Research Council, 1996; Kuhn, 2005), hands-on inquiry activities are seldom used in schools because assessing authentic inquiry cannot be done reliably or easily (cf. Alonzo & Aschbacher, 2004; Gotwals & Songer, 2006). This difficulty arises in part because students may follow a large number of productive and unproductive paths as they engage in inquiry (cf. de Jong, 2006; Buckley, Gobert, Horwitz & O'Dwyer, 2010). Towards facilitating assessment of these skills, we have presented machine learning-based detectors of four kinds of systematic data collection behavior using data from a physical science-based microworld. These behaviors were: testing stated hypotheses, designing controlled experiments, and using a hypothesis list and data table to plan their data collection. The detectors were generated using a novel approach that

combined text replay tagging, an extension to the text replay approach described in Baker, Corbett and Wagner (2006), and machine learning. Development of these detectors not only enables real-time assessment of inquiry behaviors, but also permits the construction of models to estimate inquiry skill proficiency over time. To this end, we compared two models for estimating skill at a given time, an average-based approach and Bayesian Knowledge-Tracing (Corbett & Anderson, 1995), on both their ability to predict performance within our phase change environment and on their efficacy at predicting performance on transfer assessments requiring similar skills.

Overall, our results for classifying student behavior with the detectors were very promising. We can distinguish a set of trials in which a student designed controlled experiments from a set of trials in which students did not design controlled experiments 85% of the time. We can also distinguish a set of trials in which a student tested their stated hypotheses from a set of trials in which they did not 85% of the time. Finally, we can distinguish a set of trials in which the table tool was used to plan experiments to run next from when it is not 94% of the time. Furthermore, the associated Kappa values, ranging from .40 to .47, indicate that each of these detectors is better than chance. The final detector for planning using the hypothesis viewer was less successful, but may still be acceptable for fail-soft interventions in which students can receive interventions that are not costly if misapplied.

Given this level of cross-validated performance, the detectors of designing controlled experiments, testing stated hypotheses, and planning using the table tool appear to be sufficiently accurate to be used to select students for scaffolding. However, it is important to note that the detectors bias towards false negatives, as evidenced by the fact that clips tagged as "Y" were only correctly identified between 51% and 63% of the time under cross-validation for our three best detectors. As such, any scaffolds administered based on the detectors' classification should be fail-soft. For example, if the system detects that a student is not designing controlled experiments, the student can be provide a small reminder about how to correctly design controlled experiments. Another option

is to highlight a portion of the table containing data the student already collected and provide high-level hints on how to design a single pairwise controlled experiment, using their previous experiment as a reference.

Though our detectors achieved good performance, there are other considerations that may lead to better prediction in future work. In our analyses, we compared and constructed behavior detectors using only J48 decision trees from two kinds of feature sets. J48 was chosen based on previous successes in using this algorithm to generate behavior detectors using text replays (Baker & de Carvalho, 2008; Baker, Mitrovic & Mathews, 2010). However, other data mining algorithms may have produced better detectors, and thus building detectors using other machine learning algorithms will be a valuable area of future work. Also, the goodness of our detectors is contingent on a specific set of distilled features; extending the set of features to include other types of features has the potential to improve prediction. For example, Baker, Corbett, Roll, & Koedinger (2008) analyzed student timing and counts in relation to averages and standard deviations *across students*, whereas we only computed features *within* students. Our feature set also did not capture any temporal relationships between student actions, meaning that the order in which students performed actions was not considered. To take this into account, we could either attempt to engineer new features which do so, or use a process mining approach (cf. Köck & Paramythis, 2011; van der Aalst, 2011) or machine learning-based plan generation algorithm (Gu, Wu, Tao, Pung, & Lu, 2009).

Another important area for future research will be to determine the degree to which these detectors can predict systematic behavior for a different student sample and within other physical science microworlds (e.g. Gobert, Raziuddin & Sao Pedro, 2011). Currently, we have no reason to believe our detectors are over-fit to particular kinds of students since each student was represented equally within our training corpus, and since cross-validation was conducted at a student level. However, we only used students from one school, and it is possible that our detectors are over-fit to some aspect of

learning and performance at this school. Training new detectors with students from additional schools may improve generalizability. Similarly, it is also possible that our models for these systematic behaviors are over-fit to the particular physical science domain, phase change, from which student data were obtained. Again, incorporating training data from other physical science microworlds may improve generalizability.

