Dissertation Proposal

Designing Intelligent Tutors That Adapt to Student Misuse

Ryan Shaun Baker
Human-Computer Interaction
School of Computer Science
Carnegie Mellon University

Committee:

Albert T. Corbett (Co-Chair)	Human-Computer Interaction
Kenneth R. Koedinger (Co-Chair)	Human-Computer Interaction, Psychology
Tom M. Mitchell	CALD, Human-Computer Interaction
Shelley Evenson	School of Design (External)

Introduction

In recent years, there has been increasing interest in designing systems that respond appropriately to a user's motivational and emotional state (cf. Picard 1997, Cavaluzzi, De Carolis, Carofiglio, and Grassano 2003), especially within the intelligent tutoring community (de Vicente and Pain 2002; del Soldato and du Boulay 1995) . In recent years, intelligent tutoring systems have become increasingly effective at assessing what skills a student possesses and tailoring the choice of exercises to a student's skills (Corbett and Anderson 1995; Martin and vanLehn 1995). This responsiveness to students' cognitive needs has not been matched by equal responsiveness to students' emotional needs. Although it has been observed that students in intelligent tutor-based classrooms are more motivated than students in traditional classrooms (Schofield 1995), some students misuse intelligent tutoring software in a way that suggests less than ideal motivation (Aleven 2001; Mostow et al, 2002). These students engage in a type of behavior termed "gaming the system", attempting to perform well in an educational task by systematically taking advantage of properties and regularities in the system used to complete that task, rather than by thinking about the material. This type of behavior is not just a problem for intelligent tutoring software – it has been documented both within traditional computer-aided instruction, even when that computer-aided instruction is on a topic with potentially life-threatening consequences[1], and within educational games/edutainment designed with the explicit goal of increased motivation and engagement (Klawe 1998, Miller, Lehman, and Koedinger 1999).

In a recent study (Baker, Corbett, Koedinger, and Wagner 2004), I investigated the prevalence and effects of gaming the system on students' learning in intelligent tutor classrooms. Students in this study were observed engaging in two types of gaming the system: help abuse and systematic trial-and-error. I investigated these phenomena by observing students for two class periods (along with my co-author in that paper, Angela Wagner) as the students used a tutor lesson on scatterplot generation. Each student's behavior was observed several times during the course of each class period, in a specific order determined before the class began (as in Lloyd and Loper 1986). In each observation, each student's behavior was coded as being one of the following categories: working in the tutor, talking on-task, talking off-task, silently off-task (for instance, surfing the web), inactive (for instance, asleep), and gaming the system. In this study, I found that a student's frequency of gaming was strongly negatively correlated with learning, but was *not* correlated with other off-task behavior; nor was other off-task behavior significantly correlated with learning, suggesting that not all types of unmotivated behavior are equivalent in their effects on student learning with ITSs. The evidence from this study was neutral as to whether gaming was harmful in and of itself (by hampering the learning of the specific skills gamed) or whether it was merely symptomatic of non-learning goals.

In this thesis proposal, I will outline an approach that will lead to the development of intelligent tutoring systems which respond appropriately when students attempt to game the system. My approach towards this problem will have three prongs of investigation, which can be researched separately, but which will ultimately be most effectively investigated in concert. Each prong corresponds to one of the three disciplines

[1] For instance, one user was found to have engaged in this type of behavior while using computer-aided instruction teaching people how to use laser control systems, where permanent blindness is a consequence of misusing the actual system (personal communication, Samuel Baker)

that underpin Human-Computer Interaction. This is not coincidental; ultimately, this thesis is about designing systems that guide their users to use them properly – a goal that requires a combination of diagnosis through Machine Learning, and insight into humans from Cognitive and Personality Psychology, culminating in Design that changes the human-computer interaction in a desirable fashion.

Prong I: Understanding Why Students Game the System (PSYCHOLOGY)
Prong II: Detecting When a Student is Gaming the System (COMP SCI)
Prong III: Remediating Gaming the System (DESIGN)

Each of the three prongs will inform and support each other, as is shown in Figure 1. For instance, understanding why students game the system will be necessary for effective remediation; at the same time, finding out what remediations are effective will also expand our knowledge of why students game the system. Developing a rich model (using machine learning) that can detect when a student is gaming will be necessary for assigning interventions appropriately; doing so will also give us insight into when and why students game the system.

This thesis can be expected to produce relevant and widely applicable contributions in each of its three prongs. Firstly, by using machine-learning techniques to learn an optimal psychometric model, this thesis will show how to automatically create a computational model that can be used both to *assign* interventions and to *design* interventions. This model will simultaneously fulfill the goals of effective detection and informing human understanding, goals often treated as orthogonal, but which frequently co-occur in the design of adaptive systems. Secondly, creating better understanding of how and why students misuse learning opportunities in intelligent tutoring systems will increase general understanding of how students' emotions and motivations affect their use of educational technology, and understanding of how students orient themselves towards learning opportunities in general. Finally, developing interventions to appropriately respond to when a student is gaming the system will provide a case study in motivationally appropriate design and in how intelligent systems can respond to being used inappropriately.

The final deliverable of this thesis will be an intelligent tutoring system that can respond appropriately when a student is trying to game it, guiding the student to more appropriate and educationally effective use. Although we will only implement this tutorial approach within the context of a single tutor lesson, my detection and remediation approach will be designed in such a fashion that it can be naturally and quickly extended to other intelligent tutoring systems for mathematics.

In the next three sections, I will discuss each of the three prongs of this thesis in detail. Within each prong, I will discuss relevant prior work and prior hypotheses; I will then discuss my work on this prong thus far, and I will offer a plan of action that will lead to completion of this prong of the thesis.

Finally, I will propose a schedule for completing the thesis, and offer some conclusions.

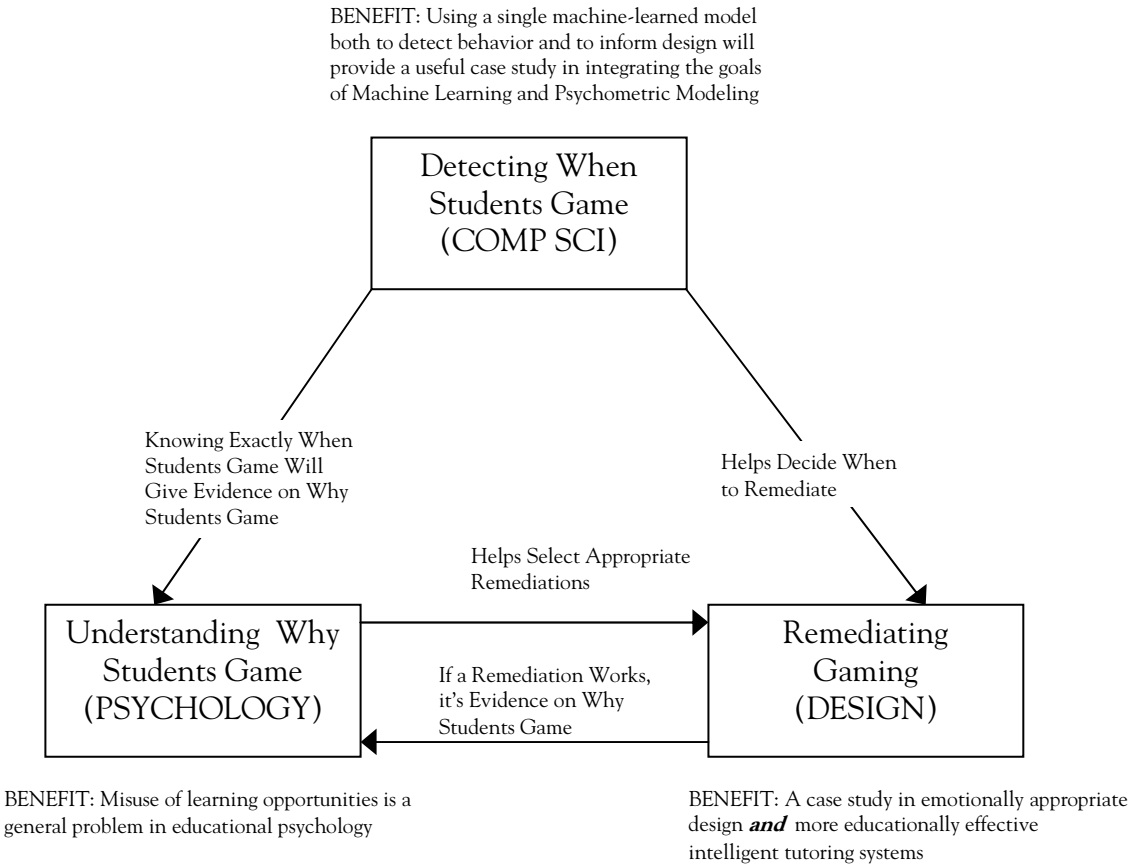

FIGURE 1: The Three Prongs of this Thesis

Prong I: Understanding Why Students Game The System

Hypotheses/ Relevant Work

Determining why students game the system will be essential to developing effective remediation for this behavior. It may be the case that different students choose to game the system for different reasons – in this case, it will be worthwhile to investigate whether the effects of gaming the system are the same for these different students.

A survey of the relevant literature gives four hypotheses for why students game the system. In my thesis, I will investigate to what degree these different motivations lead to gaming. This knowledge will inform the design of interventions in Prong III, and the success or failure of different interventions will re-inform our understanding of why students game the system.

Hypothesis 1 "LACK OF INTEREST":

One possible hypothesis for why students misuse intelligent tutoring software is that the students are insufficiently interested in the intelligent tutor and the material it presents. Lack of interest is a common problem in primary and secondary school classrooms (Lepper and Greene 1978, King-Stoops and Meier 1978, Keller 1987, Winter 1991), including intelligent tutor classes (Arroyo 2000). In this hypothesis, a student chooses to misuse the intelligent tutors in order to amuse him or herself, because he or she is not particularly interested in learning the material the tutor teaches.

Hypothesis 2 "PERFORMANCE GOALS":

A second potential reason why students game the system is because they have the goal of performing successfully -- completing problems, impressing their teacher, and earning a good grade -- instead of having the goal of learning the concepts discussed in the tutor. This type of motivation is referred to both as having performance goals and also as extrinsic motivation (Harter 1981, Elliott and Dweck 1988; McNeil and Alibali 2000). According to this hypothesis, students realize that they can complete problems more quickly by gaming the system than by thinking through each problem step. Gaming the system creates the appearance that the student is doing well, which may important both for impressing the teacher, and for competing with other students to see who can complete problems most quickly (cf. Schofield 1995).

Hypothesis 3 "FRUSTRATION/LEARNED HELPLESSNESS":

Another potential reason why students misuse intelligent tutoring systems is because they are frustrated with the material/with their inability to understand the material[2]. Instead of trying to figure out the difficult material, a frustrated student may attempt to progress through the tutor problem in a way that avoids having to think through the mathematics. Doing so prevents the experience (and feeling) of failure. It has been shown that students who are having difficulty with material often respond with withdrawal of effort, termed "learned helplessness" in the educational psychology literature (Dweck 1975, Keller 1983, Steinsmelter-Pelster and Schurmann 1990) It has been noted that performance goals and learned helplessness are often correlated with each other (Elliott and Dweck 1988); if both of these factors appear to explain a student's choice to game the system, a remediation which takes both factors into account may be especially effective. Generalized anxiety about computers may also contribute to students' feelings of helplessness (Smith and Caputi 2001).

