Using Text Replay Tagging to Produce Detectors of Systematic Experimentation Behavior Patterns

Michael A. Sao Pedro[1], Ryan S.J.d. Baker[2,1], Orlando Montalvo[2], Adam Nakama[2], and Janice D. Gobert[2, 1]
{mikesp, rsbaker, amontalvo, nakama, jgobert}@wpi.edu
[1]Computer Science Department, Worcester Polytechnic Institute
[2]Social Science and Policy Studies Department, Worcester Polytechnic Institute

Abstract. We present machine-learned models that detect two forms of middle school students' systematic data collection behavior, designing controlled experiments and testing the stated hypothesis, within a virtual phase change inquiry learning environment. To generate these models, we manually coded a proportion of the student activity sequence clips using "text replay tagging" of log files, an extension of the text replay method presented in Baker, Corbett & Wagner (2006). We found that feature sets based on cumulative attributes, attributes computed over all predecessor clips, yielded better detectors of CVS-compliant and hypothesis-testing behavior than more local representations of student behavior. Furthermore, our detectors classify behaviors well enough to use them in our learning environment to determine which students require scaffolding on these skills.

1 Introduction

While inquiry learning is generally considered an important part of science education at all levels, many students lack inquiry skills [14]. Students have difficulties focusing on relevant variables, stating testable hypotheses, drawing correct conclusions from experiments, and linking hypotheses and data. They also struggle with basic experimental processes such as designing effective experiments, translating theoretical variables from their hypotheses into manipulable variables, and adequately monitoring their progress [10]. To that end, we are developing a learning environment, Science Assistments (http://users.wpi.edu/~sci_assistments), with the goal of assessing and scaffolding students' scientific inquiry as they engage in inquiry using interactive microworlds [13]. We place special emphasis on students' development of skills for conducting experiments since it has been argued that learning to correctly plan and execute controlled experiments is necessary to the development of other scientific inquiry skills [13]. In order to properly identify students needing inquiry support, we must be able to distinguish students who exhibit appropriate systematic behaviors from those who do not.

In this paper, we present machine-learned models for detecting two forms of systematic data collection behavior exhibited as students conduct experiments in a phase change microworld. The first behavior we detected was designing controlled experiments using the Control of Variables Strategy (CVS) [9], a strategy stating that one should change only the variable to be tested, the target variable, while keeping all extraneous variables constant to test the effects of that variable on an outcome. The second was collecting data to test a stated hypothesis, as opposed to collecting data that does not pertain to the stated hypothesis. To train our behavior detectors we generated training instances by manually inspecting and coding a proportion of student activity sequences using "text replay tagging" of log files. Similar to a video replay or screen replay, a text replay [2] is a pre-specified chunk of student actions presented in text that captures information such as

each action's time, type, widget selection, and input selection. Our approach leverages the success of [4, 6] in using text replays to provide training instances for machine-learned detectors of gaming the system within intelligent tutors. We are building detectors of these constructs, as opposed to detectors that identify specific kinds of unsystematic behavior, with the eventual goal of auto-scoring students' systematicity. Despite the ill defined nature of science inquiry, we can tutor students' inquiry skills with this approach, an approach similar to model tracing (cf. [16]).

This approach differs from previous text replays in two ways. First, whereas text replays allow for the classification of a replay clip as a single category out of a set of categories, text replay tagging allows multiple tags to be associated with one clip. For example, a clip may be tagged as CVS-compliant, hypothesis testing-compliant, both, or neither. Second, the behaviors we are studying are temporally more coarse-grained than in [4] or [6], requiring the display of the entire sequence of work on part of a problem rather than specific attempts to answer a problem or problem step. This permits coders to obtain a more comprehensive view of students' inquiry processes necessary for labeling processes like these that unfold over time. After producing these "gold standard" classifications, we summarized each student's activity sequences by creating a feature set from the data and used classification methods to find models that predict the labels from the data. In accordance with our data, we considered problem-level features of the student data rather than step or transaction-level data, unlike in many prior EDM models of student behavior (e.g. [1, 4, 6, 8, 20]).

