Automatic Detection of Student Off-Task Behavior while using an Intelligent Tutor for Algebra

Allan Edgar C. BATE
Ateneo de Manila
Loyola Heights
allbate@gmail.com

Ma. Mercedes T. RODRIGO
Ateneo de Manila
Loyola Heights
mrodrigo@ateneo.edu

ABSTRACT

As more and more modern classrooms use intelligent tutoring systems, it becomes imperative for our educators to determine whether these systems are being used properly. While using an intelligent tutor, it is possible for students to engage in off-task behavior, defined as actions that show disengagement from learning. Off-task behavior can range from resting one's eyes, to talking to one's seatmate, to "gaming the system" defined as abusing regularities of the intelligent tutor to progress through the curriculum without actually learning the material. These behaviors constitute time away from the learning task and are therefore considered detrimental to learning. In this paper, we attempt to create a model that automatically detects learner off-task behavior while using Aplusix, an intelligent tutor for algebra. By analyzing logs of interactions recorded by the Aplusix, we determine off-task behavior's quantifiable characteristics. Afterwards, we use machine learning techniques to create a model of off-task behavior. Automatic detection can lead to interventions that can retain student attention and increase learning.

Keywords

Affective Computing, Intelligent Tutoring Systems, Machine-learning, Aplusix, Off-task behavior

1. INTRODUCTION

Intelligent tutoring systems (ITSs) are a subtype of computer-based learning system that makes use of artificial intelligence to increase teaching effectiveness. They are composed of three main models that interact with each other to create a system capable of not only to teach the student but to learn from the student's performance and thus improve itself:

The *expert* model or *domain* model is the domain of knowledge the tutor teaches. Intelligent Tutoring Systems cover a certain field of expertise in which it aims to tutor students. This model contains the problem solving expertise, skills, concepts, and facts of its curriculum. [5]

The *student* model describes the students' problem-solving performance. It records interactions between the tutor and student and analyzes performance to whether the student got the problem correct or not, how many tries it took, and so on. As more and more information regarding the student, such as a means to track motivation, the student model continues to expand as developers add more sub-models into it [8].

The *pedagogical* model combines the knowledge of these two models and designs a teaching method, thus providing ample explanations and exercises for the students to learn based on its domain or expertise [5].

ITSs' interactivity, their ability to provide customized feedback, and their ability to adjust the level of challenge, have been shown to increase student motivation [4]. This motivation sparks individual initiative from students as a research by Koedinger finds students coming into the lab outside of regular class time to work with the system. They also found that the use of the ITS generally raised the average of students' scores [9].

1.1. Statement of the Problem

While the use of ITS does show improvements in learning it still has its limitations. ITSs are susceptible to off-task behavior, a behavior that denotes disengagement from the learning experience [7] and is associated with poor learning [1].A study by Koedinger et al. [9] found that while students are indeed motivated to learn using an ITS, there are cases where they end up doing trial-and-error or randomly entering statements as they use the software. Baker et al [1] found that the students who engaged in off-task behavior while using an ITS learned only two-thirds of the subject matter compared to students who used the tutor properly.

1.2. Research Objectives

In this paper, we will attempt to automatically detect off-task behavior when it is exhibited during the use of an ITS. In order to prevent the loss of learning opportunities for the student while using ITSs, we will attempt to create a model that will be able to detect off-task behavior during the students' use of the ITS. We will make use of the particular ITS, Aplusix, for this research. If the nature of this off-task behavior indeed affects learning, it may make ITS more effective if it can immediately call to the attention of the student when the student begins to fall under the category of performing such off-task behaviors.

To this end, we record student interactions with Aplusix, ITS for algebra, e.g. key pressed, state of the problem with regard to the solution, and so on. We ask two experts in the field of education to label each record as indicating student on-task or off-task behavior.. We then use machine-learning software to analyze our labeled data, using WEKA, following the methods of Walonoski and Heffernan [11].