Hypothesis 4 "LACK OF METACOGNITIVE KNOWLEDGE":

A fourth possible reason why students misuse intelligent tutoring systems is because they don't fully understand how to use intelligent tutors properly, or lack appropriate knowledge on how to use the tutoring software properly. According to this hypothesis, a student clicks rapidly through help to get the answer because they don't understand that reading the help messages will be beneficial. This hypothesis is suggested in Aleven (2001), where a model of appropriate help-seeking behavior is presented, and a Metacognitive Tutor is proposed to teach

[2] Note that this is distinct from being frustrated with the computer or tutor, because it is difficult to use or does not work correctly.

students proper help-seeking behavior, using the same methods other tutors use to teach cognitive skill.

This hypothesis contrasts strongly with the previous three hypotheses, which suggest that students game the system intentionally, and are quite intentionally not attempting to learn.

Results So Far

Thus far, we have conducted two studies on gaming the system that will give evidence on the role that these hypotheses might play in why students choose to game the system. The first study has been analyzed thoroughly, and is discussed in more detail in (Baker, Corbett, Koedinger, and Wagner 2004); the second study was completed in the first week of April 2004, and the data is currently being coded. Given that we have not completed analysis of Study 2, we will discuss Study 2 primarily in the "Plan of Action" section. In this section, we will discuss Study 1.

Study 1 Design

Study 1 was conducted in a set of 5 middle-school classrooms at 2 schools in the Pittsburgh suburbs. Student ages ranged from approximately 12 to 14. The classrooms studied were taking part in the development of a new 3-year cognitive tutor curriculum for middle school mathematics. Seventy students were present for all phases of the study. The classrooms were studied during the course of a short (2 class period) cognitive tutor unit on scatterplot generation and interpretation – this unit is discussed in detail in (Baker, Corbett, Koedinger, and Schneider 2004). Each student took a pre-test after viewing a PowerPoint presentation giving conceptual instruction, but before beginning use of the tutor lesson; each student took a post-test after the tutor lesson had concluded. The pre-test and post-test were isomorphic to each other and were counterbalanced between the pre-test and post-test.

We also collected evidence about the pattern of students' behavior during tutor usage. Each student's behavior was observed a number of times during the course of each class period, in a pre-determined order (as in Lloyd and Loper 1986), for 20 seconds. In each observation, the student was coded as on-task working, on-task talking, off-task talking, off-task alone, inactive, or gaming the system.

Finally, we collected each student's end-of-course test scores (which incorporated both multiple-choice and problem-solving exercises) as a measure of general academic achievement and noted each student's gender.

Study 1 Results

First, we determined within Study 1 that the frequency of gaming the system was significantly negatively correlated with learning, $F(1,68)=11.82$, $p<0.01$, $r= -0.38$. This correlation remained significant even when we controlled for the students' prior knowledge of the material (as measured by the pre-test) and their general level of academic achievement, $F(1,59)=7.73$, $p<0.01$, partial correlation = -0.34.

We determined in that study that prior knowledge of the material (as measured by the pre-test) was negatively correlated to the frequency of gaming the system, $F(1,68)=5.31$, $p=0.02$, $r= -0.27$. Every student who was observed gaming the system had

low prior knowledge of the material; on the other hand, many students with low prior knowledge of the material did not game the system. Thus, low prior knowledge of the material appeared to be a necessary but not sufficient condition for predicting which students would game the system. General academic achievement, as measured by the students' scores on their school district's end-of-year tests, was also negatively correlated with gaming the system, but only marginally, $F(1,61)$[3]$=2.77$, $p=0.10$, $r=-0.21$. When we controlled for prior knowledge of the material, students' scores on their end-of-year tests were no longer significantly correlated with the frequency of gaming the system, $F(1,60)=0.22$, $p=0.64$. Gender was also not correlated with the frequency of gaming the system, $F(1,68)=1.02$, $p=0.31$.

Gaming the system was not significantly correlated to the frequency of other off-task behaviors, such as talking off-task to other students, $F(1,68)=0.33$, $p=0.57$. However, there was a significant correlation between the frequency of gaming the system and talking to the teacher or another student about the subject matter, $F(1,68)=10.52$,$p<0.01$, $r=0.37$; this correlation remained significant even when we controlled for prior knowledge and general academic achievement, $F(1,59) = 8.90$, $p<0.01$, partial correlation $= 0.36$.

Further machine-learning based analysis of the data set from Study 1 (discussed in greater detail in the Prong II section) produced some surprises. First, students who game the system appear to fracture into two distinct groups. Students who game the system and show poor learning are easily identified by a machine-learning algorithm trained on both groups of students, ($A' =0.82$, 95% Confidence Interval(A') = 0.63-1.00, chance $A' =0.50$). The same recognizer is no better than chance at identifying the students who game the system but still do well on the post-test, ($A' =0.57$, 95% CI(A')=0.35-0.79. One possible explanation is that different students may game the system for different reasons, and that why a student chooses to game the system may affect the degree to which gaming the system harms them. An alternate explanation is that our machine-learning analysis has exposed some false positives in our study 1 observations. It will be interesting to investigate this question in greater detail with our study 2 results.

The algorithm's predictions about *when* the students gamed the system also differed between these two groups of students. Students who gamed the system and showed poor learning gamed (according to the algorithm's predictions) 11.9% of the time on difficult skills but only 1.5% of the time on easy skills, a significant difference, $t(7)=2.99$,$p<0.05$. Interestingly, students who gamed the system but still did well on the post-test (indicating either positive learning or high prior knowledge), were predicted to have gamed 1.8% of the time on difficult skills but 4.2% of the time on easy skills, which was not a significant difference, $t(8)=1.69$, $p=0.13$. However, the predicted difference between these two groups must be taken considered with some skepticism – we are not yet certain that our algorithm is reliable at detecting *when* a student is gaming the system (in study 1, we only obtained data on which students gamed the system, not precisely when they did so). Being certain about this difference between the two groups will require first verifying that we are accurately determining when students game the system – this issue will be discussed further in the Prong II section.

Implications for our Hypotheses

[3] End-of-year test scores could only be obtained for 63 of the 70 students who participated in the study.

The first implication from our data is that there may be more than one reason why students game the system. The fact that students who game the system and do poorly on the post-test are behaviorally distinguishable (according to our machine learning algorithm) from students who game the system but do well on the post-test is quite interesting, and merits further investigation. In general, if gaming the system for one reason is correlated with poor learning whereas gaming the system for another reason is correlated with positive learning, it will be more important to design remediations for the type of gaming correlated with poor learning.

Considering the hypotheses individually, the first hypothesis ("LACK OF INTEREST") was not supported by these findings, but was also not explicitly disconfirmed. We would expect students who were uninterested in the material to engage in non-gaming off-task behavior, such as talking to neighbors or staring into space. If students chose to game the system out of the same motivation, we would expect the frequency of gaming and non-gaming off-task behavior to be correlated with each other; but in fact, they were not. On the other hand, it is conceivable that gaming may substitute for other types of off-task behavior for those students who engage in it; hence, our current data does not explicitly disconfirm the first hypothesis.

The second hypothesis ("PERFORMANCE GOALS") was also not consistent with the students' other off-task behavior. If students are focused on performance (but not learning), we might expect them to talk off-task less frequently –there was no such negative correlation. However, we *would* expect high-gaming students to talk on-task more frequently to the teacher and to other students; and we did observe this pattern of behavior. In fact, Arbreton (1998) has observed there is a connection between performance goals and a specific type of student-teacher interaction termed "executive help-seeking" (Nelson-LeGall and Glor-Scheib 1985, cited in Arbreton 1998), where students request help from their teacher immediately, before attempting to solve the problem at hand. Hence, the evidence for performance goals is mixed but reasonably positive.

The third hypothesis ("FRUSTRATION/LEARNED HELPLESSNESS") is supported by our data. The learned helplessness hypothesis would not predict more or less off-task conversation, in line with our observed results. Additionally, to the extent that students talking about the tutor may be engaging in "executive help-seeking", learned helplessness may also explain the correlation between gaming the system and on-task conversation. Thirdly, the negative correlation between prior knowledge of the material and gaming the system is definitely consistent with learned helplessness; we would expect the students who know the least in advance to find the material most frustrating and difficult.

One intriguing aspect of the machine learning algorithm's split of gaming students into two groups is that students who game and do poorly on the post-test seem to game far more frequently on difficult skills, which would be consistent with learned helplessness; the students who game but still do well on the post-test may have a trend towards gaming more often on the *easy* skills, which would be more consistent with a lack of interest (hypothesis 1), perhaps stemming from insufficient challenge. It will be interesting to investigate this question further, in the light of the data from study 2.

As with the first hypothesis, our data gives negative evidence about the fourth hypothesis ("LACK OF METACOGNITIVE KNOWLEDGE") but comes nowhere near disconfirming it. If a student had low domain knowledge and frequently talked to the teacher and other students (about the tutor), we might actually think this student has *high* metacognitive knowledge, since the student is aware he or she does not know how to answer the questions and needs help. Thus, such behavior could be interpreted as

evidence that students who frequently game the system actually have high metacognitive awareness (combined, perhaps, with performance goals), directly contradicting the fourth hypothesis. In order to view this evidence as solid disconfirmation of the fourth hypothesis, however, we would need to better understand *why* the students were talking to the teacher or other students. Another factor that seems to suggest that lack of metacognitive knowledge is not a complete explanation for why students choose to game the system is that students attempt to hide their gaming from the teacher and the researchers, suggesting that they do – at some level – know it's not the correct way to use the tutor.

Plan of Action

Study 2

As mentioned earlier, we have just completed a second study on gaming the system. This study enabled us to collect a substantially richer data set than in study 1, which should give additional evidence about what motivations underlie a student's behavior when he or she is gaming the system.

In this study, we used the same observational technique to assess the frequency of each student's gaming – with the exception that we divided on-task conversation into two categories – conceptual discussions and discussions about answers. We did this to see if students who game the system also attempt to solicit answers from other students (cf. Arbreton 1998).

We also wrote down all student outbursts (defined as utterances that could be heard from across the room) relevant to the tutor or materials. Coding these utterances should allow us to see whether outbursts indicating frustration or boredom were correlated with gaming the system.

Unlike study 1, which took place during only one tutor lesson, we analyzed gaming across the duration of three tutor lessons – in one school, lessons on scatterplots, 3D geometry, and permutations; in another school, lessons on scatterplots, 3D geometry, and probability. This allowed us to more closely investigate the relationship between prior knowledge and gaming the system. A pre-test was given at the beginning of the study, with questions about each lesson; a post-test was given at the conclusion of the study.

The pre-test and post-test also contained items drawn from and adapted from previous motivational assessments (cf. Cohen and Waugh 1989; Mueller and Dweck 1998; Harnisch, Hill, and Fyans 1980; Sarason 1978). These items may provide evidence for or against the four hypotheses – and evidence for whether students who game the system and learn have different motivational profiles from students who game the system and do not learn (as well as different motivational profiles from students who do not game).

In order to further investigate the relationship between gaming and classroom incentive (and to get causal evidence of gaming's impact on learning), we manipulated the reward for doing well on the post-test. We paired classes together based on school and student population (ie, two regular classes in school A, two "modes"[4] classes in school A, and two regular classes in school B) and then assigned one class from each pair to do the 3D-Geometry lesson second and the scatterplot lesson third, and the other class to do these lessons in the opposite order. All students received a reward manipulation during the third lesson, preventing confounds between reward condition and mathematical

[4] Below average intelligence or with behavior problems; but not special education.

topic. We did not control for ordering/reward confounds, under the hypothesis that if a student received a reward manipulation during the second lesson, it might contaminate their behavior during the third lesson.

The reward offered was a Krispy Kreme donut for high performance on the third lesson's post-test. The teacher was instructed to tell the students

"Okay, everyone, listen up. The people at CMU want to thank you for helping out with their study. We're going to have a new lesson today and next Tuesday. Anyone who does really well on the post-test for this lesson is going to get a donut from Krispy Kreme. So work hard, and learn a lot, so you can do well on the post-test."