We have also discussed potential advantages stemming from using a machine learning-based approach to detecting inquiry skill, relative to previous approaches that used knowledge engineering to model inquiry behaviors and skills (e.g. Koedinger, Suthers & Forbus, 1998; Schunn & Anderson, 1998; Buckley, Gobert, & Horwitz, 2006; Buckey et al., 2010; McElhaney & Linn, 2010). Arguably, the most important advantage of machine-learned models is that it is easier to validate their goodness, since test labels are available (they are necessary for classification), and cross-validation is possible. This validation is important since it provides researchers with a measurable level of assurance that each behavioral construct is properly represented in the models. In terms of practical significance, however, it is still an open question whether our machine-learned models better represent our inquiry behaviors than their knowledge-engineered counterparts. One valuable future step will be to compare knowledge-engineered approaches to our machine-learned models in terms of which better predicts a new set of text-replay tagged data.

Since three of the four detectors achieved acceptable classification of inquiry behaviors, comparable to other detectors of other behaviors used in effective interventions (e.g. Baker, Corbett, Roll, & Koedinger, 2008; Baker & de Carvalho, 2008), we leveraged them to classify all students' inquiry behaviors, including unlabeled data. This permitted us to generate and compare two estimates of students' authentic inquiry skill proficiency. We compared two approaches for generating these estimates, an average-based method that assumes no learning, and Bayesian Knowledge-Tracing (BKT) (Corbett & Anderson, 1995), in terms of their ability to accurately estimate student skill at each practice

opportunity. Both approaches estimated students' skills at each practice opportunity acceptably, as indicated by A' values in the .70-.80 range. Hence, either of these models can be used to assign scaffolds within the microworld. For example, if the system detects that a student has low skill for designing controlled experiments and testing hypotheses, the system could let them specify only one hypothesis and then provide the student with proactive scaffolding on designing effective experimental contrasts. Alternately, if the system detects that a student has high knowledge of these skills, the system could let them explore more freely, specifying as many hypotheses to test as they wish (the current design of the environment), and provide hints only on-demand. The system's estimates of proficiency can also be used to provide formative assessment feedback to teachers on inquiry proficiency for individual students or entire classes.

Within this paper, the models of skill proficiency were also used to study the relationship between skill demonstrated in the phase change environment, and performance on the multiple-choice and authentic transfer measures of inquiry skill. Overall, each model of authentic inquiry skill was significantly, albeit modestly, correlated to its corresponding posttest, i.e. the testing hypothesis behavior correlated with the standardized-test style questions on hypotheses. This provides some external validation of the skill estimates derived from performance within the phase change environment. These transfer findings also support the notion that authentic inquiry skills are not necessarily tied to the domain in which the skills were learned.

Surprisingly, the predictive strength of the average-based estimates of skill was as good as the BKT estimates, and in the case of the multiple-choice test for testing hypotheses, the average-based model was significantly better than the BKT estimate. This finding may have several potential interpretations. First, the data set used here was quite small. It may be that more student data, both in terms of number of students and practice attempts, is required to obtain BKT models, due to the greater number of parameters. Another possibility is that we found relatively little learning occurring in our

phase change environment, and Bayesian Knowledge-Tracing assumes that students' skills improve with each practice opportunity, on the whole. This is evidenced by the low learning rate for the best-fitting BKT models (the learning parameter *T* equaled 0.038 and 0.058 for designing controlled experiments and testing hypotheses, respectively). These findings, though, are not surprising since our environment provided no explicit learning support, and it has been found that students need more time and repeated practice to develop data collection skills in the absence feedback (Dean Jr. & Kuhn, 2007). Thus, BKT may be more appropriate for skill assessment in environments that provide explicit learning support, the context where it is typically used (e.g. Corbett & Anderson, 1995; Koedinger & Corbett, 2006; Baker, Corbett, & Aleven, 2008; Ritter et al., 2009; Feng, Heffernan & Koedinger, 2009; Baker, Corbett, Gowda, et al., 2010; Pardos, Heffernan, Anderson, & Heffernan, 2010). BKT may become more effective for our science inquiry microworlds once we have added explicit scaffolding based on the assessments of student inquiry described in this paper. However, it may also be that another model is more appropriate than both BKT and the average-based approach. More empirical evidence is needed to test this.