1.3. Research Questions

To accomplish this we will break down our goal into two

questions we aim to answer as we go about our research. Certain concepts such as low-fidelity playback and off-task behavior will be discussed later on:

1. *What information do we need to have a significantly valid low-fidelity playback of the use of Aplusix?* As we will later explain, we will base our data not on live observations of the students' usage but on log files generated by Aplusix on the actions that took place during each exercise. Work by Baker [2] shows that low-fidelity playbacks can be used to develop accurate models of off-task behavior. The challenge rests in determining how many features are necessary to make the recognition of the behavior possible.

2. *What are the different patterns of behavior that displays off-task behavior of the student?* By asking for the help of two experts in identifying student behavior in the classroom, we identify the different patterns of behavior found in our data logs to whether they are on-task or off-task. For further clarification, we will ask our experts to why they are labeled as such in order to help us reinforce the heuristics in detecting off-task behavior.

1.4. Significance

In the continuous development and evolution of these educational programs, we hope to tackle problems like these in order to increase the effectiveness of computer-based learning. In traditional classrooms, teachers are able to identify when students start to lose interest and intervenes to correct them.. Being able to detect off-task behavior in real time will allow ITS developers to create interventions that correct the student disengagement. By allowing developers to provide more ways and opportunities to give feedback to students, we hope to ultimately increase the learning that students can achieve when using ITS.

2. THEORETICAL FRAMEWORK

2.1. Off-task Behavior

Off-task behavior occurs when students exhibit disengagement from the learning experience, usually due to the lack of motivation [10]. It occurs in traditional classrooms, where students disengage from participating in class and begin to perform actions unrelated to the subject matter at hand. Common examples would be talking to one's seatmate, reading a book, doodling, or passing notes about things that do not concern the lesson. Off-task behavior can also exhibit itself as inactivity such as resting one's eyes, putting one's head on the table, daydreaming, or sleeping [1].

The same lack of motivation surfaces with computer-aided learning as it is in traditional classrooms. Lack of prior knowledge about the lesson, lack in confidence in learning the lesson, lack of experience with the use of the computer, lack of interest in the matter are a few of the reasons where students are found to perform off-task behavior when using computer-based tutors [11].

Studies show that off-task behavior indeed takes a toll on the learning gained by students [1]. Students that show unexpected behavior not only undermines the learning process but also affects the ITS's capability to analyze the students' performance and improve [5]. Such behavior is not the intended use of the ITS and thus developers continue to improve to software to overcome such limitations [9].

Because off-task behavior becomes is detrimental to learning with ITSs, studies have attempted to improve ITS capacity to detect this behavior. Such studies, like those of Baker [1] and Walonoski [11], make use of live-observation during experimentation to record when students are off-task. These researchers based their judgments on students' actions and facial expressions.

Their focus was specifically on "gaming the system", a type of off-task behavior in which students abuse the limitations of the ITS in order to progress through the curriculum without learning the subject matter. Examples of gaming the system include systematic guessing and hint abuse. Baker's model predicted gaming based the number of errors on each problem, quick reaction times after an error, and identifying if the student is supposed to know this problem (based on pre-test and prior problems solved) but has made some slips. Their classifier was able to detect 88% of students who gamed and 15% of students who did not [1]. Walonoski's study attempted to detect student gaming within the *Assistments* System. They used a machine-learned decision-tree model that detected gaming at level of accuracy. The study also validated certain findings such as how much gaming affects the learning of the student, and that low prior knowledge is greatly correlated to off-task behavior [11].

2.2.Low-Fidelity Playbacks

Fidelity refers to the accuracy of data[1]. Data gathering is done with different levels of fidelity depending on how accurate our information will be in re-enacting or replaying our experiment. High fidelity refers to the gathering of a variety of information, some of which is not visible to the naked eye. High fidelity data includes full videos of the experiment, from different angles, live observations, interaction logs, and biometrics sensors. High fidelity data, however, require a lot of time and resources to gather. It also requires special equipment such as cameras and sensors.