This reward appeared to be a strong motivator for the students – several students immediately cheered and students were overheard discussing Krispy Kreme donuts in multiple classes, and bragging that they were going to do well and get donuts. Two weeks after the study was completed, the experimenter was still referred to by students in one class as "The Krispy Kreme Guy".

The goal of this manipulation was to gain additional information on the hypotheses for why students game. Specifically, if students are primarily motivated to game out of lack of interest (H1) or have performance goals (H2), the possibility of a reward is likely to induce them to game less (because they are *interested* in donuts, and want to *perform* in a fashion that gets them donuts). It has been repeatedly noted that rewards increase performance on moderately difficult and easy tasks – see Allscheid and Cellar (1996) for a review of the relevant literature – although such rewards can also decrease long-term interest and performance on that task in the future, especially when the task is already of high interest (Lepper and Greene 1978). On the other hand, if students don't realize that gaming harms learning (H4), they might be expected to talk less off-task, but to game the same amount. Most interestingly, if students game out of learned helplessness (H3), the increased reward should increase anxiety and self-focus, and paradoxically increase the frequency of gaming (cf. Baumeister 1984).

Beyond Study 2

Given the data from study 2, we will develop a set of interventions in Prong III, mapping to the most likely hypotheses. Seeing which interventions are effective for reducing the incidence of gaming behavior in students with different motivational profiles will provide further evidence for why students choose to game the system; for instance, if we predict that a student is gaming the system based on learned helplessness, then interventions targeted to that hypothesis should be effective, and others should not. If the opposite appears to be true, we may have to reconsider our hypotheses. Being able to draw such conclusions will depend on designing interventions that make clear contrasts between our hypotheses.

Additionally, if our evidence from study 2 is inconclusive, it may be useful to interview students who frequently game before we attempt to create interventions. One way to create appropriate context for the student, so that they would understand what we were asking would be to conduct a retrospective think-aloud (Nielsen 1993), showing the student a replay of their tutor usage, and asking them to explain what they were thinking at that point in the tutor. Alternatively, if students seemed to be uncomfortable discussing their own actions, it might be possible to show a student another student's pattern of behavior (including gaming), and ask them to speculate on what that other student might have been thinking.

Prong II: Detecting When A Student is Gaming the System

Goals

Regardless of what intervention we choose to use to remediate gaming, it is likely this intervention will have some negative side-effects. I will attempt to design interventions that "fail-soft", creating minimal confusion or problems if inappropriately delivered. Nonetheless, at minimum any remediation will take time – and may still be annoying or frustrating if given to the wrong students.

The potential negative side-effects can be mitigated by delivering our intervention, so far as is possible, only to those students who game the system in a fashion that reduces their learning. Hence, it is our goal to develop an algorithm which can detect whether -- and when -- a student is gaming the system. The better its ability to correctly categorize gaming students while avoiding false positives, the more useful it will be for assigning interventions.

Beyond choosing *who* to give interventions to, and *when* to give those interventions, developing such an algorithm may have a secondary benefit – expanding our knowledge about *how* and *why* students game the system. This benefit will depend on our algorithm being inspectable: that is to say, it should be possible for a human to inspect and understand the factors which lead our algorithm to decide a student is gaming. It should also be possible for a human to inspect and understand the factors which, according to our algorithm, appear to be correlated with gaming or which regularly occur just prior to gaming.

These two goals, highly accurate detection and easy inspectability, will drive our efforts to develop a model of how and when students game the system.

Relevant Work

There are three broad bodies of work that relate to our goal of developing inspectable models that can detect whether and when a student is gaming: machine learning, computational cognitive modeling, and statistical modeling. Verbal models such as ARCS (Keller 1983), can also inform design, but are not useful for detection.

Machine Learning

Machine Learning is frequently used to develop models that are successful at classification and detection, in a wide variety of domains. In domains related to gaming the system, for instance, machine learning models have been successful at detecting students' learning style preferences (ie visual versus verbal) (Castillo, Gama, and Breda 2003), determining whether students are in need of assistance (Beck, Jia, Sison, and Mostow 2003; Luckin and du Boulay 1999), detecting whether a student is frustrated (Heiner, Mostow, and Beck 2003), determining what types of intrinsic motivation a student has (deVicente and Pain 2002), and determining whether students are

exhaustively seeking information even after they have sufficient information to answer a question (Vendlinski and Stevens 2003).

One significant advantage that machine learning brings to bear on detection is its ability to automatically discover relationships which may not have even occurred to the researcher using the algorithm. Especially for highly complex phenomena , or phenomena where small changes in the magnitudes of the input can produce qualitatively different results, automatic discovery often makes it possible to discover relationships which are prohibitively difficult to discover on one's own.

However, many highly-effective machine learning models are highly difficult to inspect and understand. For instance, decision trees for real-world phenomena often have hundreds of if-then branchings, support vector machines require figuring out multi-dimensional spatial relationships, and neural networks require tracing through a large number of different "neurons" to figure out the relationships that produce the given classification. One type of machine learning of machine learning which produces reasonably inspectable models is learning if-then rules with genetic algorithms (Romero, Ventura, de Bra, and de Castro 2003) – however, it is not yet clear how broadly applicable this method is.

Psychometric Modeling

Statistical Models are systems of equations that can be used to characterize a phenomena. Systems of equations underlie many of the techniques currently popular in machine learning, such as support vector machines – however, the mathematical equations and forms that describe machine learning models have deviated considerably from those used in mathematical psychology. What I refer to here as psychometric models are the quantitative statistical modeling approaches that are in standard use within mathematical psychology – general linear regression models, hierarchical linear models, Rasch models, latent response models, and so on. Such models are highly comprehensible by individuals with experience in mathematical modeling, and are therefore highly inspectable. Additionally, conducting statistical analyses of these mathematical models is quite easy and thus these approaches are routinely used by psychological researchers.

For these reasons, psychometric models are frequently used to provide accounts about student learning and cognition. Specifically, psychometric models have been used in many recent accounts of student motivation and motivation-related behavior – for instance, understanding the factors that lead preschoolers to mastery orientation (Turner and Johnson 2003), and understanding the relationship between a student's self-reported learning goals and their reported attempts to process educational materials deeply (Bandalos, Finney, and Geske 2003).

Psychometric and other mathematical models are in many cases highly successful at detection – in a prominent recent case, being used to detect when teachers write answers for their students on standardized exams (Jacob and Levitt in press). Nonetheless, psychometric models are used less frequently for automated detection than machine learning models (with the possible exception of Item-Response Models, which are frequently used within adaptive standardized exams – cf. Rudner, 1998). As psychometric models are developed by human insight, they are limited to capturing a smaller number of potential relationships than current state-of-the-art machine learning models can capture.

Computational Cognitive Modeling

Computational Cognitive Models in frameworks such as ACT-R and SOAR provide very clear and understandable accounts for why observed behavior occurs. In this, they serve much the same purpose as psychometric models – but allow the expression of conceptual ideas in a form that facilitates the creation and comprehension of substantially more complex models than are possible with traditional psychometric approaches. However, as a result of this complexity, there are not currently techniques to automatically discover computational cognitive models, and their development is very labor-intensive. Additionally, there are not currently accepted statistical techniques for comparing running computational cognitive models, although there is ongoing work in this area (cf. Baker, Corbett, and Koedinger 2003). On the other hand, the symbolic character of computational cognitive models, where each parameter of a model is identified one-to-one with constructs that have real-world meaning has a considerable advantage over most machine learning approaches: a reasonably clearly-written computational cognitive model can be understood very quickly by anyone with experience in that modeling formalism. Hence, computational cognitive models tend to be highly inspectable.

Computational cognitive models have been highly successful in modeling student learning, including how students acquire skill (Anderson, 1993) and the manner in which skill transfers between domains (Singley and Anderson, 1981). When combined with Bayesian knowledge-tracing (a psychometric approach implemented in code – Corbett and Anderson, 1995), the same computational cognitive models are quite successful at detecting whether a student has acquired a cognitive skill. In cognitive tutoring systems, these model assessments drive the selection of remedial feedback and what exercises the student is given, resulting in significant improvements in learning (Anderson, Corbett, Koedinger, and Pelletier, 1995).

Computational cognitive models have also been used to model motivation and emotion related phenomena – for instance, modeling the relationship between different levels of pressure to succeed and student learning gains[5] (Belavkin 2001). A model of how distress, interest, and pleasure can affect problem-solving behavior has also been created (Belavkin, Ritter, and Eliman 1999). However, these models provide fairly basic accounts for how motivation affects behavior, compared to most of the theories discussed in the Prong I section. It seems likely that there is a niche for a computational cognitive modeling framework designed to create sophisticated models of emotions and motivation (cf. Simon 1967, Scheutz 2003); but at the current time the lack of such a modeling framework limits the usefulness of computational cognitive models for our current modeling goals. Interestingly, in order to circumvent this limitation, one group of researchers has built hybrid models which model emotion through machine learning techniques and the behavior based on that emotion with computational cognitive models (Jones, Henninger, and Chown 2002).

Another disadvantage of computational cognitive models for our current purpose is that they are not particularly designed for automatic discovery (models exist of how *people* conduct scientific discovery -- cf. Schunn and Anderson 1998; Langley et al 1987 -- but computational cognitive models that actually discover optimal models have not been developed). Best-fit parameters can be found using gradient descent algorithms (Baker, Corbett, and Koedinger 2003), but the structure of the model must be developed by the modeler, a task that requires considerable expertise. Extremely complex phenomena can,

[5] Specifically, the Yerkes-Dodson (1908) law, where high and low pressure both produce lower learning gains than moderate pressure.

and have, been modeled in computational frameworks – but the sophistication of the models created is dependent on the creativity of the model designer.

My Approach

In my thesis, I will argue that a hybrid approach is an excellent way to accomplish the joint goals of highly accurate detection and easy inspectability. Instead of combining two different types of model (cf. Corbett and Anderson 1995, Jones, Henninger, and Chown 2002, Stevens, Soller, Cooper, and Sprang submitted), I propose to combine methods within one model. Specifically, I propose using machine learning to automatically learn a psychometric model. Whereas it is extremely time-consuming and difficult to develop a psychometric model which is sufficiently good at detection by hand, and whereas most traditional machine learning approaches are not acceptably inspectable, this approach will accomplish both goals. Using hybrid approaches of this character may have considerable value to researchers in adaptive systems – the joint goals of detection and increasing understanding (to inform design) are common for designers of adaptive systems.

Algorithms already exist that could easily be used for automatic discovery of relationships within a class of psychometric models – for instance, forward and backward selection (Ramsey and Schafer 1997) – intriguingly, these algorithms are frequently run by hand by researchers, within statistical packages, rather than being automatically conducted.

This type of approach could be used in a principled fashion with just about any type of psychometric model; in this paper, I will use a variant on Latent-Response Models (LRM) (Maris 1995). This modeling framework is comprehensible, and supports integrating multiple sources of data into one model. It facilitates relating each individual actions each student makes to the overall pattern of gaming, across students, observed in our data. Finally, such a model can be extended to other contexts (ie different tutor lessons) with standard statistical techniques such as data imputation (Fichman and Cummings 2003).

Results So Far

We have developed an algorithm that, within the original data set from study 1, can reliably detect which students frequently game the system and which student do not. It can successfully make this distinction after cross-validation, suggesting that it will generalize appropriately to students similar to those in study 1, who use the same tutor lesson under similar conditions. This algorithm is also reasonably inspectable, giving insight into when students game the system.

<u>Data Sources</u>

In order to develop an algorithm to detect when a student was gaming the system, we combined three sources of data on student performance and behavior from our cognitive tutor lesson teaching about scatterplot generation (Baker, Corbett, Koedinger, and Schneider 2004). All data was drawn from a group of 70 students using that cognitive tutor lesson as part of their normal mathematics curricula.