Past research has attempted to model and analyze inquiry behavior using knowledge engineering approaches. In [7], the authors defined rules that encapsulated behaviors for differing levels of systematic experimentation skill when solving problems with interactive genetics simulations. They defined their rules over a set of domain-specific features extrapolated from student interactions, such as the types of genetic crosses made and if crosses were repeated, and domain-general features, such as time spent solving a problem. In [19], the authors constructed an ACT-R model based on an assessment of skill differences in novices and experts that designed and interpreted psychological experiments within a computerized environment. They developed a detailed set of rules and hierarchical high and low-level goals and actions to represent the cognitive processes of how an expert hypothesized, explored, analyzed and concluded about two competing theories. Finally, they tested the efficacy of their model by adding and removing key productions and comparing the model's simulated performance to experts and novices. Like both these approaches, we use low-level student actions as a basis for creating our behavioral models and, particularly like [19], we are interested in quantifying how well our detectors predict behaviors. However, our approach is different in that it does not prescribe rules for systematicity; instead, given data, human classified labels, and a feature set, we use machine learning techniques to discover rules. This approach has several advantages. The resulting models capture relationships that humans cannot easily intuit. They also identify boundary conditions more precisely than knowledge engineering approaches. Finally, unlike knowledge engineering approaches, they are easier to verify, since cross-validation is possible.

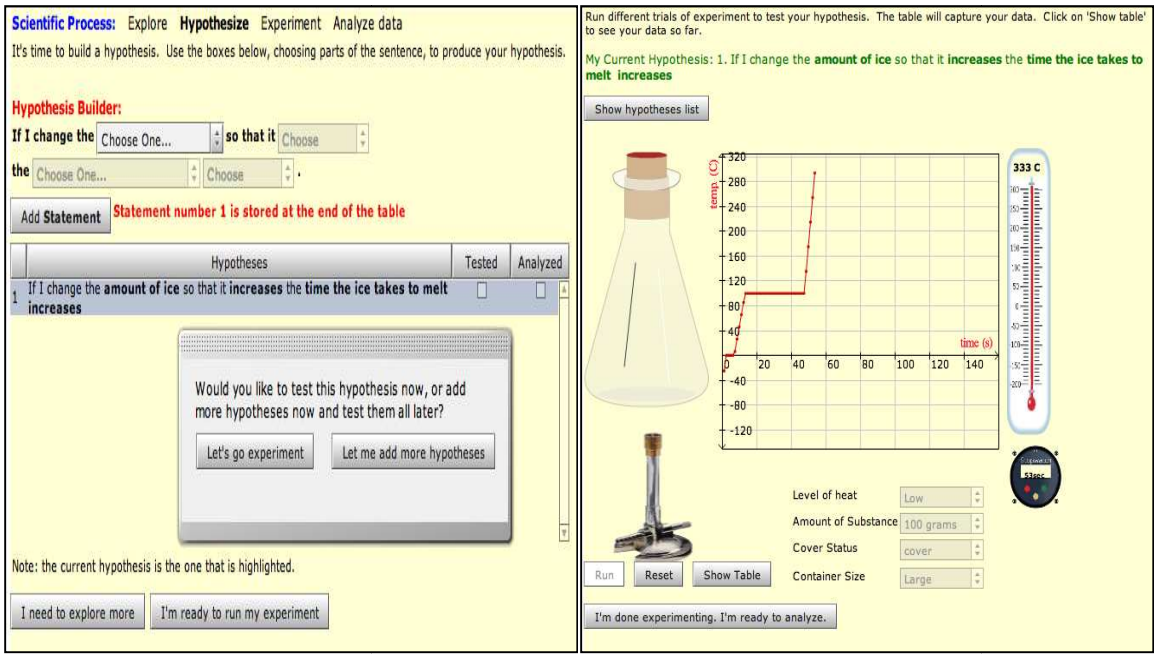

Figure 1. Hypothesizing widget (left) and data collection panel (right) for the phase change microworld.