Finally, we leveraged the proficiency models to determine if authentic skill at designing controlled experiments in the phase change microworld was assessed more accurately with a hands-on performance assessment than with multiple choice questions. This issue is important to the assessment and science education communities since it is unclear whether multiple-choice tests can adequately measure inquiry (Black, 1999; Pellegrino, 2001). We found no significant differences between the two sets of correlations, namely the correlations between each authentic skill estimate and the hands-on assessment, and the correlations between each estimate and the multiple choice assessment. Therefore, we could not determine whether one transfer test better measured latent inquiry skill than the other. The similarity between correlations may be because the ramp and multiple choice assessments are both domain-neutral and isolate the assessment of a single skill. The phase change microworld, on the other hand, is complex and domain-rich, meaning that domain knowledge may

influence inquiry performance (Glaser, Schauble, Raghavan & Zeitz, 1992; Schauble, Klopfer & Raghavan, 1991). A further study which uses two domain-rich measures of inquiry skill (e.g. state change and another physical science domain) could help disentangle this issue. We do note, though, our results are in accordance with earlier findings that multiple choice measures are poor assessments of inquiry performance (Black, 1999; Pellegrino, 2001), since, as previously mentioned, the correlation between authentic skill and the multiple choice test was modest.

In summary, the work presented here shows that machine learning/educational data mining methods can be employed, in combination with human classifications, to produce verifiable models of low-level inquiry behavior within an inquiry environment. We then aggregated the outputs of these behavior detectors to form estimates of skill proficiency. These estimates enabled us to study the relationship between authentic inquiry skill and performance on multiple-choice test-based measures of inquiry. Going forward, these detectors can be used as a basis for formulating estimates of skill to support scaffolding. At this point, an interesting question for future research will be to determine how the relationships between authentic inquiry skill and performance on multiple-choice tests change once scaffolding has been incorporated. Scaffolding may enable students to perform well, despite incomplete knowledge, and may enable them to acquire these skills while as they engage in inquiry. Given our findings, we anticipate that these models will also enable future "discovery with models" analyses (cf. Baker & Yacef, 2009) that can shed light on the relationship between a student's mastery of systematic experimentation strategies and the student's domain learning (Gobert, et al., 2007; Gobert, et al., 2009). However, additional research will be needed to determine if these findings are robust over different student populations and if the feature set and associated detectors are general enough (cf. Ghazarian & Noorhosseini, 2010) to be applicable to microworlds in other scientific domains. In general, research on how effectively these detectors generalize to microworlds in other scientific domains will increase the

models' potential for broad usefulness, both to support student learning, and to support research on assessment of these types of ill-defined science inquiry skills.

Acknowledgements

This research is funded by the National Science Foundation (NSF-DRL#0733286, NSF-DRL#1008649, and NSF-DGE#0742503) and the U.S. Department of Education (R305A090170). Any opinions expressed are those of the authors and do not necessarily reflect those of the funding agencies.

References

Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2006). Toward Meta-Cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education, 16*(2), 101-130.

Alonzo, A., & Aschbacher, P. (2004, April 15). Value Added? Long assessment of students' scientific inquiry skills. *Paper presented at the annual meeting of the American Educational Research Association.* San Diego, CA: Retrieved December 20, 2010, from the AERA Online Paper Repository.

Amershi, S., & Conati, C. (2009). Combining Unsupervised and Supervised Machine Learning to Build User Models for Exploratory Learning Environments. *Journal of Educational Data Mining, 1*(1), 71-81.

Azevedo, R. (2007). Understanding the complex nature of self-regulatory processes in learning with computer-based learning environments: an introduction. *Metacognition and Learning, 2*(2-3), 57-65.

Azevedo, R. (2009). Theoretical, conceptual, methodological, and instructional issues in research on metacogntion and self-regulated learning: A discussion. *Metacognition and Learning, 4*(1), 87-95.

Baker, R. (2007). Is Gaming the System State-or-Trait? Educational Data Mining Through the Multi-Contextual Application of a Validated Behavioral Model. *Complete On-Line Proceedings of the Workshop on Data Mining for User Modeling, at the 11th International Conference on User Modeling (UM 2007)*, (pp. 76-80). Corfu, Greece.

Baker, R. S., Corbett, A. T., Roll, I., & Koedinger, K. R. (2008). Developing a Generalizable Detector of When Students Game the System. *User Modeling and User-Adapted Interaction, 18*(3), 287-314.

Baker, R. S., Mitrovic, A., & Mathews, M. (2010). Detecting Gaming the System in Constraint-Based Tutors. In P. De Bra, P. Kobsa, & D. Chin (Ed.), *Proceedings of the 18th Annual Conference on User Modeling, Adaptation and Personalization, UMAP 2010. LNCS 6075*, pp. 267-278. Big Island of Hawaii, HI: Springer-Verlag.