The first source of data was the log files of every action each student performed while using the tutor. For each action, we distilled 24 features from the log files. The features included such aspects as the tutoring software's assessment of the action (correct,

incorrect, a "bug" – cf. Anderson et al 1995, vanLehn 1990, or a help request) and how likely it was that the student knew the underlying skill, the type of interface widget involved in the action, the number of attempts on this problem step, the length of time taken on this problem step, and so on. A fuller discussion of the features used can be found in (Baker, Corbett, and Koedinger submitted)

The second source of data was the set of human-coded observations of student behavior during the lesson. From the observed frequencies of different behaviors in each student, we determined the approximate proportion of time each student spent gaming the system, $G_0 \ldots G_{69}$.

Since it is not clear that all students game the system for the same reasons or in exactly the same fashion, we used student learning outcomes as a third source of data, to help us determine which students to train and test our classifier on. We divided students into three sets: a set of 53 students who were never observed gaming the system, a set of 9 students who were observed gaming the system but who either had a high pretest score or a high pretest-posttest gain and thus were not obviously hurt by their gaming behavior (this group will be referred to as GAMED-NOT-HURT), and a set of 8 students who were observed gaming the system and had a low score on their post-test (referred to as GAMED-HURT). It is important to distinguish GAMED-HURT from GAMED-NOT-HURT students during analysis, since these two groups may behave differently (even if an observer sees their actions as similar), and it is substantially more important to target interventions to the GAMED-HURT group than the GAMED-NOT-HURT group. This sort of distinction has been found particularly effective in developing algorithms to differentiate cheating from other categories of behavior (Jacob and Levitt in press).

Data Modeling

Using these data sources, we trained a density estimator to predict how frequently an arbitrary student was gaming the system. The algorithm we chose was forward-selection (Ramsey and Schafer 1997) on a set of mathematical models that can be described as Latent Response Models (LRM) (Maris 1995). This type of model has two distinct advantages for our purposes. First, using a hierarchical modeling framework (such as LRMs) makes it very easy and natural to integrate multiple sources of data into one model. Additionally, LRMs can be interpreted much more easily by humans than the results of more traditional machine learning algorithms such as neural networks, support vector machines, or even most decision tree algorithms. The mathematical equations generated by our algorithm can be interpreted with reasonable effort, facilitating thought about design implications – we will discuss how in further detail in the next section. Finally, it can be used to develop a classifier via setting thresholds.

Traditional LRMs, as characterized in Maris, are composed of two levels: our model is composed of three. In the outermost layer of an LRM, the LRM's results are compared to observable data. In the outermost layer of our model, our model makes a prediction about how frequently each student is gaming the system, labeled $G'_0 \ldots G'_{69}$. The model's prediction for each student is compared to the observed proportions of time each student spent gaming the system, $G_0 \ldots G_{69}$.

In a traditional LRM, each observed prediction is derived by composing a set of predictions on unobservable latent variables – for example, by adding or multiplying the values of the latent variables together. Similarly, in our model, the model's prediction of the proportion of time each student spends gaming is composed as follows: First, the model makes a (binary) prediction as to whether each individual student action (denoted P'_m) is an instance of gaming – a "latent" prediction which cannot be directly validated

within our current data set. From these predictions, $G'_0 \ldots G'_{69}$ are derived by taking the percentage of actions which are predicted to be instances of gaming, for each student.

In a traditional LRM, there is only one level of latent predictions. In our model, the prediction about each action P_m is developed by means of a linear combination of the characteristics of each action. Each action is described in terms of a set of parameters drawn from linear effects on the 24 features of each action found in our log files (parameter*feature), quadratic effects on those 24 features (parameter*feature^2), and 23x24 interaction effects between features (parameter* $feature_A$*$feature_B$). Thus, a prediction P_m as to whether action m was an instance of gaming the system was computed as $P_m = \alpha_0 X_0 + \alpha_1 X_1 + \alpha_2 X_2 + \ldots + \alpha_n X_n$, where α_i is a parameter value and X_i is the data value for the corresponding feature, for this action, in the log files. Each prediction P_m was then thresholded using a step function[6], such that if $P_m \leq 0.5$, $P'_m = 0$, otherwise $P'_m = 1$. This gave us a set of classifications P'_m for each action within the tutor, which was used to create the predictions of each student's proportion of gaming, $G'_0 \ldots G'_{69}$.

Our modeling approach started with a model with no variables. At each forward step in the model selection process, the potential parameter was added that most reduced the mean absolute deviation between our model predictions $G'_0 \ldots G'_{69}$ and the original data $G_0 \ldots G_{69}$, using iterative gradient descent to find the best parameter value for each candidate parameter. Forward-selection continued until no parameter could be found which appreciably reduced the mean absolute deviation. In practice, no model with more than six parameters was ever considered.

Rather than finding the best model for the entire data set, we conducted Leave One Out Cross Validation (LOOCV) (cf. Moore 2004) to get a measure of the model's generalizability. The results of cross-validation give us a metric for assessing how effectively the model will generalize to students who were not in the original data set, although they do not necessarily suggest how well the model will generalize to different tutor lessons (which will be an area of further investigation in this thesis). In doing a LOOCV, we fit to sets of 69 of the 70 students, and then investigated how good the model was at making predictions about the 70th student.

Developing A Classifier

Given our model, I developed a classifier to identify which students are gaming and in need of an intervention. Since most potential interventions will have side-effects and costs (in terms of time, if nothing else), it is important both that the classifier is good at correctly identifying the GAMED-HURT students who are gaming and not learning, and that it rarely assigns an intervention to the other students (especially those who do not game).

I developed a classifier from our model by setting a threshold on how often the model perceives a student is gaming. Any students above this threshold are considered gaming, and all other students are not considered gaming, for the purpose of assigning interventions. Given different possible thresholds, there is a tradeoff between correctly identifying gaming students (hits) and incorrectly identifying non-gaming students as gaming students (false positives). This tradeoff is shown in the Receiver Operating Characteristic (ROC) curve in Figure 1. The classifier's overall ability to distinguish gaming is assessed with an A' value, which gives the probability that if the model is given one gaming student and one non-gaming student, it will accurately identify which is

[6] It has been recommended that I switch from using a step function to using a logistic function, in order to facilitate future statistical analysis (personal communication, Brian Junker); I intend to do so.

which[7] (Hanley and McNeil 1982). All discussions are with reference to the cross-validated version of our model/classifier, in order to assess how well our approach will generalize to the population in general, rather than just our sample of 70 students.

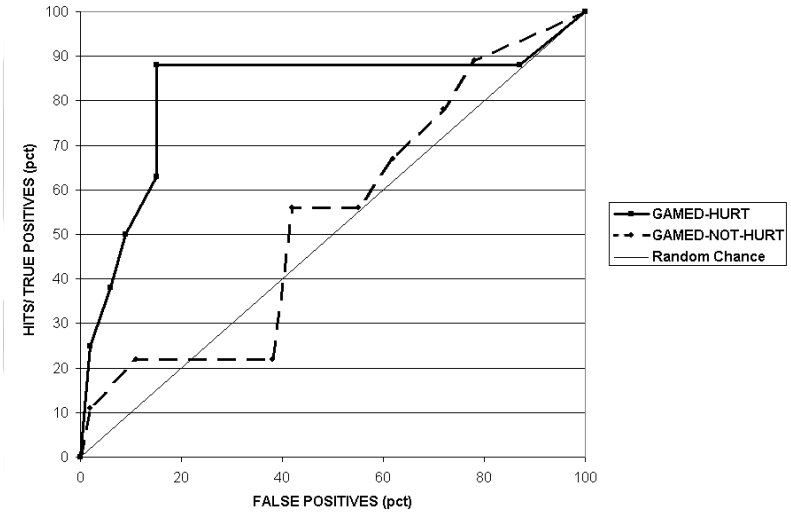

FIGURE 1: The BOTH model's ratio between true positives and false positives, at varying levels of sensitivity. (empirical ROC curves shown rather than smoothed Gaussian curves to give a feel for our model's performance within the current data set)

If we take a model trained to treat both GAMED-HURT and GAMED-NOT-HURT students as gaming, it is quite successful at classifying the GAMED-HURT students as gaming (A' =0.82, 95% Confidence Interval(A') = 0.63-1.00, chance A' =0.50). At the best possible threshold value[8], this classifier correctly identifies 88% of the GAMED-HURT students as gaming, while only classifying 15% of the non-gaming students as gaming. Hence, this model can be reliably used to assign interventions to the GAMED-HURT students.

By contrast, the same model is not significantly better than chance at classifying the GAMED-NOT-HURT students as gaming (A' =0.57, 95% CI(A')=0.35-0.79). Even given the best possible threshold value, it can not do better than correctly identifying 56% of the GAMED-NOT-HURT students as gaming, while classifying 36% of the non-gaming students as gaming. One potential explanation is that these two groups are, in fact, behaviorally different.

Since it is more important to detect GAMED-HURT students than GAMED-NOT-HURT students, it is conceivable that there may be extra leverage gained from training a model only on GAMED-HURT students. In practice, however, a cross-validated model trained only on GAMED-HURT students (A' =0.77, 95% CI(A') = 0.57-0.97) does not do better at identifying the GAMED-HURT students than the model trained on all students. Thus, in our further research, we will use the model trained on both groups of students to identify GAMED-HURT students.

[7] This construct is also referred to in other contexts as discriminability, as AUC (q-hat), and as the area under the empirical ROC curve. It is equivalent to W, the Wilcoxon statistic between signal and noise (Hanley and McNeil 1982). It is considered a more robust and atheoretical measure of sensitivity than D' (Donaldson 1993).

[8] ie, the threshold value with the highest ratio between hits and false positives, given a requirement that hits be over 50%

It is important to note that despite the significant negative correlation between a student's frequency of gaming the system and his/her post-test score, both in the original data (r= -0.38, F(1,68)=11.82, p<0.01) and in the cross-validated model (r= -0.26, F(1,68)=4.79, p=0.03), our classifier is not just classifying which students fail to learn. Our model is not better than chance at classifying students with low post-test scores (A' =0.60, 95% CI(A')=0.38-0.82) or students with low learning (low pre-test *and* low post-test) (A' =0.56, 95% CI(A')=0.34-0.78).

Thus, our model is not simply identifying all gaming students, nor is it identifying all students with low learning – it is identifying the students who game *and* have low learning: the GAMED-HURT students.

Model Description

In this section, we will briefly describe the current model trained on both groups of students. This model predicted that a specific action was an instance of gaming when the expression shown in Table 1 was greater than 0.5:

Name	Coefficient	Feature
F_0 "ERROR-NOW,MANY-ERRORS-EACH-PROBLEM"	-0.0375	pknow-direct[9] * (**multiplied by**) number of errors the student has made on this problem step (across all problems)
F_1 "QUICK-ACTIONS-AFTER-ERROR"	+ 0.09375	pknow-direct * (**multiplied by**) time taken, in standard deviations above (+) or below (-) average time taken, across all students, on this problem step (across all problems)
F_2 "MANY--ERRORS-EACH-PROBLEM-POPUP"	+ 0.231	number wrong on this problem step (across all problems), if the problem step uses a popup menu
F_3 "SLIPS-ARE-NOT-GAMING"	- 0.225	the estimated probability that the student knew the skill * (**multiplied by**) how many errors the student had made on the last 5 actions

Table 1: The model's expression for predicting whether a specific action is an instance of gaming.

