Muldner, K., Burleson, W., van de Sande, B., & VanLehn, K. (2010). An analysis of gaming behaviors in an intelligent tutoring system. In V. Aleven, J. Kay & J. Mostow (Eds.), Intelligent Tutoring Systems: 10th International Conference, ITS 2010 (pp. 184-193). Heidelberg, Germany: Springer.

An Analysis of Gaming Behaviors in an Intelligent Tutoring System

Kasia Muldner[1], Winslow Burleson[1], Brett Van de Sande[1], Kurt VanLehn[1]

[1] Arizona State University
{katarzyna.muldner, winslow.burleson, bvds, kurt.vanlehn}@asu.edu

Abstract. We present results from an analysis of students' shallow behaviors, i.e., *gaming*, during their interaction with an Intelligent Tutoring System (ITS). The analysis is based on six college classes using the Andes ITS for homework and test preparation. Our findings show that student features are a better predictor of gaming than problem features, and that individual differences between students impact where and how students game.

1 Introduction

Students have long found ways to avoid reasoning about instructional materials, e.g., by copying from examples to generate problem solutions [1] or by avoiding effective study strategies such as self explaining domain principles [2]. In human tutoring contexts, students often passively listen to tutors' didactic explanations, without providing substantial follow ups even though more active participation is needed for effective learning [3]. These behaviors also occur in students' interactions with *intelligent tutoring systems* (ITSs). A name given to shallow reasoning in the context of an ITS is gaming, "*attempting to succeed in a learning environment by exploiting properties of the system rather than by learning the material*" [4]. Not surprisingly, gaming can be detrimental to learning [5], and so there have been efforts in detecting [6, 7], understanding [4, 8, 9] and preventing [10, 11] gaming.

We add to this research by presenting an in-depth analysis of log data corresponding to several years worth of students interacting with Andes, a tutoring system for Newtonian physics [12]. To identify gaming episodes in this data, we applied a computational gaming detector that we calibrated with a hand-analysis of the data. Contrary to some prior findings (e.g., [4]), we found that gaming is best predicted by student features, rather than instructional aspects. This lead us to perform a descriptive analysis of students' gaming behaviors that focused in part on understanding which tutor actions lead students to game. While we found individual differences between low and high gamers, high-level hints were one of the most gamed features. However, in contrast to other work [4], our analysis suggests that poor hint usability may not be the culprit, and so that other factors such as student motivation (or lack of) are at play.

We begin with a survey of related work, and then present our gaming detector, the log data analysis and results, and finally a discussion of our findings and future work.

2 Related Work

Several approaches for detecting gaming have been used. Some research has relied on human observers for real-time gaming identification [5]. This approach is challenging as observers may miss nuances in a fast-paced classroom environment and so others have turned to post hoc hand labeling of log data [4]. The latter approach affords the human coder time to consider all student actions but is costly, since copious amounts of data must be hand labeled. A potential issue with using human coders is that they may be inconsistent in identifying gaming episodes, something that machine algorithms for gaming identification address [6, 7, 13, 14].

A key challenge is understanding what causes gaming. Some researchers propose that gaming is due to features of the instructional materials, including (poor) ITS design. For instance, Baker et al. [4] found that ITS features, such as unhelpful hints and non-intuitive toolbar icons, explained more of the variance in the data than prior approaches using student features; a similar result was obtained in [15]. Other work focuses on identifying student characteristics that drive gaming. For instance, since students game on steps that they do not know [13], it has been proposed that item difficulty leads to gaming. Another student characteristic influencing gaming is affect, with boredom being the most frequent emotion to precede gaming [8]. There has also been research on how performance goal orientation impacts gaming [16]; this work failed to find the anticipated link between gaming and performance goals.

As far as gaming prevention is concerned, a number of strategies have been developed, including the use of animated agents that show disapproval when gaming occurs [17], software design via a mandatory delay before a student can ask for a hint [10, 18], and/or by letting students choose the hint level [19].