2 Learning Environment

Our phase change environment (Figure 1), hosted by Science Assistments [13], enabled students to engage in authentic inquiry using a microworld and inquiry support tools. Each problem in our learning environment required students to conduct full experiments to determine if a particular independent variable, e.g., container size, affected various outcomes like the melting point or boiling point of a substance. For a given variable, students demonstrated proficiency by hypothesizing, collecting data, reasoning with tables and graphs, analyzing data, and communicating their findings about how that variable affected the outcomes. We helped organize students' inquiry processes by arranging these tasks into different inquiry phases: "observe", "hypothesize", "experiment", and "analyze data". Students start in the hypothesizing phase and move between phases in a suggested order but can navigate back and forth between some of the inquiry phases. For example, from the "analysis" panel students could collect more data by returning to the "experiment" phases, they could create new hypotheses by returning to the "hypothesize" phase (starting a new inquiry loop), or could submit their final experimentation procedures and analyses and begin the next problem. While in the hypothesizing phase (left side of Figure 1), they could either explore the microworld or begin collecting data in the experiment phase (right side of Figure 1). Finally, within the experiment phase, students can only move to the analysis phase.

This learning environment has a moderate degree of learner control, less than in purely exploratory learning environments [1], but more than in classic model-tracing tutors [16] or constraint-based tutors [18]. Though our scaffolding restricts when students can switch

inquiry phases, there is enough freedom such that students could approach these inquiry tasks in many ways. For example, a student could choose to specify only one hypothesis like, “If I change the container size so that increases, the melting point stays the same” (left side of Figure 1), and then test that single hypothesis. Alternately, they could generate several and test them all sequentially. While experimenting (right side of Figure 1), a student could set up and run as many different experiments as they desired, including repeating the same trial multiple times. A table tool was provided within the learning environment to display the results of the student’s previous experiments and to display their hypothesis list to determine which experiments to run next.

As students engage in inquiry using our tools and microworld, they can exhibit several different inquiry behaviors. Students acting in a systematic manner [7] collect data by designing and running controlled experiments that test their hypotheses. Also, students acting systematically use the table tool and hypothesis viewer in order to reflect and plan for additional experiments. Students who are unsystematic, by contrast, may exhibit haphazard behaviors such as: constructing experiments that do not test their hypotheses, not collecting enough data to support or refute their hypotheses, not following CVS, running the same experimental setup multiple times, or failing to use the inquiry support tools to analyze their results and plan additional trials [10].

3 Dataset

Participants were 148 eighth grade students, ranging in age from 12-14 years, from a public middle school in Central Massachusetts. These students used the phase change microworld as part of a broader study to determine if inquiry skills learned in one domain will transfer to inquiry skill in other domains [13]. Students engaged in authentic inquiry problems using the phase change and density microworlds within the Science Assistments learning environment. Students were randomly assigned to one of two conditions that counterbalanced the order in which students engaged in a science domain: phase change followed by density vs. density followed by phase change. In this paper, we discuss detectors of systematic data collection for student actions within the phase change microworld only, as the version of the density microworld used lacked the hypothesizing scaffold used in the phase change microworld. In building these detectors, we look specifically at what students did in the “hypothesizing” and “experimenting” phases of inquiry. As part of the phase change activities, students attempted to complete four tasks using our interactive tools.

Each of these students completed at least one data collection activity in the phase change environment (two other students did not use the microworld, and were excluded from analysis). As students solved these tasks, we recorded fine-grained actions within the inquiry support tools and microworlds. The set of actions logged included creating hypotheses, setting up experiments, showing or hiding support tools, running experiments, creating interpretations of data, and transitioning between inquiry activities (i.e. moving from hypothesizing to data collection). Each action’s type, current and previous values (where applicable – for instance, a variable’s value), and timestamp were recorded. In all, 27,257 student actions for phase change were logged. These served as

the basis for generating text replay clips consisting of contiguous sequences of actions specific to experimenting.