On the surface, this model is tricky to interpret, but can be more easily understood if one remembers that pknow-direct serves two functions at once[9]. If the current action is the student's first action on the problem step, pknow-direct is the probability that he or she knows the skill. If the current action is not the student's first action, then it is a

[9] Pknow-direct is a serendipitous feature, drawn directly from the tutor log files; it is useful because it captures a fairly sophisticated branching interaction effect. If the current action is the student's first attempt on this problem step, then pknow-direct is equal to the tutor's assessment that the student knew the skill after the problem-step; on the other hand, if the student has already made an attempt on this problem step, then pknow-direct is -1.

negative multiplier on the other data feature. Hence, pknow-direct allows a contrast between a student's first attempt on a skill he/she knows very well and a student's later attempts.

Feature F_0,"ERROR-NOW, MANY-ERRORS-EACH-PROBLEM", identifies a student as more likely to be gaming if the student has already made at least one error on this problem step within this problem, and has also made a large number of errors on this problem step in previous problems – in extreme cases, as many as 10 errors in each past problem. It identifies a student as less likely to be gaming if the student has made a lot of errors on this problem step in the past, but now probably understands it.

Feature F_1,"QUICK-ACTIONS-AFTER-ERROR", identifies a student as more likely to be gaming if he or she has already made at least one error on this problem step within this problem, and is now making extremely quick actions. It identifies a student as less likely to be gaming if he or she has made at least one error on this problem step within this problem, but is working slowly during subsequent actions, or if a student answers quickly on his or her first opportunity (in a given problem step) to use a well-known skill.

Feature F_2,"MANY-ERRORS-EACH-PROBLEM-POPUP", indicates that making many errors across multiple problems is even more indicative of gaming if the problem-step involves a popup menu.

Feature F_3,"SLIPS-ARE-NOT-GAMING", identifies that if a student has a high probability of knowing a skill, the student is less likely to be gaming, even if he or she has made many errors recently. Feature F_3 may serve to counteract the fact that features F_0 and F_1 ignore the probability that the student knows a skill if he or she has already made an error on the current problem step within the current problem.

Although the model discussed above was developed by training on all students, it is highly similar to the 70 models generated during cross-validation. Features F_0, F_2 and F_3 appear in over 97% of the cross-validated models, and feature F_1 appears in 71% of the cross-validated models. No other feature was used in over 10% of the cross-validated models.

One interesting aspect of this model is that none of the features involve student use of help. We believe that this is more a limitation of the tutor log files we used than a limitation of our modeling approach: current research in identifying help abuse relies upon considerable data about the timing of each internal step of a help request (as in Aleven, McLaren, Roll, and Koedinger submitted), data that our log files for this lesson did not contain. Despite this limitation, it is interesting that a model can be developed which accurately detects gaming without specifically detecting help abuse. One possibility is that students who game the system in the ways predicted by our model also game the system in the other fashions observed in our original study.

Other Interesting Model Features

Two other aspects of our model may be relevant to understanding the phenomenon of gaming better. The first aspect is our model's predictions about how gaming actions are distributed across a student's actions. In our original observations of whether students were gaming the system, it was not possible to determine if a student was gaming the system by observing a single mouse click; determining that a student was gaming required observing multiple actions over the 20-second observation window. This same pattern appeared in our model's action-by-action predictions. 49% of our model's 21,520 gaming predictions occurred in clusters where at least 2 of the nearest 4 actions were also instances of gaming. To determine the chance frequency of such clusters, we ran

a Monte Carlo simulation where each of the 70 students' instances of predicted gaming were randomly distributed across that student's 71 to 478 actions. In this simulation, only 5% (SD=1%) of gaming predictions occurred in clusters of 3 or more actions. Hence, our model predicts a substantially higher number of gaming actions occurring in clusters (49%) than one could expect from chance.

Additionally, we noted that gaming the system was correlated to substantially lower learning, but off-task behavior was not significantly correlated to lower learning. One of our two hypotheses for this difference (discussed in Prong I) is that students may be more likely to choose to game the system if they are at a step in the problem-solving process that they find particularly difficult. One way of measuring how difficult a skill is for a student is the tutor's assessment of the probability that the student knows the skill. If this hypothesis were true, we should expect the student to game more frequently when the system estimates a low probability that the student knows the skill. To investigate this hypothesis, we compared the frequency of gaming on "difficult skills", which the tutor estimated the student had under a 20% chance of knowing[10], to the frequency of gaming on "easy skills", which the tutor had estimated the student had over a 90% chance of knowing. The model predicted that students in the GAMED-HURT group gamed 11.9% of the time on "difficult skills", but only 1.5% of the time on "easy skills", confirming our prediction, $t(7)=2.99$, $p<0.05$ for a two-tailed paired t-test. By comparison, the model predicted that students in the GAMED-NOT-HURT group did not game a significantly different amount of the time on "difficult skills" (1.8%), compared to "easy skills" (4.2%), $t(8)=1.69$, $p=0.13$. This result suggests that GAMED-HURT students may learn less than GAMED-NOT-HURT students specifically because they choose to game exactly when it will hurt them most.

Plan of Action

Limitations of the Current Model

Our model is currently effective at detecting which students game the system and have low learning. Given our process of cross-validation, we can expect the model to make accurate predictions about other students using the same tutor for scatterplot generation and interpretation.

However, it has some limitations which need to be addressed. First of all, its predictions are only validated to be correct at the level of predicting which student is gaming after an entire class session. The model would be more useful for assigning many types of interventions if it could be relied upon to predict exactly when a student was gaming. At this point, it also might be worth investigating whether a cluster of gaming actions (which about half of gaming actions appear to occur in, according to our current model) is more useful for identifying a time to assign an intervention than a single gaming action.

Additionally, since our goal is to develop interventions which could be deployed across the entire scope of mathematics cognitive tutors, it would be preferable if our model could detect gaming in lessons other than the current one.

Verifying moment-by-moment predictions

[10] 20% was the tutor's estimated probability that the student knew a skill when they started the tutor.

In the recently completed Study 2, we made an effort to conduct our classroom observations in a fashion that will enable us to directly link our observations of gaming to specific places within the log files. We routinely synchronized our watches to the classroom machines' clocks, and synchronized all of the machines' clocks to one another. When we noted that a student was gaming the system, we wrote down the exact second displayed on our watches.

This data can be used both to validate our current model's predictions about what actions are gaming, and in training new models (a model can be trained both to the overall proportion of gaming per student, and to individual gaming actions). Note that doing so may be more complex than simply identifying the one action corresponding to our observations – gaming often takes place across several actions (both in our observations, and in our model), and there may be some work in figuring out when an episode of gaming begins and ends. Therefore, even with close synchronization, it may prove to be a challenge to map between a specific time of observation and the actions in a log that actually represent that episode of gaming.

In the event that this data is insufficient to verify that our model can predict gaming at the action-by-action level (or to improve it so that it can do so), it may be worthwhile to conduct an additional study. A tool has been developed by Vincent Aleven and his colleagues (submitted), named the Protocol Player, which graphically visualizes a log of a student's actions in the original tutor interface. This tool can be used to verify the model's action-by-action predictions of gaming by selecting a set of action-sequences which the model identifies as including gaming, and a "similar"[11] set of action-sequences which do not include gaming. Samples from each set could be shown to the original observers from studies 1 and 2, and they could identify whether each sequence included gaming or not. If their classifications were similar to the model's, we could feel confident in the model's accuracy. The tool could also be used to label individual actions as gaming or not gaming, by giving observers a random selection of actions to watch. This labeled data could be used to train a model that was more effective at identifying exactly when a student was gaming, a method similar to the one deVicente and Pain (2002) used to train a model to detect student intrinsic motivation.

Extending our model to additional tutor lessons

Our data from Study 2 gives us an excellent opportunity to learn how to extend our model to additional tutor lessons. In that study, we collected data about students' gaming behavior within four different tutor lessons (including the scatterplot lesson investigated in Study 1).

Rather than fitting a new model for each lesson, or one model for all four lessons, we would like to try to develop a replicable method for extending this model to tutor lessons where we have log files, but do not have observational data. Given that our lab has tutor data from dozens of lessons within several different tutor curricula (from Algebra to Geometry to LISP programming), but no observational data from any of those tutor lessons, developing such an approach would enable us to extend our model to all of these tutor lessons.

Hence, I will convert the model to the three other tutors investigated in Study 2 without using the observational data for those tutor lessons. I will do so in the following fashion: First, I will test whether our model is effective at making predictions about the frequency of gaming in the scatterplot lesson within Study 2. I expect that it will be

[11] For example, including a similar proportion of help, bugs, correct answers, and errors

effective at making predictions about this lesson, since the model performed well under cross-validation within Study 1.

Second, I will choose one of the other lessons researched in Study 2 (say, permutations), and I will convert the model for use within that lesson. The modeling approach was designed from the start with conversion to new lessons in mind. None of the 24 features used in developing the model were specific to the given tutor – for instance, the specific step a student was on and the specific skill exercised in that step were not used as predictors. Nonetheless, some features were used which may not generalize well between tutors. For instance, our current model finds that popup menus are highly prone to being gamed (or gaming in popup menus is highly easy to discern); but none of the other three tutor lessons studied incorporate popup menus. Directly using the same model on a tutor which lacks popup menus is likely to underpredict the true frequency of gaming. Instead, I will attempt to adapt the model directly to the differences between tutor lessons. I will do so by using data imputation (cf. Fichman and Cummings 2003) to determine what other characteristics those popup menu actions share, and to find similar actions in the new tutor's logs, which can then be identified as "popup-menu-like" – making it possible to use the same model in the new tutor with some statistical confidence as to its appropriateness. An alternate approach, if this does not work, would be to re-fit the original model with the popup menu actions removed; but doing so risks lowering the model's predictiveness.

I will convert the model to the second lesson without any reference to the observational data from the second lesson. Only after I have generated model predictions will I compare the model's predictions to the observational data. At this point, I will see where the model diverges from the actual observational data, and I will make changes to the imputation approach to address the divergences. It may be necessary to fit a model simultaneously to both the first and second tutor lessons, although this course of action will be a last resort, since the goal is to develop a method to extend the model to lessons where observational data is unavailable.

When the model is successfully generalized to the second tutor lesson, I will then attempt to generalize it to the third lesson. Hopefully, there will be considerably fewer tweaks required in generalizing to the third lesson; the goal is to then generalize to the fourth lesson, obtaining accurate prediction with no tweaks or re-fitting whatsoever. If I succeed in this goal, we will be able to be confident that the model of gaming can be extended to tutor lessons for which we have no observational data.

Prong III: Remediations for Gaming the System

Goals

Once we understand why students game the system, and can detect gaming's occurrence, we can develop interventions to guide students towards more productive use of tutoring systems. My goal is to redesign an intelligent tutor lesson in a way that reduces gaming and improves learning; it is also important that the intervention developed can be extended to a wide variety of other tutor lessons.

In this section, I will outline some possible interventions based on the different hypotheses from Prong I, and discuss my plan of action for brainstorming interventions, choosing an intervention (or set of interventions), implementing and deploying an intervention, and testing our intervention's effectiveness. I will also discuss how this process of design-and-evaluation can feed back into increasing understanding of why students game the system.

Possible Interventions/ Relevant Work

In Prong I, I discussed four possible hypotheses for why students game the system. In this section, I will discuss possible interventions which would be reasonable, given each of these hypotheses, culminating by discussing one intervention in detail for three of the hypotheses. It is important to note that the set of interventions presented here is not exhaustive. It is very possible that when we understand better why students game the system, a different intervention may seem most beneficial; it is also quite possible that another, better intervention will emerge during the process of brainstorming.

Hypothesis 1 "LACK OF INTEREST":

If students game the system because of low interest in the system, the obvious solution is to attempt to increase their intrinsic interest in using the tutor to learn. This can be accomplished either by increasing the student's interest in the material, or by increasing the student's interest in the tutor.

The students' interest in the material could be increased by increasing the material's relevance to the student – making the material seem more relevant to the student's interests and long-term goals. (Keller 1987; Malone and Lepper 1987; Joseph and Edelson 2002) This could be done by selecting examples – or allowing students to choose between examples – that seem relevant or interesting to the students.