3 The Data and Gaming Detector

The Data. Our data, obtained from the Pittsburgh Learning Center DataShop, corresponds to logs of students using the Andes ITS [12] for assigned class homework and test preparation (from six different physics classes over the span of about three years). Andes tutors Newtonian physics and is described in detail elsewhere (e.g., [12]); here we provide a very brief overview. Students solve problems in the Andes interface by drawing diagrams and typing equations. Such a user interface action will be called an *entry*. Andes provides immediate feedback for correctness on students' entries, by coloring the entry red (incorrect) or green (correct). As students solve problems, they can ask Andes for a hint; the Andes hint sequence starts out general and ends with a bottom-out hint that indicates precisely the step to enter (e.g., "*Why don't you continue with the solution by working on setting the pressure at a point open to the atmosphere " … "Write the equation Pa = Pr0*"). To discourage students from always going to the bottom-out hint, Andes assigns a score to each problem, which is decremented slightly every time a bottom-out hint is requested.

The Gaming Detector. After irrelevant actions are removed from the log data, a log consists of a time-stamped sequence of *tutor-student turn* pairs (e.g., tutor indicates an

Table 1: *Tutor-Student* turn pairs (gamed cells shaded).

	(a) Student: hint request		(b) Student: Entry	
	fast	slow	fast	slow
(1) Tutor: bottom-out hint	Skip hint (S)	-	Copy hint (C)	-
(2) Tutor: High-level hint	Skip hint (S)	-	-	-
(3) Tutor: Incorrect (Red)	-	-	Guess (G)	-
(4) Tutor: Correct (Green)	No planning (P)	-	-	-

entry is incorrect, student responds by asking for a hint). To address our research questions, we needed to know which of these turn pairs corresponded to gaming. Given that our data comprised over 900,000 pairs, manual analysis was not feasible. Thus, we first hand-analyzed a fragment of the log data to identify rules to detect gamed turn pairs, which we then encoded into a computational gaming detector that could automatically label the data. We then applied the detector, hand-checking its output on a new data fragment, revising as necessary.

For purposes of this analysis, we considered the following *tutor* turns: (1) coloring an entry red (incorrect), (2) coloring an entry green (correct), (3) giving a bottom-out hint, or (4) giving a high-level hint (we did not further subdivide the *high-level* hints since the number and characteristics of such hints varied considerably). We classified a *student's* turn as either (a) asking for a hint or (b) generating an entry. Thus, there are 4*2=8 types of turn pairs (see Table 1). Each turn pair has a time duration associated with it, which is how long the student paused between seeing the tutor's turn and starting to take action. We assume that turn pairs with long durations are *not* gaming. Of the eight possible turn pairs with short durations (see Table 1), we consider the following five to be gaming: (1-2) *Skipping a hint:* the tutor presents a hint and the student skips the hint by quickly asking for another hint (see '*S*' cells in Table 1); (3) *Copying a hint:* the tutor presents a bottom-out hint and the student quickly generates a solution entry, suggesting a shallow copy of the hint instead of learning of the underlying domain principle[1] (see 'C' cell, Table 1); (4) *Guessing:* after the tutor signals an incorrect entry, the student quickly generates another *incorrect*[2] entry, suggesting s/he is guessing instead of reasoning about why the entry is incorrect (see 'G' cell in Table 1); (5) *Lack of planning:* after the tutor signals a correct entry, the student quickly asks for a hint, suggesting reliance on hints for planning the solution (see 'P' cell, Table 1). Note that item 1, *skipping hints*, does not take into account the possibility that a student may copy the hint and *then* reason about it. This was explored in [20], by analyzing time after a hint was copied. Time alone, however, does not necessarily indicate that the student is reasoning about the hint, since they may be, for instance, thinking about the next step. Thus, for the time being, we decided to only consider time before an entry is generated, as we felt this was more likely to correspond to reasoning about the entry.

[1] A high-level hint followed by a fast entry is not gaming since you can't copy high-level hints.

[2] This is the only student entry where we take into account correctness of the student entry, as not doing so might incorrectly classify fixing slips as gaming.

Accurate gaming detection relies on having reasonable time thresholds, one for each of the five gamed turn pairs. To set the threshold, we obtained a value for each pair based on our review of the log file data. As a final check, we obtained a frequency distribution graph for each of the gamed pairs. The graph allowed us to ensure that the threshold we chose was not unrealistic (e.g., so high that all students would be considered gaming). Note that we were conservative when setting our thresholds: for instance, we set the *skipping* hint threshold $T = < 3$sec. While this threshold may not afford enough time to read all of a hint, it captures instances when students are skipping most of the hint.