Low fidelity data such as interaction logs alone does not require as many resources. These can be easily gathered, assuming the software in use has a recording feature.

To label low fidelity data, we play back user interactions and infer the correct label from the actions of the user., A study by Baker on the use of low-fidelity replays instead of live-observation compares the interrater reliability between the two types of observation. The interrater reliability of high fidelity observation was found to be higher but low fidelity replays were still found to be sufficient for analyzing captured behavior and have become the preferred method given for its convenience and availability [2].

[1] http://www.merriam-webster.com/dictionary/fidelity

2.3. Log File Analysis

Log file analysis is the systematic approach to examining and interpreting the content of behavioral data [3]. Log file analysis approaches include:

Transition analysis refers to the analysis of the changes in behavior. This requires the experimenter to define a strict domain of actions of interest, distinguishable based on a set of variables. .

Frequency analysis is the tallying frequencies of the actions and computing for their different statistics such as averages, and standard deviations. This method can derive different statistics for individuals and groups of subjects and thus examine their interactional patterns. Implementation of this method is easy but it has its drawbacks as a standalone, because interpretations of its results are vague and wide in range.

The **learning-indicator approach**, similar to frequency approach, consists of clustering actions that have close-to-similar frequencies and determining groups in a global coverage. As with frequency approach, it ignores behavior changes or progresses over time. It gives broad overviews of behavior but does not show reasons behind these behaviors.

Sequence analysis is based on the belief that actions are sequential. One action is the result of the action before it and the reasons for the actions after it. It attempts to examine connections between the actions as they occur. This analysis considers the probabilities that a certain action will follow another specific action and takes into account interaction over time.

For this paper, we decided to use a combination of sequence analysis and frequency analysis in examining student behavior. During the data labeling, our experts used sequence analysis within each clip to determine directionality of the students' sequences of actions, i.e. were they converging to or away from the solution. During the analysis, we tallied the actions and events in conjunction with the features of interest that we derived from the feedback from our experts. This will be further explained in the discussion section.

3. METHODOLOGY

3.1. Aplusix II

Aplusix is an ITS for Algebra. Its academic scope ranges from numerical calculations, manipulation of polynomial equations, solving equations, inequations, and systems. It also gives range up to nine levels of difficulty for each type of exercise. For its usage, it allows the students to solve problems step-by-stem, as they would on paper. Figure 1 shows us a screenshot of the interface. It displays a small example of how to solve the

Figure 1. A sample screenshot of Aplusix.

exercise in steps and the virtual keyboard to provide the user with special symbols.

One of the research-related features of Aplusix is that it logs all user interactions with the system in text files. The text logs used for this experiment were generated from an earlier experiment conducted by Rodrigo et al [6]. Their experiment was conducted with 140 high school students from five different private high schools with ages 12 – 15. Figure 2 shows the raw text version of the section of the log containing the interactions of a student during an exercise. For our research, we only make use of the following attributes in analyzing student behavior:

- Move number – the count of how many actions the user has performed so far.
- Time – the amount of time in seconds that has passed before this action was done.
- Action – the action performed by the user or, in some cases, done by the program.
- Step – Aplusix allows the student to solve each problem using a series of equations called steps. Each step must be equivalent to one another and this is the indicator to which step the action is being done on.
- Expression – This is the state of the equation of the step after the action.
- Status – This is the solution state of the student. It indicates: first, if the current step is equivalent to the previous one and second, if the current step is equal to the answer to the problem.