The students' interest in the tutor could also be increased by giving the student a greater degree of personalization and choice (cf Cordova and Lepper 1996) – allowing the student to choose what types of cover stories are used in the problems they complete, for instance. The student could also be given greater ability to personalize elements of the interface, such as choosing a character to give them hint feedback. (cf. Mathan and Koedinger 2003). Finally, a richer context and story-line could be given to the problem-solving within the tutor (Cordova and Lepper 1996, Swartout et al 2001). However, if we add context and story-line, we must be careful to incorporating these added motivating features in a way that does not cause students to develop inappropriate schemas about their new knowledge, centered around the motivating features (cf. Harp and Mayer 1998).

A Possible Intervention, In Detail

If students game the system out of lack of interest in the cognitive tutors, it would likely be maximally effective to combine a number of the interest-supporting features discussed above. Past studies have suggested that interest-generating features are highly effective in concert (Cordova and Lepper 1996, Bickford 1989). It might be valuable to combine fantasy, personalization, and choice, as in Cordova and Lepper (1996), as well as relevance and story-line.

To this end, I would first determine what sorts of story-lines are both interesting and relevant to students. I would do so by bringing in middle school students to

participate in brainstorming sessions, where they could first discuss their interests and career goals, and then brainstorm story contexts they found personally interesting, relevant, and inspirational.

Then, a tutor lesson would be developed, so that the student's pattern of interaction would be as follows. First, the student would be given the **choice** of working within one of several story lines which were **relevant** to their interests, corresponding to the different topics suggested by the students during the brainstorming sessions. Whichever story line they chose, they would receive both conceptual instruction and problems that corresponded to one of their choice. The instruction and problems would be integrated into an overall **story-line** through the use of consistent characters who occur within multiple problems and whose goals and needs change across problems (cf. Cognition and Technology Group at Vanderbilt 1997) – the student could be identified as one of those characters in order to create **fantasy**. Finally, the student could be given the option of **personalizing** the problems in small ways (by changing the names of characters or places, for instance).

Despite the presence of story-line and fantasy, the mathematical terminology and explanations would still correspond to the mathematical concept being taught (rather than the story context), in order to avoid the students' developing conceptual frameworks centered around the story context. For instance, whether students were creating a graph to show the relationship between unemployment and crime, basketball training and baskets made, or forest rangers and poachers caught, they would still be asked identify whether the data set's variables were categorical or numerical.

A full system, with all of these interest-increasing features, would be tested first, and compared to a traditional system. If it was effective at reducing the incidence of gaming, an area of future work would be to see which of the individual features made particular contributions to the system's effectiveness (as in Cordova and Lepper 1996).

Hypothesis 2 "PERFORMANCE GOALS":

If students game the system because of performance goals, a different set of interventions will be preferred. In this case, remediating gaming the system will require addressing the possibility that gaming the system is a desirable strategy for students with performance goals, because the classroom incentive structure is set up in a way that does not immediately punish performance-oriented behavior.

If a student games through problem after problem, it appears to the teacher that they have completed a substantial number of problems, and that they are working hard. Eventually, they complete all of the problems in the tutor lesson (even the extra remedial ones), and at this point they either advance automatically to the next lesson, or they start seeing the same problems over and over (and may remember the answers). Sometimes, teachers even choose to advance lagging students who have completed many problems, under the hypothesis that the student will not learn the material this year, and it is better to advance them to material they *can* learn.

The most ideal solution for this problem, of course, would be to shift gaming students' behavior and deeper attitudes towards a learning orientation. Previous efforts to inculcate learning goals have had effects with interventions as minimal as reading a paragraph of instructions to the student (cf. McNeil and Alibali 2000, Elliott and Dweck, 1998); it seems possible that this approach will be less effective in less tightly supervised settings where the student is using the same tutoring software over long periods of time, although this is, of course, an empirical question. Other approaches, especially those from Behavior Modification Therapy (Sarafino 2001), may be successful at shifting students' behavior and eventually their attitudes towards learning orientation. Behavior

Modification Therapy techniques, while not widely used in educational technology, have had considerable classroom success at reducing the frequency of off-task behavior. Specifically, self-modeling, where a student repeatedly studies examples of their past positive behavior, has been effective at remediating more obviously off-task behaviors, such as disrupting class (Clare, Jenson, Kehle, and Bray 2000), and asking students to regularly monitor their own on and off task behavior has produced positive gains in the proportion of on-task behavior (Dalton, Martella, and Marchand-Martella 1999). It may be possible to combine these techniques with data from our student models, an extension of the idea of "Inspectable Student Models" in Zapata-Rivera and Greer (2003).

Another approach to eliminating gaming, if it is performance-oriented behavior, may be to remove or reduce the rewards of gaming. For instance, it may be possible to make it clear to the teacher that the student is not trying hard. We could introduce an "EFFORT METER" to correspond to the "SKILL METERS" our tutors already have (Anderson, Corbett, Koedinger, and Pelletier 1995). The "EFFORT METER" could show the teacher that, despite the fact that the student is sitting quietly and apparently working, they're actually gaming the system. The teacher could then notify the student that this will negatively affect their grade for effort (which is a significant part of the grade at the classes where we conduct our research), thus reducing the incentive for this type of performance oriented behavior.

A final possibility is to use our system's assessments of gaming to identify times when we should modify the game-able features of the system. Aleven (2001) has noted that some tutor designers eliminate "bottom-out" hints which give students the answer, and that some tutors have delays between successive levels of hints. It is also possible to replace more game-able interface widgets (such as popup menus) with slower interactions such as having the student type in the entire word which appears in the popup menu. While we do not recommend these interventions in general (because they make tutors more difficult and frustrating for students who use the tutors properly), they may be an appropriate response to performance-oriented gaming. If a student was informed by the system that his/her gaming behavior had led the system to become more difficult to use, he or she might game less when the system returned to being more game-able.

A Possible Intervention, In Detail

If students game the system out of performance orientation, one possible intervention, as discussed above, would be to use a combination of self-modeling and self-monitoring.

A student would be chosen to receive this intervention if they were observed frequently gaming the system on the previous class day (by the system). At the beginning of the class session, they would be shown a 5-minute collection of examples of proper use of the tutor, from their own behavior on the previous day (cf. Clare et al, 2000). These examples would be automatically identified using a variant of the gaming detection algorithm, training instead on the behavioral patterns of non-gaming students with high learning gains. Since no student was observed gaming more than half of the time in Study 1, it should be possible to find a reasonably high number of positive examples. The examples would be shown using Aleven et al's (submitted) Protocol Player.

As the student watched the examples of proper use, annotation would be automatically created by the system and given to the student. This annotation would explain what the student was doing in these examples, and why this type of behavior was an effective way to learn from the tutor. The annotation would be modeled on previous protocols for delivering self-modeling interventions, used by school psychologists. The

annotation would emphasize the fact that the student was watching his or her own behavior.

After the collection of examples had concluded, the student would begin work in the tutor, as normal. At this point, the self-monitoring part of the intervention would begin. Every 5 minutes, the system would ask the student to identify whether they thought they had been using the software in a learning-oriented fashion (cf. Dalton et al, 1999), and would give them appropriate feedback on their self-assessment. Rather than interrupting the student in the middle of thinking about a step, the self-monitoring system would pop up immediately after the student had completed a step. The self-monitoring part of the system would also pop up sooner than after 5 minutes, if the detection algorithm determined that the student was gaming the system.

If the student's gaming did not reduce below a pre-chosen threshold during the course of the intervention, the student would receive the intervention again on the following tutor day. If the student's gaming did reduce, then they would not receive the intervention again.

The system with a combination of these two interventions would be tested first, and compared to a traditional system. If it was effective at reducing the incidence of gaming, an area of future work would be to see whether it was necessary to use both interventions, or whether one would suffice.

Hypothesis 3 "FRUSTRATION/LEARNED HELPLESSNESS":

If students game the system out of frustration/learned helplessness, then the tutor should adapt to become more encouraging and less frustrating.

One approach is to attempt to reduce the impact of the elements of the tutor which are frustrating. An approach that has been successful at reducing frustration in some situations is emotional scaffolding. In emotional scaffolding, praise, encouragement, and other feedback is given to the student in order to moderate the student's frustration and keep them on task (Aist et al 2002; cf. Lepper and Chabay 1985) – the goal is to encourage the student that they will eventually be able to accomplish the task (Keller and Suzuki 1988), so long as they apply sufficient effort (Dweck 1975). A rich variety of emotional scaffolding techniques are used by expert human tutors (Lepper, Wolverton, Mumme, and Gurtner 1993), and human-delivered emotional scaffolding has been shown to increase the educational effectiveness of intelligent tutoring systems (Aist et al 2002). However, emotional scaffolding has been less successful when delivered by a system, possibly because of difficulty in assessing exactly when a student is frustrated (Mostow, Beck, and Valeri 2003).

Another approach is for the tutor to attempt to give different assistance when the student is frustrated. Human tutors seldom give frustrated students the sort of direct negative feedback used in the current cognitive tutors, due to concerns about harming student self-confidence (Lepper, Woolverton, Mumme, and Gurtner 1993). Instead, human tutors instead often give error-detection-and-correction help, as in Mathan and Koedinger (2003), or respond to errors by breaking down problems, as in Heffernan (2002). It has also been found that giving students help appropriate to their level of cognitive development[12] both increases the student's self-confidence and their self-report as to how much they like mathematics (Arroyo 2003).

A third approach, recommended in (Keller and Suzuki 1988), would probably not be appropriate for our specific educational situation. Keller and Suzuki recommend that

[12] Arroyo also reports an effect due to gender, although in her study gender and preferred learning style were correlated.

students experiencing frustration be given easier problems to first build confidence. However, our data from Prong II indicates that even high-gaming students (who learn little) find some problem steps easy, and game considerably less on those problem steps, compared to harder problem-steps. Hence, it appears that the confidence students are learning on the easier steps is not translating, in this case, to higher confidence on the more difficult problem-steps.

A Possible Intervention, In Detail

If students game the system out of frustration and learned helplessness, one possible intervention, as discussed above, would be to have the tutor give emotional scaffolding. Aist, Mostow, and their colleagues (2002) have developed a corpus of audio emotional scaffolding given by a teacher to frustrated students, which would be useful for developing an intervention.

First, audio segments would be selected out of that corpus (or, if their corpus is not appropriate for our older student population, a wizard of oz study could help us generate such a corpus). As soon a student was detected gaming the system, a supportive audio sample would be played for the student. In order to encourage proper use, positive and encouraging audio samples would also be played for the student whenever they mastered a skill, or had been working for a long time without gaming the system.

Some adjustment would need to be made to be certain that the audio segments were not being given too frequently – and thus becoming annoying; it might be wise to limit the number of audio interventions that take place within a certain time period. Pilot studies could help us determine what the maximum number of audio interventions should be.

A system with this intervention would be tested for its effects on the frequency of gaming, compared to a traditional system. It would also be important to test whether the audio interventions continued to be effective over a number of weeks, or whether their effectiveness reduced over time. Judicious use of audio interventions might reduce the speed with which they lost their effectiveness.

Hypothesis 4 "LACK OF METACOGNITIVE KNOWLEDGE":

If students game the system because they do not understand that it hurts their learning, it may be relatively easy to develop remediations. In this case, gaming could probably be treated as a "bug" (procedural misconception) (Anderson et al 1995; vanLehn 1990) or conceptual misconception (Clement 1982; Minstrell 1989; Baker, Corbett, Koedinger, and Schneider 2004) in the student's metacognitive knowledge, and we could use the same methods that are effective for cognitive misconceptions.