4 Results

Our analysis is based on applying the above-described gaming detector to data from a set of 318 unique problems and 286 students. We now describe our results.

4.1 What is a Better Predictor of Gaming: Student or Problem?

As we mentioned above, a central question pertains to what causes gaming, and in particular, whether student or problem features better predict gaming. To address this question, we obtained the following measures:

$PerGaming_{sp}$	*percentage of gaming by a student s on a problem p*	(1)
$\sum_{p=0}^{p=N} \text{PerGaming}_{sp} / N$	*i.e., average gaming by a student s across all N problems p solved by that student*	(2)
$\sum_{s=0}^{s=M} \text{PerGaming}_{sp} / M$	*i.e., average gaming on a problem p across all M students s*	(3)

We used *problem* as the unit of analysis (see equation 1; equations 2 and 3 rely on it). Some research has used lesson as the primary unit of analysis [4]. In fact, the ideal unit would correspond to tutor-student turn pairs, as these are when student makes a game vs. no-game decision. However, we need a unit of analysis that can be compared across students, so that we can determine whether all students tend to game at "the same" place. It would be difficult to determine if turn-pair from one student are "the same" as a turn-pair from another student. The smallest unit of analysis that allows simple equivalence across students is the problem. Thus, we chose problems as the unit of analysis instead of lesson (too large) or tutor-student turn pairs (not equatable; too small). We used percentage of gaming (see equation 1) instead of raw values to avoid biasing the analysis towards, for instance, short problems.

To investigate predictors of gaming we conducted a linear regression analysis, with $PerGaming_{sp}$ (equation 1 above) as the dependent variable, and two independent variables: (1) *student*, the average gaming by a student *s* across all *N* problems *p* solved by that student (equation 2 above), and (2) *problem*, the average gaming on a

problem *p* across all *M* students *s* who solved that problem (equation 3 above). The model is significant ($F=16915$, $p < 0.001$), and accounts for 60.7% of the variance ($R^2 = .607$). In this model, both *student* and *problem* yield a significant correlation with the dependent variable (*student*: standardized coefficient=.658, $t=152.7$, $p<0.001$; *problem*: standardized coefficient=.325, $t=74.23$, $p<0.001$). If we enter the independent variables separately to analyze the variance explained by each, (1) the *student* variable accounts for 49.6% of the variance, while (2) the *problem* variable accounts for 18.6% of the variance.

To identify the impact of a particular data set (i.e., class/semester), we re-ran the regression analysis with a third independent variable, namely *data set id*. This variable explained only an additional 1% of the variance, showing that data set had at best a weak effect on gaming, and so we did not consider it in subsequent analysis.

Another way to verify whether students are more consistently gaming across problems or if instead problems are more consistent across students is to randomly sub-divide students (or problems) into buckets and then check for correlation between the buckets. To this end, we created two buckets by randomly assigning students to a given bucket. For each bucket, we obtained the average percentage of gaming for each problem in that bucket (equation 3 above), and then performed correlation analysis between buckets *A* and *B*. We found a high degree of association between the two data sets ($R^2=.89$ $p < 0.001$). That is, if a problem was often gamed by students in bucket A, then it was also often gamed by students in bucket B. We used an analogous technique to verify that problems were consistently gamed on between students (i.e., obtained two bucket by randomly assigning problems to a given bucket, and applied equation 2 above to obtain the average percentage of gaming by a student); the analysis yielded a high degree of association ($R^2=.963$, $p<0.001$). That is, if a student tended to game the problems in the A bucket, then that student also tended to game problems in the B bucket. Jointly, these analyses show that *students* are more consistent than *problems*: if a student is a high gamer on one half of the problem, then the student is also a high gamer on the other half; in contrast, if a problem is a high-gaming problem for half the students, then it is less likely to be a high-gaming problem for the other half. Thus, these analyses support the above regression results.