4 Text Replay Tagging Methodology

In designing our text replays, it was necessary to use a coarser grain-size than in prior versions of this method (e.g. [4, 6]). In particular, it was necessary to show significant periods of experimentation so that coders could precisely evaluate experimentation behavior relative to stated hypotheses. We decided our text replays should include actions from only the hypothesis and the experimenting phases. Another important issue was that trial run data from one hypothesis test could be used in another hypothesis test to make inferences about the hypothesis at hand (i.e. comparing a current trial to one conducted earlier). To compensate for this, we code using both the actions in testing the current hypothesis, and cumulative measures that include actions performed when testing previous hypotheses. Hence, each tagged clip focuses on actions in the current part of the inquiry process, but may take into account the context of the cumulative section.

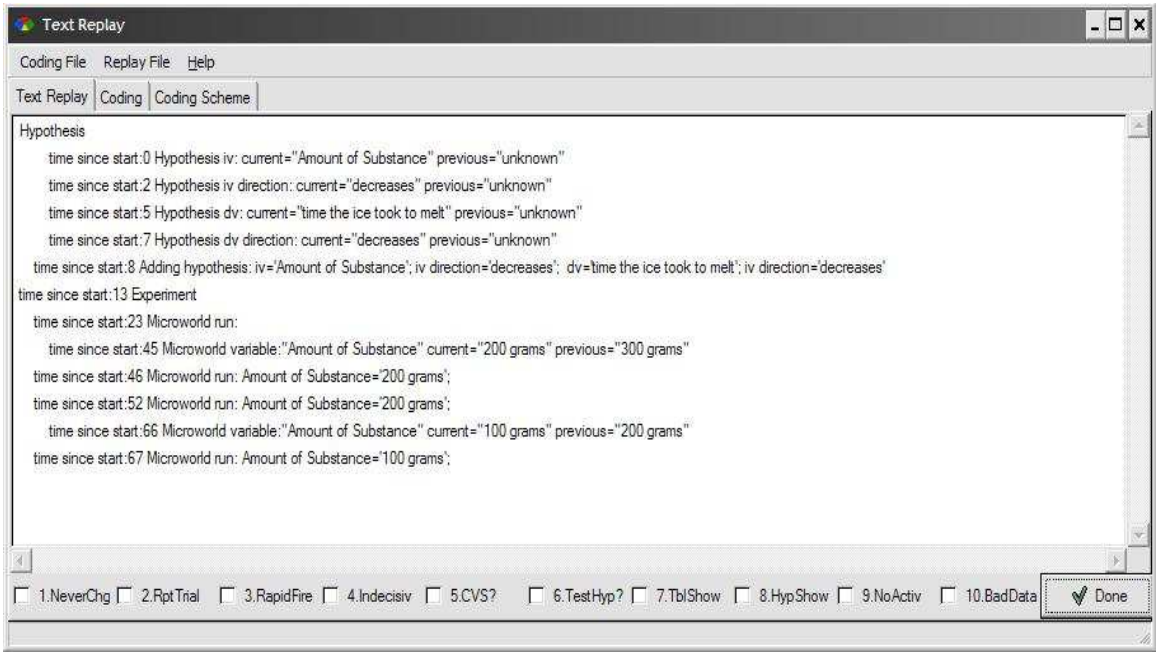

Figure 2. Text Replay Tagging Tool with an example student's clip coded as running repeated trials, following CVS, and testing stated hypotheses.