For instance, if a student was observed to be gaming the system, they could be given conceptual instruction on appropriate strategies for using the tutor (cf. Nastasi and Clements 1992, Paris and Winograd 1990, Aleven 2001) and then, if the tutor observed the student gaming the system, they could be given immediate feedback (as in Corbett and Anderson 1995) reminding them that gaming the system was not effective for learning, and suggesting that they (for instance) ask for a hint and read it slowly. Ongoing work in developing immediate feedback for students' lack of knowledge of how to properly use help (Aleven, McLaren, Roll, and Koedinger submitted) may provide useful inspiration and opportunities for cross-fertilization.

Given the low evidence for this hypothesis within our current data, and the existence of a body of work in analyzing the proper way for students to use tutors,

developing models of proper usage, and designing feedback based on that model (Aleven et al, submitted), it is unlikely that I will choose this type of intervention as a primary focus of this thesis. However, it may be a valuable comparison condition to an intervention stemming from one of the other three hypotheses; it may also be a valuable supplement to such an intervention. In this case, I will probably adopt and adapt the examples of metacognitive feedback currently under development by Aleven and his colleagues.

Plan of Action

Our first step towards developing interventions will be to advance further on Prong I: understanding why students game the system. Our evidence from study 2 will give considerable insight as to which of our hypotheses are most reasonable. It is also possible that our results will produce results inconsistent with any of our hypotheses, or that they will provide evidence suggesting that students game the system for different reasons, and that different reasons for gaming are consistent with different learning outcomes (for instance, GAMED-NOT-HURT students who game the system but still learn may game out of lack of interest, choosing to game on the least interesting (and easiest) parts of the tutor, whereas GAMED-HURT students who game the system and learn very little may game out of frustration, choosing to game on the most difficult parts of the tutor).

After we have completed our analysis of study 2, I will lead structured brainstorming sessions on possible interventions. Outside experts will be invited to participate, including teachers that our project collaborates with, other educational/cognitive psychology researchers at CMU and U. Pitt, and as possible, behavior modification therapists and researchers (cf. Sarafino 2001). Participants will be given a short presentation on the Prong I research and data, including evidence for each hypothesis. They will also be shown clips (using Aleven et al's [submitted] Protocol Player) of students gaming the system. In situations where some brainstorming session participants are not familiar with classroom conditions, ethnography as theatre techniques may be used to help the teachers convey an understanding of classroom conditions to the less familiar participants (Laurel 2003). The participants will then be asked to brainstorm possible interventions, to select a couple of interventions they particularly like, and to sketch their proposed interventions (Beyer and Holzblatt 1998); I will act as a facilitator in these brainstorming sessions. Finally, I will show the participants a couple of sketches (either produced by me before the first session, or from previous participants), and have them critique it.

My goal is to conduct a classroom pilot study of a prototype in at least one classroom in Fall 2004. Depending on the intervention chosen, it may be a prototype implementation, or it may be a Wizard of Oz intervention. In the Wizard of Oz case, the system will still choose when the intervention should be delivered (using the detection algorithm discussed in Prong II) but the intervention will be human-delivered (either by me, or by the student's teacher, as appropriate). In order to avoid potential observer bias, the intervention's effectiveness will be measured by the system's assessment of how often the student games the system before and after the intervention (including during the next class session). Multiple interventions may be tried in this study.

A full study will be put into several classrooms the following spring (or, if unforeseen circumstances occur, the following fall). In this study, we will test the effectiveness of our intervention, when it is delivered entirely without human participation. Depending on the clarity of our results from Prong I, we may test the relative effectiveness of interventions based on different hypotheses (and in this case, it

would be very important to choose interventions which seem very unlikely to work with the other hypotheses) – alternatively, we may test whether two interventions targeted towards the same hypothesis are more effective in concert. Students will complete three tutor units. In the first unit, we will determine how often each student games the system. We will then divide students who game a similar proportion of the time (in a way that the detector believes is inconsistent with learning), and who have similar pre-test scores, into pairs (intervention and no intervention) or triads (intervention one, intervention two, and no intervention). Each member of each pair/triad will be randomly assigned to a different condition. In the second tutor unit, the intervention will be given, in order to test the intervention's immediate effectiveness for the current subject matter. In the third tutor unit, no intervention will be given, in order to test whether the interventions produce sustained effects. Results will be assessed both by the system's assessment of the frequency of gaming, and by pre-post gains.

A third study may be conducted within Prong III, if the results of the second study are inconclusive, or if it appears that the effectiveness of the intervention was reduced by implementation errors or design flaws unrelated to the design's overall intention.

Schedule (proposed)

A proposed schedule is given below. Please note that it is liable to change. One factor that is especially likely to delay this schedule – by about four months – is the possibility of delays in running the Apr/May 2005 study. Such delays could arise from difficulty arranging a sufficient number of school sites, or from difficulties in negotiations with CMU's Institutional Review Board. Additionally, if a third study needs to be run in Fall 2005, the thesis defense may occur around four months later.

Date	Prong I Understanding Why (PSYCHOLOGY)	Prong II Detecting (COMP SCI)	Prong III Remediating (DESIGN)
Summer 2004	Analyze Study 2 Data	Verify Detection Algorithm's Effectiveness for Scatterplot Lesson Verify Effectiveness at Determining *When* a student is gaming	Structured Brainstorming
Fall 2004		Extend Detection Algorithm to Other Lessons	Develop Prototype Conduct Study 3
Spring 2005			Develop Full System Conduct Study 4
Summer- Fall 2005	Analyze Study 4 Data Writing	Develop Final Model Writing	Analyze Data Writing
Dec 2005	Defend Thesis	Defend Thesis	Defend Thesis

Conclusions

In this thesis proposal, I have presented a multi-disciplinary program of research that can be expected to produce positive results in the classroom, to make contributions to each of the disciplines which contribute to it, and to provide a valuable case study in how these disciplines can be integrated into the still-emerging discipline of Human-Computer Interaction.

This research project will serve as a valuable case study in motivationally appropriate design and adaptation to how users' different goals shape their interaction with a system. It will demonstrate the use of machine learning of a psychometric model for the joint goals of detecting and understanding behavior, in order to build adaptive systems and help a designer decide which adaptive system to build. This project will increase our general knowledge of how and why students misuse educational opportunities. Finally, it will produce measurable improvements in the educational effectiveness of cognitive tutors for mathematics, which are already one of the most effective educational systems in wide-spread use today.

Acknowledgements

I would like to thank my advisors Albert T. Corbett and Kenneth R. Koedinger for their exceptionally helpful advice and guidance. Tom Mitchell and Shelley Evenson have given very helpful suggestions and feedback. The research discussed here could not have happened without Angela Wagner's assistance in coordinating studies and collecting observations; similarly, the studies discussed could not have taken place without the participation of teachers such as Jay Raspat, Meghan Naim, Katy Getman, Russell Hall, Susan Cameron, Patricia Battaglia, and Dina Crimone, and assistance with data-entry from Jane Kamneva, Heather Frantz, and Pauline Masley. Elspeth Golden, Amy Hurst, Ido Roll, and Peter Scupelli lent much-needed processor cycles and participated in very helpful discussions. Finally, I would like to thank Shaaron Ainsworth, Vincent Aleven, Lisa Anthony, Daniel Baker, Samuel Baker, Joseph Beck, Darren Gergle, Brian Junker, Andrew Ko, George Loewenstein, Santosh Mathan, Jack Mostow, Rachel Roberts, Michael Schneider, Deborah Small, and Desney Tan for their advice and suggestions throughout this research project so far.

References

Aist, G., Kort, B., Reilly, R., Mostow, J., & Picard, R. (2002). Experimentally Augmenting an Intelligent Tutoring System with Human-Supplied Capabilities: Adding Human-Provided Emotional Scaffolding to an Automated Reading Tutor that Listens. *ITS 2002 Workshop on Empirical Methods for Tutorial Dialogue Systems.*

Aleven, V. (2001) Helping Students to Become Better Help Seekers: Towards Supporting Metacognition in a Cognitive Tutor. Paper presented at *German-USA Early Career Research Exchange Program: Research on Learning Technologies and Technology-Supported Education,* Tubingen, Germany.

Aleven, V., McLaren, B., Roll, I., Koedinger, K. (submitted) Toward Tutoring Help-Seeking: Applying Cognitive Modeling to Meta-cognitive Skills. Submitted to *Intelligent Tutoring Systems.*

Allscheid, S.P., Cellar, D.F. (1996) An Interactive Approach to Work Motivation: The Effects of Competition, Rewards, and Goal Difficulty on Task Performance. *Journal of Business and Psychology,* 11 (2), 219-237.

Anderson, J.R. (1983) *Rules of the Mind.* Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences*, 4(2), 167-207.

Arbreton, A. (1998) Student Goal Orientation and Help-Seeking Strategy Use. In S.A. Karabenick (Ed.), *Strategic Help Seeking: Implications For Learning And Teaching,* pp. 95-116, Mahwah, NJ: Lawrence Erlbaum Associates.

Arroyo, I. (2000) AnimalWatch: An Arithmetic ITS for Elementary and Middle School Students. Paper presented at "Learning Algebra with the Computer" Workshop, Fifth International Conference on Intelligent Tutoring Systems.

Arroyo, I. (2003) *Quantitative evaluation of gender differences, cognitive development differences and software effectiveness for an elementary mathematics intelligent tutoring system*. Dissertation, Department of Education, University of Massachusetts at Amherst.

Baker, R.S., Corbett, A.T., Koedinger, K.R. (2003) Statistical Techniques For Comparing ACT-R Models of Cognitive Performance. In *Proceedings of the 10th Annual ACT-R Workshop*, 129-134.

Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". To appear at *ACM CHI: Computer-Human Interaction*.

Baker, R.S., Corbett, A.T., Koedinger, K.R., Schneider, M.P. (2004) Learning to Distinguish Between Representations of Data: a Cognitive Tutor That Uses Contrasting Cases . To appear at *Learning Sciences Conference*.

Bandalos, D.L., Finney, S.J., Geske, J.A. (2003) A Model of Statistics Performance Based on Achievement Goal Theory. *Journal of Educational Psychology,* 95 (3), 604-616.

Baumeister, R.F. (1984). Choking under pressure: self-consciousness and paradoxical effects of incentives on skillful performance. *Journal of Personality and Social Psychology*, 46, 610-620.

Beck, J.E., Jia, P., Sison, J., Mostow, J. (2003) Predicting student help-request behavior in an intelligent tutor for reading. In *Springer-Verlag Lecture Notes on Artificial Intelligence¸* 2702 (User Modeling).

Beyer, H., Holtzblatt, K. (1998) *Contextual Design: Defining Customer-Centered Systems.* London, UK: Academic Press.

Belavkin, R.V. (2001). The Role of Emotion in Problem Solving. In *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing*, 49-57.

Belavkin, R. V., Ritter, F. E., & Elliman, D. G. (1999). Towards including simple emotions in a cognitive architecture in order to fit children's behaviour better. In *Proceedings of the 1999 Conference of the Cognitive Science Society*.

Bickford, N. (1989) The Systematic Application of Principles of Motivation to the Design of Instructional Materials. Doctoral Dissertation, Florida State University.

Castillo, G., Gama, J., Breda, A.M. (2003) Adaptive Bayes for a Student Modeling Prediction Task Based on Learning Styles. *Springer-Verlag Lecture Notes on Artificial Intelligence¸* 2702 (User Modeling), 328-332.