Yet another way to test our hypotheses is to examine histograms of gaming frequency. That is, we can look at how many students are high frequency gamers vs. middle vs. low frequency gamers. If individual differences among students are completely unimportant, and all students tend to solve roughly the same set of problems, then gaming frequency should be normally distributed. In fact, the distribution is significantly different from the normal (Shapiro-Wilks test of normality $W=.8$, $p<0.01$), and appears bimodal (see Figure 1, left). There seems to be one group of students who are frequent gamers, and another group who seldom game. This again suggests that individual differences play an important role in gaming frequency.

On the other hand, if the characteristics of problems are completely unimportant, then a histogram of the number of problems (y-axis) gamed at a certain range of frequencies (x-axis) should be normally distributed (Figure 1, right). This is in fact the case: the Shapiro-Wilks test of normality showed that the problem distribution is not significantly different from normal ($W=.9$, $p>0.05$). Thus, it appears once again that characteristics of students are more important than characteristics of problems in determining the frequency of gaming.

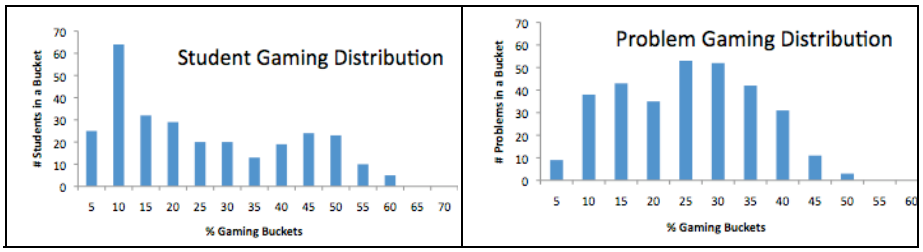

Figure 1: Student (left) and problem (right) gaming distributions. Each bucket contains students (or problems) with a gaming range (e.g., bucket 10 has 5% < gaming < 10%).

4.2 Gaming Profiles: How Much and Where are Student Gaming?

This section presents a descriptive analysis of the data, starting with how much students are gaming overall. On average, 22.5% of the tutor-student turn pairs were gamed. While this is higher than reported in [5], in that study students were in the presence of observers. This may have provided social deterrents for gaming, while in our study students were using Andes in private. We then analyzed where the gaming was occurring; Table 2 shows the results (the shaded cells indicate gamed turn pairs). In general, students most frequently took advantage of the opportunity to game when the tutor presented a high-level hint: on average, 18.4% of all actions corresponded to gaming on these hints; when given such a hint, students gamed 58.5% of the time.

In order to compare the gaming patterns of students who frequently gamed with those who infrequently gamed, we divided students into low gamers and high gamers based on a median split. On average, low gamers were significantly more likely than high gamers to game by guessing (46% vs. 13.2%; $F(1,283)=126$, $p<.01$). On the other hand, in contrast to low gamers, high gamers had a significantly higher proportion of skipped high-level hints (61.6% vs. 43.4%; $F(1, 283)=64$, $p<.01$), lack of planning (18.5% vs. 8.9%; $F(1,283)=159$, $p<.01$) and bottom-out hint copying (6.5% vs. 1.7% ; $F(1,283)=215$, $p<.01$).

4.3 Are Hints the Culprit or is it the Students?

Over all students' gaming opportunities (see Table 2), as well as proportion of gaming for high gamers, high-level hints elicited the most gaming. Thus, we wanted to explore how students used hints and if hints were helpful during problem solving.

Hint Viewing. The most basic analysis is to calculate time students spent on hints. To do so, we obtained the latency between the provision of a hint and the next student action. On average, students spent 9.2 sec. vs. 5.7 sec. on bottom-out vs. high-level hints. High gamers spent significantly less time on hints than low gamers, both on bottom-out hints (7.5sec vs. 10.9sec.; $F(1, 277)=71$, $p<.01$) and high-level hints (3.2sec vs. 8.1sec; $F(1,286)=246$, $p<.01$). Since the bottom-out viewing is well above the gaming threshold, on average, neither low or high gamers skipped bottom-out hints. In contrast, high gamers average viewing time for high-level hints is just above

Table 2: Gaming opportunities for each *Tutor–Student* turn pair. Shown in each cell: (1) the mean % of a student response given a tutor action over all 16 possible combinations, (2) (*mean % of a student response for that row's tutor action*).