To support coding in this fashion, a new tool for text replay tagging was developed in Ruby, shown in Figure 2. The start of the clip is triggered by a hypothesis variable change after the beginning of a new problem. The tool displays all student actions (hypothesis and experiment) until the student transitions to the analysis stage. Subsequent clips include previous clips and any single new cycle which includes the Hypothesis and Experiment stage. A clip could be tagged with one of 9 tags corresponding to data collection behaviors: "Never Change Variables", "Repeat Trials", "Non-Interpretable Action Sequence", "Indecisiveness", "Used CVS", "Tested Hypothesis", "Used Table to Plan", "Used Hypothesis Viewer to Plan", "No Activity",

and one extra category for unclassifiable clips, "Bad Data", for a total of 10 coding categories. Specifically for the analyses in this paper, we tagged a clip as "Used CVS" if the clip contained actions indicative of designing and running controlled experiments. "Tested Hypothesis" was chosen if the clip had actions indicating attempts to test the *stated* hypotheses, regardless of whether or not proper CVS procedure was used.

4.1 Clip Tagging Procedure

Two coders (the first and fourth authors) tagged the data collection clips using at least one of the ten tags. To ensure that a representative range of student clips were coded, we stratified our sample of the clips on condition, student, problem, and within-problem clip order (e.g. first clip, second clip, etc.) The corpus of hand-coded clips contained exactly one randomly selected clip from each problem each student encountered, resulting in 571 clips. Each coder tagged the first 50 clips; the remaining clips were split between the coders. For the 50 clips tagged by each coder, there was high overall tagging agreement, average κ=0.86. Of particular relevance to this study, there was also better agreement on the CVS and testing hypotheses tags, κ=.69 and κ=1.00 respectively, than has been seen for previous text replay approaches that led to successful behavior detectors (e.g. [4, 6]).

4.2 Feature Distillation

Features were extracted relevant to the 10 categories of behavior within the microworld. These included: all actions, total trial runs, incomplete trial runs, complete trial runs, pauses, data table display, hypothesis list display, field changes in hypothesis builder (left side of Figure 1), hypotheses made, and microworld variable changes. For each category, we traced the number of times the action occurred and the time taken for each action. For timing values, we also computed the minimum, maximum, standard deviation, mean and mode for each student and compared these values relative to all other students. We also included the number of pairwise trials where only one independent variable differed between them and a count for repeated trials, trials with the same independent variable selections. These last two had no time associated with them.

We extracted feature values from student actions as follows. As stated in Section 2, student microworld activity was divided into tasks, each focusing on a specific independent variable. Also, within a task, the student could make and test several hypotheses. For each of the categories, we extracted data for each hypothesis the student tested (local data), and across all hypotheses in the set (cumulative data). We did this because within each set, the data table accumulated the trial run data across hypotheses, enabling students to compare trial runs testing previous hypotheses with the runs made in the current hypothesis.

4.3 Machine Learning Algorithms

Machine-learned detectors of the two behavioral patterns of interest, CVS and hypothesis testing, were developed within RapidMiner 4.6 [17]. Detectors were built using J48 decision trees, with automated pruning to control for over-fitting, the same technique used in [4, 20]. Before running the decision tree algorithm, we filtered redundant features

correlated at or above 0.6. Six-fold cross-validation was conducted at the student level (e.g. detectors are trained on five 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. We assessed the classifiers using two metrics. First, we used A' [15]. A' is the probability that if the detector is comparing two clips, one involving the category of interest (CVS or Hypothesis Testing) 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 [15]. 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 used at the level of clips, rather than students. Statistical tests for A' are not presented in this paper. The most appropriate statistical test for A' in data across students is 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. [3]) – however, the standard error formula for A' [15] requires multiple examples from each category for each student, which is infeasible in the small samples obtained for each student 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 Kappa (κ), which assesses whether the detector identifies 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. As Kappa looks only at the final label, whereas A' looks at the classifier's degree of confidence, A' can be more sensitive to uncertainty in classification than Kappa.