Cavaluzzi, A., De Carolis, B., Carofiglio, V., Grassano, G. (2003) Emotional Dialogs with an Embodied Agent. *Springer-Verlag Lecture Notes on Artificial Intelligence*, 2702 (User Modeling), 86-95.
Clare, S.K., Jenson, W.R., Kehle, T.J., Bray, M.A. (2000) Self-Modeling As a Treatment For Increasing Off-Task Behavior. *Psychology in the Schools,* 37 (6), 517-522.
Clement, J. (1982) Students' preconceptions in introductory mechanics. *American Journal of Physics,* 50, 66-71.
Cognition and Technology Group at Vanderbilt (1997) The Jasper Project: Lessons in Curriculum, Instruction, Assessment, and Professional Development. Mahwah, NJ: LEA.
Cohen, B. A. & Waugh, G. W. (1989). Assessing computer anxiety. *Psychological Reports*, 65, 735-738.
Corbett, A.T. & Anderson, J.R. (1995) Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction,* 4, 253-278.
Cordova, D. I., & Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology, 88,* 715-730.
Dalton, T., Martella, R.C., Marchand-Martella, N.E. (1999) The Effects of a Self-Management Program in Reducing Off-Task Behavior. *Journal of Behavioral Education,* 9 (3-4), 157-176.
de Vicente, A., Pain, H. (2002) Informing the detection of the students' motivational state: an empirical study. In S. A. Cerri, G. Gouarderes, F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems*, 933-943.
Donaldson, W. (1993) Accuracy of d' and A' as estimates of sensitivity. *Bulletin of the Psychonomic Society,* 31 (4), 271-274.
Dweck, C.S. (1975) The Role of Expectations and Attributions in the Alleviation of Learned Helplessness. *Journal of Personality and Social Psychology,* 31 (4), 674-685.
Elliott, E.S., Dweck, C.S. (1988) Goals: An Approach to Motivation and Achievement. *Journal of Personality and Social Psychology,* 54 (1), 5-12.
Fichman, M., & Cummings, J. (2003). Multiple imputation for missing data: Making the most of what you know. *Organizational Research Methods, 6*(3), 282-308.
Hanley, J.A., McNeil, B.J. (1982) The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology,* 143, 29-36.
Harnisch, D. L., Hill, K. T., & Fyans, L. J., Jr. (1980). Development of a Shorter, More Reliable, and More Valid Measure of Test Motivation. Paper presented at the meeting of the National Council on Measurement in Education (NCME).
Harp, S.F., Mayer, R.E. (1998) How Seductive Details Do Their Damage: A Theory of Cognitive Interest in Science Learning. *Journal of Educational Psychology,* 90 (3), 414-434.
Harter, S. (1981) A New Self-Report Scale of Intrinsic Versus Extrinsic Orientation in the Classroom: Motivational and Informational Components. *Developmental Psychology,* 17 (3), 300-312.
Heffernan, N. T., (2002) An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor. *International Conference on Intelligent Tutoring Systems.*
Heiner, C., Mostow, J., Beck, J. (submitted) Learning to Detect Frustration in Children's Oral Reading.
Jacob, B.A., Levitt, S.D. (in press) Catching Cheating Teachers: The Results of an Unusual Experiment in Implementing Theory. To appear in *Brookings-Wharton Papers on Urban Affairs.*

Jones, R.M., Henninger, A., Chown, E. (2002) Interfacing Emotional Behavior Monitors With Intelligent Synthetic Forces. Presentation at *22nd Annual SOAR Workshop.* Ann Arbor, MI.
Joseph, D., Edelson, D. (2002) Engineering Motivation: Using research knowledge about motivation in the design of learning environments. *International Conference on the Learning Sciences,* 2002.
Keller, J.M. (1983) Motivational Design of Instruction. In C.M. Reigeluth (Ed.) *Instructional Design Theories and Models: An Overview of the Current Status* (pp. 383-434). Hillsdale, NJ: Erlbaum.
Keller, J.M. (1987) Strategies for Stimulating the Motivation to Learn. *Performance and Instruction,* 26(8), 1-7.
Keller, J.M., Suzuki, K. (1998) Use of the ARCS Model in Courseware Design. In D.H. Jonassen (Ed.), *Instructional Designs for Computer Courseware.* New York: Lawrence Erlbaum.
King-Stoops, J., Meier, W. (1978) Teacher Analysis of the Discipline Problem. *Phi Delta Kappan,*59, 354.
Klawe, M.M. (1998) Designing Game-based Interactive Multimedia Mathematics Learning Activities. *Proceedings of UCSMP International Conference on Mathematics Education.*
Langley, P., Simon, H., Bradshaw, G., Zytkow, J. (1987) *Scientific Discovery.* Cambridge, MA: MIT Press.
Laurel, B. (2003) *Design Research: Methods and Perspectives.* Cambridge, MA: MIT Press.
Lepper, M.R., Greene, D. (1978) *The Hidden Costs of Reward.* Hillsdale, NJ: Lawrence Erlbaum Associates.
Lepper, M.R., Chabay, R.W. (1985) Intrinsic Motivation and Instruction: Conflicting View on the Role of Motivational Processes in Computer-Based Education. *Educational Psychologist,* 20 (4), 217-230.
Lepper, M. R., Woolverton, M., Mumme, D. L., & Gurtner, J. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as cognitive tools* (pp. 75-105). Hillsdale, NJ: Erlbaum.
Lloyd, J.W., Loper, A.B. (1986) Measurement and Evaluation of Task-Related Learning Behavior: Attention to Task and Metacognition. *School Psychology Review,* 15 (3), 336-345.
Luckin, R., du Boulay, B. (1999) Capability, Potential, and Collaborative Assistance. In *Proceedings of the Seventh Annual Conference on User Modeling,* 139-147.
Malone, T. W., & Lepper, M.R. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. In R. E. Snow & M. J. Farr (Eds.). *Aptitude, learning and instruction*. Volume 3: Cognitive and affective process analysis. Hillsdale, NJ: Lawrence Erlbaum.
Maris, E. (1995) Psychometric Latent Response Models. *Psychometrika,* 60 (4), 523-547.
Martin, J., vanLehn, K. (1995) Student Assessment Using Bayesian Nets. *International Journal of Human-Computer Studies,* 42, 575-591.
Mathan, S., Koedinger, K. (2003) Recasting the Feedback Debate: Benefits of Tutoring Error Detection and Correction Skill. *Conference on Artificial Intelligence in Education*, 13-20.
McNeil, N.M., Alibali, M.W. (2000) Learning Mathematics From Procedural Instruction: Externally Imposed Goals Influence What is Learned. *Journal of Educational Psychology,* 92 (4), 734-744.

Miller, C.S., Lehman, J.F., Koedinger, K.R. (1999) Goals and Learning in Microworlds. *Cognitive Science,* 23 (3), 305-336.

Minstrell, J. (1989) Teaching Science for Understanding. In Resnick, L.B., and Klopfer, L.E. (Eds.) *Toward the Thinking Curriculum: Current Cognitive Research.* Alexandria, VA: Asssociation for Supervision and Curriculum Development, 129-149.

Moore, A. (2004) Cross-Validation. In *Statistical Data Mining Tutorials.* http://www.cs.cmu.edu/~awm/tutorials/

Mostow, J., Aist, G., Beck, J., Chalasani, R., Cuneo, A., Jia, P., Kadaru, K. (2002) A La Recherche du Temps Perdu, or As Time Goes By: Where does the time go in a Reading Tutor that listens? Paper presented at *Sixth International Conference on Intelligent Tutoring Systems (ITS'2002).*

Mostow, J., Beck, J. E., & Valeri, J. (2003). Can Automated Emotional Scaffolding Affect Student Persistence? A Baseline Experiment. *Workshop on "Assessing and Adapting to User Attitudes and Affect: Why, When and How?" at the 9th International Conference on User Modeling (UM'03)*

Mueller, C.M., Dweck, C.S. (1998). Praise for intelligence can undermine children's motivation and performance. Journal of Personality and Social Psychology , 75 (1), 33-52.

Nastasi, B.K., Clements, D.H. Social-Cognitive Behaviors and Higher-Order Thinking in Educational Computer Environments. *Learning and Instruction*, 2(3), 215-238.

Nielsen, J. (1993) *Usability Engineering.* San Diego, CA: Morgan Kauffman.

Paris, S. G., & Winograd, P. (1990). How metacognition can promote academic learning and instruction. In B. F. Jones & L. Idol (Eds.), *Dimensions of thinking and cognitive instruction* (pp. 15-51). Hillsdale, NJ: Erlbaum.

Picard, R.W. (1997) *Affective Computing.* Cambridge, MA: MIT Press.

Ramsey, F.L., Schafer, D.W. (1997) *The Statistical Sleuth: A Course in Methods of Data Analysis.* Belmont, CA: Duxbury Press.

Romero, C., Ventura, S., de Bra, P., de Castro, C. (2003) Discovering Prediction Rules in AHA! Courses. *Springer-Verlag Lecture Notes on Artificial Intelligence,* 2702 (User Modeling), 25-34.

Rudner, L. (1998) *An On-Line, Interactive, Computer Adaptive Testing Tutorial.* http://edres.org/scripts/cat

Sarafino, E.P. (2001) *Behavior Modification: Principles of Behavioral Change.* Mountain View, CA: Mayfield Publishing.

Sarason, S.B. (1978) *Anxiety in elementary school children; a report of research.* Westport, CT: Greenwood Press.

Scheutz, M. (2003) Affective Agent Architectures. Presentation at the *23rd Annual SOAR Workshop.* Ann Arbor, MI.

Schofield, J.W. (1995) *Computers and Classroom Culture.* Cambridge, UK: Cambridge University Press.

Schunn, C. & Anderson, J. R. (1998). Scientific discovery. In J. R. Anderson, & C. Lebiere (Eds.). *The atomic components of thought*, 255-296. Mahwah, NJ: Erlbaum.

Simon, H.A. (1967) Motivational and Emotional Controls of Cognition. *Psychological Review*, 74, 29-39.

Singley, M.K., Anderson, J.R. (1981) *The Transfer of Cognitive Skill.* Cambridge, MA: Harvard University Press.

Smith, B., Caputi, P. (2001) Cognitive interference in computer anxiety. *Behavior and Information Technology,* 20 (4), 265-273.

Stevens, R., Soller, A., Cooper, M., Sprang, M. (submitted) Modeling the Development of Problem-Solving Skills in Chemistry with a Web-Based Tutor.

Stiensmeier-Pelster, J., Schurmann, M. (1990) Performance Deficits Following Failure: Integrating Motivational and Functional Aspects of Learned Helplessness. *Anxiety Research,* 2, 211-222.
Swartout, W., Hill, R., Johnson, W.L., Kyriakakis, C., LaBore, C., Lindheim, R., Marsella, S., Miraglia, D., Moore, B., Morie, J., Rickel, J., Thiébaux, M. Tuch, L., Whitney, R., Douglas, J. (2001) Toward the Holodeck: Integrating Graphics, Sound, Character, and Story. *Proceedings of the Fifth International Conference on Autonomous Agents,* 409-416.
Turner, L.A., Johnson, B. (2003) A Model of Mastery Motivation for At-Risk Preschoolers. *Journal of Educational Psychology,* 95 (3), 495-505.
vanLehn, K. (1990) *MindBugs: The Origins of Procedural Misconceptions.* Cambridge, MA: MIT Press.
Vendlinski, T., Stevens, R. (2002) Assessing Student Problem-Solving Skills With Complex Computer-Based Tasks. *Journal of Technology, Learning, and Assessment,* 1 (3).
Winter, S. (1991) Are Behavioral Classroom Management Approaches Appropriate to a non-Western Educational System? *School Psychology International,* 12 (3), 211-223.
Yerkes, R.M., Dodson, J.D. (1908) The Relation of Strength of Stimulus to Rapidity of Habit-Formation, *Journal of Comparative Neurology and Psychology,* 18, 459-482.
Zapata-Rivera, J.D., Greer, J. (2003) Student Model Accuracy using Inspectable Bayesian Student Models. *International Conference on Artificial Intelligence in Education,* 65-72.