Baker, R., & de Carvalho, A. (2008). Labeling Student Behavior Faster and More Precisely with Text Replays. In R. S. Baker, T. Barnes, & J. E. Beck (Ed.), *Proceedings of the 1st International Conference on Educational Data Mining, EDM 2008*, (pp. 38-47). Montreal, Quebec, Canada.

Baker, R., & Yacef, K. (2009). The State of Educational Data Mining in 2009: A Review and Future Visions. *Journal of Educational Data Mining, 1*(1), 3-17.

Baker, R., Corbett, A., & Aleven, V. (2008). Improving Contextual Models of Guessing and Slipping with a Truncated Training Set. In R. S. Baker, T. Barnes, & J. E. Beck (Ed.), *Proceedings of the 1st International Conference on Educational Data Mining, EDM 2008*, (pp. 67-76). Montreal, Quebec, Canada.

Baker, R., Corbett, A., & Aleven, V. (2008). More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge-Tracing. In B. Woolf, E. Aimeur, R. Nkambou, & S. Lajoie (Ed.), *Proceedings of the 9th International Conference on Intelligent Tutoring Systems, ITS 2008. LNCS 5091*, pp. 406-415. Montreal, Quebec, Canada: Springer-Verlag.

Baker, R., Corbett, A., & Wagner, A. (2006). Human Classification of Low-Fidelity Replays of Student Actions. *Proceedings of the Educational Data Mining Workshop held at the 8th International Conference on Intelligent Tutoring Systems, ITS 2006*, (pp. 29-36). Jhongli, Taiwan.

Baker, R., Corbett, A., Gowda, S., Wagner, A., MacLaren, B., Kauffman, L., et al. (2010). Contextual Slip and Prediction of Student Performance After Use of an Intelligent Tutor. In P. De Bra, P. Kobsa, & D. Chin (Ed.), *Proceedings of the 18th Annual Conference on User Modeling, Adaptation and Personalization, UMAP 2010. LNCS 6075*, pp. 52-63. Big Island of Hawaii, HI: Springer-Verlag.

Baker, R., Corbett, A., Koedinger, K., Evenson, E., Roll, I., Wagner, A., et al. (2006). Adapting to When Students Game an Intelligent Tutoring System. In M. Ikeda, K. Ashlay, & T.-W. Chan (Ed.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems, ITS 2006. LNCS 4053*, pp. 392-401. Jhongli, Taiwan: Springer-Verlag.

Baker, R., Pardos, Z., Gowda, S., Nooraei, B., & Heffernan, N. (2011). Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems. In J. Konstan, R. Conejo, J. Marzo, & N. Oliver (Ed.), *Proceedings of the 19th International Conference on User*

Modeling, Adaptation and Personalization, UMAP 2011. LNCS 6787, pp. 13-24. Girona, Spain: Springer.

Beck, J. (2005). Engagement Tracing: Using Response Times to Model Student Disengagement. In C. K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Ed.), *Proceedings of the 12th International Conference on Artificial Intelligence in Education, AIED 2005* (pp. 88-95). Amsterdam, Netherlands: IOS Press.

Beck, J., & Chang, K. (2007). Identifiability: A Fundamental Problem of Student Modeling. In C. Conati, K. Mccoy, & G. Paliouras (Ed.), *Proceedings of the 11th International Conference on User Modeling, UM 2007. LNCS 4511*, pp. 137-146. Corfu, Greece: Springer-Verlag.

Beck, J., & Sison, J. (2006). Using Knowledge Tracing in a Noisy Environment to Measure Student Reading Proficiencies. *International Journal of Artificial Intelligence in Education, 16*(2), 129-143.

Ben-David, A. (2008). About the Relationship between ROC Curves and Cohen's Kappa. *Engineering Applications of Artificial Intelligence, 21*, 874-882.

Bernardini, A., & Conati, C. (2010). Discovering and Recognizing Student Interaction Patterns in Exploratory Learning Environments. In V. Aleven, J. Kay, & J. Mostow (Ed.), *Proceedings of the 10th International Conference of Intelligent Tutoring Systems, ITS 2010, Part 1* (pp. 125-134). Pittsburgh, PA: Springer-Verlag.

Black, P. (1999). *Testing: Friend or Foe? Theory and Practice of Assessment and Testing.* New York, NY: Falmer Press.