	(a) Student: Hint Request		(b) Student: Entry	
	fast	slow	fast	slow
(1) Tutor: B-O Hint	0.02 *(.3)%*	*.2 (2)%*	1.8 *(23.6)%*	5.7 *(74)%*
(2) Tutor: H-L Hint	18.4 *(58.5)%*	5.8 *(18.3)%*	*.7 (2.3)%*	6.4 *(20.6)%*
(3) Tutor: Incorrect	2.5 (10.1)%	3 *(12.4) %*	5.4 *(21.9) %*	13.6 *(55.5)%*
(4) Tutor: Correct	5.2 *(15.6) %*	3.7 *(10.2)%*	14.1 *(38.2)%*	13.3 *(36)%*

Legend: *fast: student action < gaming threshold; slow: student action >gaming threshold; B-O: Bottom-out, H-L: High-level*

the gaming threshold of 3 seconds, showing that in contrast to low gamers, these students did not pay much attention to high-level hints.

Are Hints Helpful? Prior work suggests that a factor related to gaming is *reading hints does not influence solution entry success* [4]. Sophisticated techniques exist for analyzing the utility of help by looking at its impact on future student performance [21]. It is not clear, however, how these methods account for gaming, which can make it difficult to interpret results (e.g., if a student skips a hint repeatedly, is the hint not helpful or is the student unmotivated to use it?). Thus, for the time being, we analyze hint impact on short-term performance, i.e., can the student generate an entry after seeing a hint. Specifically, for each student, we obtained the percentage of time s/he was successful at generating a *correct* entry after receiving each type of hint (bottom out, high level). Note that (1) students may require several attempts to generate a correct entry and (2) if hint *B* is requested after hint *A* but prior to generating a correct entry, then hint *A* is not counted as "successful" for helping the student.

If for a moment we don't consider entry correctness, high gamers tried to generate an entry only 18% of the time after receiving a high-level hint, immediately asking for another hint the other 82% of the time. Low gamers, on the other hand, responded to a high-level hint with an entry 43% of the time. This is in contrast to bottom-out hints, when *both* low and high gamers responded to the hint with an entry about 97% of the time. When students did generate an entry after seeing a bottom-out hint, on average, they were successful in 90% of instances (i.e., obtained a correct entry). There was little difference between low and high gamers for this analysis (89% vs. 92%, respectively, NS difference). After high-level hints, students generated a correct entry 73% of the time. Again, there was little difference between low and high gamers (72% vs. 73%, respectively, NS difference). This suggests that high-level hints helped students generate the solution in about three out of four instances.

Now let's look at time and number of attempts needed to produce a correct entry. After bottom-out hints, on average students required 1.1 attempts (1.23 for low gamers vs. 1.19 for high gamers, NS), and took 29 sec. to do so (34sec. for low gamers vs. 23sec. for high gamers, $F(284,1)=4$, $p = .052$). After high-level hints, on average students required 1.83 attempts; here low gamers needed significantly fewer attempts than high gamers (1.66 vs. 2.01, $F(1,284)= 17$, $p<0.001$), suggesting that

perhaps the low gamers were more diligent about applying high-level hints. This conjecture is supported by the fact that low-gamers spent significantly longer than high-gamers to generate a correct entry after seeing a high-level hint (37 sec vs. 28 sec; $F(1,286)=9$, $p<0.01$).