5 Results

We constructed and tested detectors using our corpus of hand-coded clips. The CVS and hypothesis testing detectors were constructed from a combination of the subset of the first 50 clips that the two coders agreed on, and the remaining clips, tagged separately by the two coders. Of all clips, 31.2% were tagged as showing evidence of CVS and 34.4% were tagged as showing evidence of collecting data to test specified hypotheses.

Detectors were generated for each behavior using J48 decision trees and two sets of attributes, cumulative and non-cumulative attributes. As a reminder, non-cumulative attributes were tallied over a single clip, irrespective of other clips, whereas cumulative attributes included data from earlier clips from the same problem. Thus, four different detectors were constructed. The CVS detector using cumulative attributes (A'=.85, κ=.47) appeared to perform better than the detector built with non-cumulative attributes (A'=.81, κ=.42). Likewise, the hypothesis testing detector built with cumulative attributes (A'=.86, κ=.46) scored higher on our metrics than the non-cumulative detector (A'=.84, κ=.44). We believe the detectors built from cumulative attributes perform better because students may perform actions within particular clips that, when taken in conjunction with actions from previous clips, represent a more complete picture of student behavior. For example, while analyzing results a student may realize they need to run one more

experiment to correctly test their hypothesis (which would start a new clip). The human coders would correctly label this as CVS and testing a hypothesis *in reference to the previous context*, but values for noncumulative attributes would most likely indicate that the student was not systematic because these attributes' values are not computed based on previous clips.

6 Discussion and Conclusions

The goal of this research was to develop machine-learned models that can automatically detect if a student is systematic in their inquiry, particularly in their data collection actions, using text replay tagging. This work showed that combining text replay clip tagging of low-level student actions and machine learning can lead to the successful development of behavior detectors in an ill-defined domain such as scientific experimentation. This work also presents a contribution to the text replay process since it is more efficient to code a clip with multiple tags. Our results were promising; using cumulative attributes, we can distinguish students who are successfully applying the Control of Variables Strategy (CVS) in the phase change environment from students not applying CVS 85% of the time and can distinguish students testing their hypotheses 86% of the time. Furthermore, the Kappa values indicate that each of these detectors are substantially better than chance. In other words, though these detectors are not perfect, they can be used to select students for scaffolding. Since they are not perfect, some students may receive help when they do not need it and vice versa. Hence, interventions used should be fail-soft, relatively non-harmful when given incorrectly. As such, we aim to use these detectors to determine which students will receive scaffolding.

An important area of future work will be to improve our detectors' A' and Kappa. To this end, we plan to add lesson-wide attributes, learner attributes, and data on the other tags used to critique a clip. Lesson-wide attributes, such as task attempt number, that can benchmark a students' experience within our environment may aid in predicting systematicity, in coordination with other features. Additionally, rather than treating learner characteristics, such as prior knowledge, as external predictors of systematicity, we could incorporate those measurements into the detectors themselves. Similar to computing average attribute differences between clips (i.e. computing the difference in number of trials run for the given clip and the average number of trials run for all clips), we could compute differences between students with similar learner characteristics. Similarly, rather than using systematicity to predict content knowledge, we could incorporate student prior knowledge of content and inquiry using our standardized-test style questions [13]. Another important area of future work will be to generalize and train our detectors across different microworlds (cf. [5]) to increase their applicability across middle school science learning.

This approach also enables us to research the interactions between content knowledge and authentic inquiry performance within our learning environments. Being able to classify students as systematic according to different skills, e.g. testing hypotheses and CVS, will enable us to determine if skill proficiency in solving authentic inquiry problems will predict skill proficiency in solving standardized test-style inquiry questions. We can also determine the degree to which systematic behavior predicts robust

content knowledge. Finally, by developing and generalizing detectors across domains, we can determine the degree to which authentic inquiry skill transfers between domains. As such, these models have considerable potential to enable future "discovery with models" analyses that can shed light on the relationship between a student's mastery of systematic experimentation strategies and their domain learning.