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization.* Cambridge, UK: Cambridge University Press.

Buckley, B. C., Gobert, J. D., & Horwitz, P. (2006). Using log files to track students' model-based inquiry. In S. Barab, K. Hay, & D. Hickey (Ed.), *Proceedings of the 7th International Conference on Learning Sciences, ICLS 2006* (pp. 57-63). Bloomington, Indiana: Lawrence Erlbuam Associates.

Buckley, B., Gobert, J., Horwitz, P., & O'Dwyer, L. (2010). Looking Inside the Black Box: Assessments and Decision-making in BioLogica. *International Journal of Learning Technology, 5*(2), 166-190.

Cetintas, S., Si, L., Xin, Y., & Hord, C. (2010). Automatic Detection of Off-Task Behaviors in Intelligent Tutoring Systems with Machine Learning Techniques. *IEEE Transactions on Learning Technologies, 3*(3), 228-236.

Chen, Z., & Klahr, D. (1999). All Other Things Being Equal: Acquisition and Transfer of the Control of Variables Strategy. *Child Development, 70*(5), 1098-1120.

Cocea, M., & Weibelzahl, S. (2009). Log File Analysis for Disengagement Detection in e-Learning Environments. *User Modeling and User-Adapted Interaction, 19*, 341-385.

Corbett, A., & Anderson, J. (1995). Knowledge-Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction, 4*, 253-278.

de Jong, T. (2006). Computer Simulations - Technological advances in inquiry learning. *Science, 312*(5773), 532-533.

de Jong, T., Beishuizenm, J., Hulshof, C., Prins, F., van Rijn, H., van Someren, M., et al. (2005). Determinants of Discovery Learning in a Complex Simulation Learning Environment. In P. Gardenfors, & P. Johansson, *Cognition, Education and Communication Technology* (pp. 257-283). Mahwah, NJ: Lawrence Erlbaum Associates.

Dean Jr., D., & Kuhn, D. (2006). Direct Instruction vs. Discovery: The Long View. *Science Education*, 384-397.

Dignath, C., & Buttner, G. (2008). Components of fostering self-regulated learning among students: A meta-analysis on intervention studies at primary and secondary school level. *Metacognition and Learning, 3*(3), 231-264.

Dragon, T., Woolf, B., Marshall, D., & Murray, T. (2006). Coaching Within a Domain Independent Inquiry Environment. In M. Ikeda, K. D. Ashley, & C. Tak-Wai (Ed.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems, ITS 2006. LNCS 4053*, pp. 144-153. Jhongli, Taiwan: Springer-Verlag.

Efron, B., & Gong, G. (1983). A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation. *The American Statistician, 37*(1), 36-48.

Feng, M., Heffernan, N., & Koedinger, K. (2009). Addressing the Assessment Challenge in an Intelligent Tutoring System that Tutors as it Assesses. *User Modeling and User-Adapted Interaction, 19*, 243-266.

Ferguson, G. (1976). *Statistical Analyis in Psychology and Education, 4th Edition.* New York, NY: McGraw-Hill Inc.

Fogarty, J., Baker, R., & Hudson, S. (2005). Case Studies in the Use of ROC Curve Analysis for Sensor-Based Estimates in Human Computer Interaction. *Proceedings of Graphics Inerface, GI 2005* (pp. 129-136). Victoria, British Columbia: Canadian Human-Computer Communications Society.

Gertner, A., & VanLehn, K. (2000). Andes: A Coached Problem Solving Environment for Physics. In G. Gauthier, C. Frasson, & K. VanLehn (Ed.), *Proceedings of the 5th International Conference on Intelligent Tutoring Systems, ITS 2000. LNCS 1839*, pp. 133-142. Montreal, Quebec, Canada: Springer-Verlag.

Ghazarian, A., & Noorhosseini, S. M. (2010). Automatic Detection of Users' Skill Levels Using High-Frequency User Interface Events. *User Modeling and User-Adapted Interaction, 20*(2), 109-146.

Glaser, R., Schauble, L., Raghavan, K., & Zeitz, C. (1992). Scientific Reasoning Across Different Domains. In E. DeCorte, M. Linn, H. Mandl, & L. Verschaffel, *Computer-based Learning Environments and Problem-Solving* (pp. 345-371). Heidelberg, Germany: Springer-Verlag.

Gobert, J. (2005). Leveraging Technology and Cognitive Theory on Visualization to Promote Students' Science Learning and Literacy. In J. Gilbert, *Visualization in Science Education* (pp. 73-90). Dordrecht, The Netherlands: Springer-Verlag.