5 Discussion and Future Work

A prerequisite for the design of effective interventions to discourage gaming is understanding its causes. Past research has shown that both student and instructional aspects influence gaming, but to date there does not exist agreement as to which is the stronger predictor. Baker et al. [4] argue that it is the latter, i.e., instructional aspects, that drive gaming. In contrast, our findings suggest that student features, namely the average percentage of gaming by a student over all the problems s/he solved, was a stronger predictor of gaming. There are a number of possibilities as to the cause of the difference between our findings and those in [4]. First, the Andes system might have less instructional variability than the one in [4]. Second, [4] used *lesson* as the grain-size, while we used *problem*, a smaller grain size. We did not use *lesson* since as already described in Section 2 we felt such a large grain size may obscure the results. Third, we used data from college students working at home while data in [4] came from high school students working in classrooms. When we recently did a preliminary analysis on a set of high school honours students using Andes mostly in their classroom, we found that their gaming levels were lower than those of college students; thus it is possible that gaming behaviors differ between these two populations and contexts, something that warrants further analysis and validation. A fourth possibility pertains to the way the analysis was done. Baker et al. [4] considered lesson features (e.g., does a lesson have many problems that use the same number for different quantities), and determined how much gaming variance was associated with each feature. Similarly, Baker et al. [15] determined the variance explained by features of students. Our analysis did not use problem features or student features, but rather individual problems and individual students. Our logic was that if there was something "wrong" with a problem, then almost all students should game on that problem; similarly, if there was something "wrong" with a student, then that student should game on almost all problems. In general, this discrepancy in findings in terms of whether problem or student features best predict gaming highlights the need for more work and validation of the factors influencing gaming.

In addition to exploring predictors of gaming, we also analyzed the impact of individual differences on how students were gaming. We found that when we looked at gaming opportunities over the tutor-student turn pairs, students tended to seize the opportunity to game after the tutor presented them with a high level hint. However, when we analyzed the *proportion* of each type of gaming over the total gaming events for each class of gamers, in contrast to high-gamers (who primarily skipped hints), the low gamers had a higher incidence of guessing on entries. One possible explanation for this difference, supported by literature on individual differences in help seeking behaviors [22], is that the low gamers preferred to obtain the solution on their own, without the tutor's help. Another possibility relates to Andes' scoring system. Recall

that students were penalized for asking for a bottom out hint but were not penalized for guessing, and so perhaps the low gamers were simply more concerned about their Andes score than high gamers. Our analysis also showed, however, that low gamers spent more time with hints and took longer to generate a solution entry after seeing a hint. Since no points were awarded for taking time, this suggests that obtaining a higher score was not the only incentive for the high-gamers, indicating that perhaps these students were motivated and/or diligent in the problem-solving process. Jointly, these findings point to the need to tailor gaming interventions to student characteristics in ITS design.

Prior research suggests that poor hint usability leads to gaming [4]. To see if this was the case in our data, we analyzed how students used hints. We found that when the tutor presented a high-level hint, high gamers were quite unlikely to even *try* generating a corresponding solution entry, as compared to low gamers. If students did try to generate a solution entry, both low and high gamers were moderately successful when given a high-level hints. This provides some indication for the utility of these hints, suggesting that their abuse may be driven by other factors. It is possible, however, that students didn't bother to use the high-level hint at all, and were successful because they generated the solution on their own. To have a better understanding of hint utility, one could obtain students' base-rate performance. However, Andes makes hints available on demand, and students sometimes abuse these. This makes it less clear how to determine this base performance, and is something we leave for future work. We also plan to analyze deeper the student (and problem) features that predict gaming frequency, as well as analyze how gaming influences learning outcomes – although preliminary steps have been taken (e.g., [5]), more work is needed.

Acknowledgements. The authors thank the anonymous reviewers for their helpful suggestions. This research was funded the National Science Foundation, including the following grants: (1) IIS/HCC *Affective Learning Companions: Modeling and supporting emotion during learning* (#0705883); (2) *Deeper Modeling via Affective Meta-tutoring* (DRL-0910221) and (3) *Pittsburgh Science of Learning Center* (SBE-0836012).

References

1. VanLehn, K., *Analogy Events: How Examples Are Used During Problem Solving*. Cognitive Science. 22(3): p. 347-388 (1998)
2. Renkl, A., *Learning from Worked-Examples: A Study on Individual Differences*. Cognitive Science. 21(1): p. 1-30 (1997)
3. Chi, M., S.A. Siler, H. Jeong, T. Yamauchi, and R.G. Hausmann, *Learning from Human Tutoring*. Cognitive Science. 25: p. 471–533 (2001)
4. Baker, d.C., J. Raspat, V. Aleven, A.T. Corbett, and K.R. Koedinger. *Educational Software Features That Encourage and Discourage "Gaming the System"*. In: *14th International Conference on Artificial Intelligence in Education*. p. 475-482 (2009).