Acknowledgements

This research is funded by the National Science Foundation (NSF-DRL#0733286) 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

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

[2] Baker, R. S. J. d., Corbett, A. T., Wagner, A. Z. Human Classification of Low-Fidelity Replays of Student Actions. *Proceedings of the Educational Data Mining Workshop at the 8th International Conference on Intelligent Tutoring Systems*, 2006. p. 29-36.

[3] Baker, R. S. J. d., Corbett, A. T., Aleven, V. Improving Contextual Models of Guessing and Slipping with a Truncated Training Set. *Proceedings of the 1st International Conference on Educational Data Mining*, 2008. p. 67-76.

[4] Baker, R. S. J. d., de Carvalho, A. M. J. A. Labeling Student Behavior Faster and More Precisely with Text Replays. *Proceedings of the 1st International Conference on Educational Data Mining*, 2008. p. 38-47.

[5] Baker, R., Corbett, A., Roll, I., Koedinger, K. Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction*, 2008, 18(3), p. 287-314.

[6] Baker, R., Mitrovic, A., Mathews, M. Detecting Gaming the System in Constraint-Based Tutors. *Proceedings of the 3rd International Conference on User Modeling and Personalization*, in press.

[7] Buckley, B. C., Gobert, J., Horwitz, P. Using log files to track students' model-based inquiry. *Proceedings of the 7th International Conference on Learning Sciences*, 2006. p. 57-63.

[8] Cetintas, S., Si,., Xin, Y. P., Hord, C. Automatic Detection of Off-Task Behaviors in Intelligent Tutoring Systems with Machine Learning Techniques. *IEEE Transactions on Learning Technologies*, in press.

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

[10] de Jong, T. Computer Simulations - Technological Advances in Inquiry Learning. *Science*, 2006, 312, p. 532-533.

[11] Gobert, Janice (Principal Investigator); Heffernan, Neil; Koedinger, Ken; Beck, Joseph (Co-Principal Investigators): ASSISTments Meets Science Learning (AMSL; R305A090170). Awarded February 1, 2009 from the U.S. Dept. of Education, 2009

[12] Gobert, Janice (Principal Investigator); Heffernan, Neil; Ruiz, Carolina; Kim, Ryung (Co-Principal Investigators): AMI: ASSISTments Meets Inquiry (NSF-DRL# 0733286). Awarded September 2007 from the National Science Foundation, 2007

[13] Gobert, J., Heffernan, N., Feng, M., Sao Pedro, M., Beck, J. ASSSTments for Science and Math: Assessing and Assisting. *Presented at the Annual Meeting of the American Educational Research Association. San Diego, CA, April 13-17*, 2009.

[14] Gobert, J., Schunn, C. Supporting Inquiry Learning: A Comparative Look at What Matters. *A symposium presented at the Annual Meeting of the American Educational Research Association. Chicago, IL, April 9-13*, 2007.

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

[16] Koedinger, K. R., Corbett, A. T.: Cognitive Tutors: Technology Bringing Learning Sciences to the Classroom. In Sawyer, R. K. (Eds.) *The Cambridge Handbook of the Learning Sciences*, 2006. New York: Cambridge University Press. p. 61-77.

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

[18] Mitrovic, A., Mayo, M., Suraweera, P., Martin, B. Constraint-Based Tutors: A Success Story. Springer-Verlag (Eds.) *Proceedings of the Industrial & Engineering Application of Artificial Intelligence & Expert Systems Conference IEA/AIE-2001*, 2001. p. 931-940.

[19] Schunn, C., Anderson, J.: Scientific Discovery. In Anderson, J. (Eds.) *The Atomic Components of Thought*, 1998. Mahwah: Lawrence Erlbaum Associates, Inc. p. 385-428.

[20] Walonoski, J. A., Heffernan, N. T. Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 2006. p. 382-391.