Gobert, J., Buckley, B., Levy, S., & Wilensky, U. (2007, April 12). Teasing Apart Domain-Specific and Domain-General Inquiry Skills: Co-evolution, Bootstrapping, or Separate Paths? *Paper presented at the annual meeting of the American Educational Research Association.* Chicago, IL: Retrieved April 18, 2011, from the AERA Online Paper Repository.

Gobert, J., Raziuddin, J., & Sao Pedro, M. (2011). The Influence of Learner Characteristics on Conducting Scientific Inquiry Within Microworlds. In L. Carlson, C. Hoelscher, & T. Shipley (Ed.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (pp. 372-377). Boston, MA: Cognitive Science Society.

Gobert, J.; Heffernan, N.; Koedinger, K.; Beck, J. (2009). ASSISTments Meets Science Learning (AMSL). Proposal (R305A090170) funded by the U.S. Dept. of Education.

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

Gong, Y., Beck, J., & Heffernan, N. (2010). Using Multiple Dirichlet Distributions to Improve Parameter Plausibility. In R. S. Baker, A. Merceron, & P. I. Pavlik Jr. (Ed.), *Proceedings of the 3rd International Conference on Educational Data Mining, EDM 2010*, (pp. 61-70). Pittsburgh, PA.

Gotwals, A., & Songer, N. (2006). Measuring Students' Scientific Content and Inquiry Reasoning. In S. Barab, K. Hay, & D. Hickey (Ed.), *Proceedings of the 7th International Conference of the Learning Sciences, ICLS 2006* (pp. 196-202). Bloomington, IN: Lawrence Erlbuam Associates.

Graesser, A., Chipman, P., Haynes, B., & Olney, A. (2005). AutoTutor: An Intelligent Tutoring System with Mixed-Initiative Dialogue. *IEEE Transactions on Education, 48*(4), 612-618.

Gu, T., Wu, Z., Tao, X., Pung, H. K., & Lu, J. (2009). epSICAR: An Emerging Patterns based approach to sequential, interleaved and Concurrent Activity Recognition. *Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications, PERCOM '09* (pp. 1-9). Galveston, TX: IEEE Computer Society.

Hadwin, A., Nesbit, J., Jamieson-Noel, D., Code, J., & Winne, P. (2007). Examining Trace Data to Explore Self-Regulated Learning. *Metacognition and Learning, 2*(2-3), 107-124.

Hanley, J., & McNeil, B. (1982). The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology, 143,* 29-36.

Klahr, D., & Dunbar, K. (1988). Dual search space during scientific reasoning. *Cognitive Science, 12*(1), 1-48.

Köck, M., & Paramythis, A. (2011). Activity Sequence Modelling and Dynamic Clustering for Personalized E-Learning. *User Modeling and User-Adapted Interaction, 21,* 51-97.

Koedinger, K., & Corbett, A. (2006). Cognitive Tutors: Technology Bringing Learning Sciences to the Classroom. In R. Sawyer, *The Cambridge Handbook of the Learning Sciences* (pp. 61-77). New York, NY: Cambridge University Press.

Koedinger, K., Anderson, J., Hadley, W., & Mark, M. (1997). Intelligent Tutoring Goes to School in the Big City. *International Journal of Arificial Intelligence in Education, 8,* 30-43.

Koedinger, K., Suthers, D., & Forbus, K. (1998). Component-Based Construction of a Science Learning Space. *International Journal of Artificial Intelligence in Education (IJAIED), 10,* 292-313.

Kuhn, D. (2005). *Education for thinking.* Cambridge, MA: Harvard University Press.

Levy, S., & Wilensky, U. (2006, April 11). Emerging Knowledge through an Emergent Perspective: High-school Students' Inquiry, Exploration and Learning in the Connected Chemistry Curriculum. *Presented at the annual meeting of the American Educational Research Association.* San Francisco, CA: Retrieved April 18, 2011 from the AERA Online Paper Repository.

Manlove, S., & Lazonder, A. (2004). Self-regulation and Collaboration in a Discovery Learning Environment. *Paper presented at the first EARLI Metacognition SIG conference.* Amsterdam: Available: http://users.edte.utwente.nl/lazonder/homepage/NL/Publijst.html.

Manlove, S., Lazonder, A., & Dejong, T. (2007). Software Scaffolds to Promote Regulation During Scientific Inquiry Learning. *Metacognition and Learning, 2,* pp. 141-155.