5. Baker, R.S., A. Corbett, K. Koedinger, and Wagner. *Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game the System"*. In: *ACM CHI 2004: Computer-Human Interaction*. p. 383-390. (2004).
6. Baker, R., A. Corbett, I. Roll, and K. Koedinger, *Developing a Generalizable Detector of When Students Game the System*. User Modeling and User-Adapted Interaction. 18(3): p. 287-314 (2008)
7. Baker, R., A. Corbett, K. Koedinger, and I. Roll. *Generalizing Detection of Gaming the System across a Tutoring Curriculum*. In: *11'th International Conference on Intelligent Tutoring Systems*. p. 402-411 (2006).
8. Rodrigo, M., R. Baker, S. d'Mello, M. Gonzalez, M. Lagud, S. Lim, A. Macapanpan, S. Pascua, J. Santillano, J. Sugay, S. Tep, and N. Viehland. *Comparing Learners' Affect While Using an Intelligent Tutoring Systems and a Simulation Problem Solving Game*. In: *9th International Conference on Intelligent Tutoring Systems,* p. 40-49 (2008).
9. Baker, R., J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger, *Why Students Engage In "Gaming the System"*. Journal of Interactive Learning Research. 19(2): p. 185-224 (2008)
10. Murray, R.C. and K. VanLehn. *Effects of Dissuading Unnecessary Help Requests While Providing Proactive Help*. In: *Artificial Intelligence in Education*. p. 887-889 (2005).
11. Walonoski, J. and N. Heffernan. *Prevention of Off-Task Gaming Behavior in Intelligent Tutoring Systems*. In: *International Conference on Intelligent Tutoring Systems (ITS'06)*. p. 722-724 (2006).
12. VanLehn, K., C. Lynch, K. Schulze, J. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill, *The Andes Physics Tutoring System: Lessons Learned*. International Journal of Artificial Intelligence and Education. 15(3): p. 1-47 (2005)
13. Baker, R., A. Corbett, and K. Koedinger. *Detecting Student Misuse of Intelligent Tutoring Systems*. In: *International Conference on Intelligent Tutoring Systems*. p. 531-540. (2004).
14. Walonoski, J. and N. Heffernan. *Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems*. In: *Intelligent Tutoring Systems*. p. 382-391 (2006).
15. Baker, R. *Is Gaming the System State-or-Trait? Educational Data Mining through the Multi-Contextual Application of a Validated Behavioral Model*. In: *Workshop on Data Mining for User Modeling*. p. 76-80 (2007).
16. Baker, R.S., I. Roll, A. Corbett, and K. Koedinger. *Do Performance Goals Lead Students to Game the System*. In: *International Conference on Artificial Intelligence and Education (AIED2005)*. p. 57-64 (2005).
17. Baker, R., A. Corbett, K. Koedinger, E. Evenson, I. Roll, A. Wagner, M. Naim, J. Raspat, D. Baker, and J. Beck. *Adapting to When Students Game an Intelligent Tutoring System*. In: *8th International Conference on Intelligent Tutoring Systems*. p. 392-401 (2006).
18. Aleven, V., *Helping Students to Become Better Help Seekers: Towards Supporting Metacognition in a Cognitive Tutor*., in *Paper presented at German-USA Early Career Research Exchange Program: Research on Learning Technologies and Technology-Supported Education*. (2001)
19. Harris, A., V. Bonnett, R. Luckin, N. Yuill, and K. Avramides. *Scaffolding Effective Help-Seeking Behaviour in Mastery and Performance Oriented Learners*. In: *Artificial Intelligence in Education*. p. 425-432 (2009)
20. Shih, B., K. Koedinger, and R. Scheines, *A Response Time Model for Bottom-out Hints as Worked Examples*, In: *International Conference on Educational Data Mining*. p. 117-126 (2008)
21. Beck, J., K. Chang, J. Mostow, and A. Corbett, *Does Help Help? Introducing the Bayesian Evaluation and Assessment Methodology*, In: *International Conference on Intelligent Tutoring Systems (ITS'08)*. p. 383-394 (2008)
22. Gall, S.N.-L., *Help-Seeking Behavior in Learning* Review of Research in Education. 12: p. 55-90 (1985)