Massachusetts Department of Education. (2006). *Massachusetts Science and Technology/Engineering Curriculum Framework.* Malden, MA: Massachusetts Department of Education.

McElhaney, K., & Linn, M. (2010). Helping Students Make Controlled Experiments More Informative. In K. Gomez, L. Lyons, & J. Radinsky (Ed.), *Learning in the Disciplines: Proceedings of the 9th International Conference of the Learning Sciences (ICLS 2010) - Volume 1, Full Papers* (pp. 786-793). Chicago, IL: International Society of the Learning Sciences.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). YALE: Rapid Prototyping for Complex Data Mining Tasks. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2006* (pp. 935-940). Philadelphia, PA: ACM Press.

Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education, 13*(2-4), 173-197.

Mitrovic, A., Mayo, M., Suraweera, P., & Martin, B. (2001). Constraint-Based Tutors: A Success Story. In L. Monostori, J. Vancza, & M. Ali (Ed.), *Proceedings of the 14th International Conference on Industrial and Engineering Application of Artificial Intelligence and Expert Systems: Engineering of Intelligent Systems, IEA/AIE-2001. LNCS 2070*, pp. 931-940. Budapest, Hungary: Springer-Verlag.

National Research Council. (1996). *National Science Education Standards.* National Science Education Standards. Washington, D.C.: National Academy Press.

Papert, S. (1980). Computer-based Microworlds as Incubators for Powerful Ideas. In R. Taylor, *The Computer in the School: Tutor, Tool, Tutee* (pp. 203-201). New York, NY: Teacher's College Press.

Pardos, Z., & Heffernan, N. (2010). Navigating the Parameter Space of Bayesian Knowledge-Tracing Models: Visualizations of the Convergence of the Expectation Maximization Algorithm. In R. S. Baker, A. Merceron, & P. I. Pavlik Jr. (Ed.), *Proceedings of the 3rd International Conference on Educational Data Mining*, (pp. 161-170). Pittsburgh, PA.

Pardos, Z., Heffernan, N., Anderson, B., & Heffernan, C. (2010). Using Fine-Grained Skill Models to Fit Student Performance with Bayesian Networks. In C. Romero, S. Ventura, S. R. Viola, M. Pechenizkiy, & R. S. Baker, *Handbook of Educational Data Mining* (pp. 417-426). Boca Raton, FL: CRC Press.

Pavlik, P., Cen, H., & Koedinger, J. (2009). Performance Factors Analysis - A New Alternative to Knowledge Tracing. In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. Graesser (Ed.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education, AIED 2009* (pp. 531-540). Brighton, UK: IOS Press.

Pea, R., & Kurland, D. (1984). On the Cognitive Effects of Learning Computer Programming. *New Ideas in Psychology, 2*, 137-168.

Pellegrino, J. (2001). *Rethinking and redesigning educational assessment: Preschool through postsecondary.* Education Commission of the States, US Department of Education, Denver, CO.

Ramsey, F., & Schafer, D. (1996). *The Statistical Sleuth: A Course in Methods of Data Analysis.* Florence, KY: Wadsworth Publishing Company.

Resnick, M. (1997). *Turtles, Termintes, and Traffic Jams: Explorations in Massively Parallel Microworlds.* Cambridge, MA: MIT Press.

Reye, J. (2004). Student Modeling Based on Belief Networks. *International Journal of Artificial Intelligence in Education, 14*(1), 1-33.

Ritter, S., Harris, T., Nixon, T., Dickinson, D., Murray, R., & Towle, B. (2009). Reducing the Knowledge-Tracing Space. In T. Barnes, M. Desmarais, C. Romero, & S. Ventura (Ed.), *Proceedings of the 2nd International Conference on Educational Data Mining, EDM 2009*, (pp. 151-160). Cordoba, Spain.

Roll, I., Aleven, V., & Koedinger, K. (2010). The Invention Lab: Using a Hybrid of Model Tracing and Constraint-Based Modeling to Offer Intelligent Support in Inquiry Environments. In V. Aleven, J. Kay, & J. Mostow (Ed.), *Proceedings of the 10th International Conference on Intelligent Tutoring Systems, ITS 2010* (pp. 115-124). Pittsburgh, PA: Springer-Verlag.

Roll, I., Aleven, V., McLaren, B., & Koedinger, K. (2007). Designing for metacognition - applying cognitive tutor principles to the tutoring of help seeking. *Metacognition and Learning, 2*(2), 125-140.

Romero, C., & Ventura, S. (2010). Educational Data Mining: A Review of the State-of-the-Art. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 40*(6), 601-618.

Rosenthal, R., & Rosnow, R. (1991). *Essentials of Behavioral Research: Methods and Data Analysis (2nd Edition).* Boston, MA: McGraw-Hill.

Rowe, J., & Lester, J. (2010). Modeling User Knowledge with Dynamic Bayesian Networks in Interactive Narrative Environments. In C. G. Youngblood, & V. Bulitko (Ed.), *Proceedings of the 6th Annual AI and Interactive Digital Entertainment Conference, AIIDE 2010* (pp. 57-62). Palo Alto, CA: AAAI Press.

Sao Pedro, M. A., Gobert, J. D., & Raziuddin, J. (2010). Comparing Pedagogical Approaches for the Acquisition and Long-Term Robustness of the Control of Variables Strategy. In K. Gomez, L. Lyons, & J. Radinsky (Ed.), *Learning in the Disciplines: Proceedings of the 9th International Conference of the Learning Sciences, ICLS 2010, Volume 1, Full Papers* (pp. 1024-1031). Chicago, IL: International Society of the Learning Sciences.

Sao Pedro, M., Gobert, J., Heffernan, N., & Beck, J. (2009). Comparing Pedagogical Approaches for Teaching the Control of Variables Strategy. *N.A. Taatgen & H. vanRijn (Eds.), Proceedings of the 31st Annual Meeting of the Cognitive Science Society* (pp. 1294-1299). Amsterdam, Netherlands: Cognitive Science Society.

Schauble, L., Klopfer, L., & Raghavan, K. (1991). Students' Transition from an Engineering Model to a Science Model of Experimentation. *Journal of Research in Science Teaching, 28*(9), 859-882.

Schraw, G. (2007). The Use of Computer-based Environments for Understanding and Improving Self-regulation. *Metacognition and Learning, 2*(2-3), pp. 169-176.

Schraw, G. (2009). A Conceptual Analysis of Five Measures of Metacognitive Monitoring. *Metacognition and Learning, 4*(1), 33-45.

Schunn, C. D., & Anderson, J. R. (1998). Scientific Discovery. In J. R. Anderson, *The Atomic Components of Thought* (pp. 385-428). Mahwah, NJ: Lawrence Erlbaum Associates.

Shores, L., Rowe, J., & Lester, J. (2011). Early Prediction of Cognitive Tool Use in Narrative-Centered Learning Environments. In G. Biswas, S. Bull, J. Kay, & A. Mitrovic (Ed.), *Proceedings of the 15th International Conference on Artificial Intelligence in Education, AIED 2011. LNCS 6738*, pp. 320-327. Auckland, New Zealand: Springer.

Stevens, R., Soller, A., Cooper, M., & Sprang, M. (2004). Modeling the Development of Problem Solving Skills in Chemistry with a Web-Based Tutor. In J. C. Lester, R. M. Vicari, & F. Paraguacu (Ed.), *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004. LNCS 3220*, pp. 580-591. Maceio, Alagoas, Brazil: Springer.

Strand-Cary, M., & Klahr, D. (2008). Developing elementary science skills; Instructional effectiveness and path independence. *Cognitive Development, 23*(4), 488-511.

van der Aalst, W. (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Berlin-Heidelberg: Springer-Verlag.

Veenman, M., Van Hout-Worters, B., & Afflerback, P. (2006). Metacognition and Learning: Conception and Methodological Considerations. *Metacognition and Learning, 1*(1), 3-14.

Walonoski, J., & Heffernan, N. (2006). Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. In M. Ikeda, K. Ashlay, & T.-W. Chan (Ed.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems, ITS 2006. LNCS 4053*, pp. 382-391. Johngli, Taiwan: Springer-Verlag.

Winne, P., & Hadwin, A. (1998). Studying as Self-Regulated Learning. In D. J. Hacker, J. Dunlosky, & A. Graesser, *Metacognition in Educational Theory and Practice* (pp. 277-304). Hillsdale, NJ: Lawrence Erlbaum Associates.

Winne, P., Nesbit, J., Kumar, V., Hadwin, A., Lajoie, S., Azevedo, R., et al. (2006). Supporting Self-Regulated Learning with gStudy Software: The Learning Kit Project. *Technology, Instruction, Cognition, and Learning Journal, 3*, 105-113.