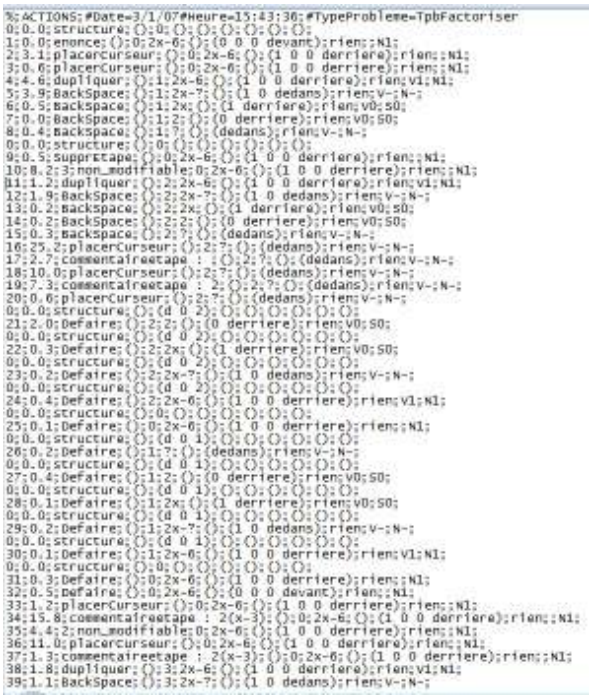

Figure 2. This raw text log was generated by Aplusix.

3.2.Data Distillation from Aplusix

For our data distillation, we parsed each log file into a MySQL database and using a web-based application, we processed the text action replay into a more readable format. The actions were grouped into 20-second clips, similar to the 20-second observation window of previous researches [1] [11]. Figure 3 shows a preprocessed clip as what our experts used in classifying our clips. Each line of the raw text action log represents an action or event and we converted to as plain English as possible. Repeated actions were grouped into a single line and keywords were highlighted for emphasis.

3.3.Sampling and Labeling

For our experiment, we came up with a total population of 11,220 clips of actions and using Slovin's formula, we were able to get a sample of 391 clips to be labeled:

$$S = \frac{N}{1 + N\,(\epsilon^2)} \qquad (1a)$$

We asked of Dr. Cornelia Soto of the Education department and Mrs. Ria Arespacochaga of the High School Math Department, both from the Ateneo de Manila University to help us classify and identify which behavioral patterns will tell us if the student is off-task. Dr. Soto is a former subject area coordinator for the Math department in the Ateneo Grade School and has published numerous books on Mathematics. Mrs. Arespacochaga is a masters graduate for Math education from Singapore. Their knowledge in math education and the off-task behavior made explaining our research to her a simple task before they can be able to categorize whether the clips of the students' actions we will be showing them are on-task or off-task.

3.4.Machine Learning Using WEKA

Our classified clips were then summarized into a feature table as shown on Figure 4. This contains basic information for each clip and the features of different statistical attributes we kept track based on the feedback from our experts. This will be explained further later on. With WEKA, we manually reduced the features by mainly removing the irrelevant columns such as ID, comments, and so on. We used the J48 algorithm supported by WEKA, which gave us an output of a C4.5 decision tree. This tree was then validated using the ten-fold cross validation.

4. DISCUSSION

4.1.Preliminary Findings

As our experts went about classifying our clips, we documented some of the more relevant conversation that occurred during the process. From this, we have picked up some insight on the train of thought our experts used in labeling off-task behavior.

During experimentation, our experts found that one of the biggest discerning points of determining off-task behavior was that if the students' actions correspond to the proper way of finding the solution. If the numbers reasonably resembled a number that was expected as part of the solution to the given problem, they surmise that the student was thinking about the lesson and thus was on-task. Translating this expert intuition to a quantifiable measure was difficult for us. Not only was it difficult to determine whether the student was on the correct path among the many paths in solving the problem, it was also difficult to determine if the student was merely being careless or over-looking simple mistakes they made; in this state, the students are still considered on-task but confused. For these cases, it is not enough to simply detect if the students have the problem partially solved, or had step equivalence, which are what Aplusix can mainly give us as feedback.

4.2.Preliminary Model

After labeling approximately half of our sample clips, we can already come up with a model to test if our experiment will give us satisfactory results.

One of the comments made by Mrs. Arespacochaga during classification was that one of the main factors she used in determining on-task behavior was that if a student pauses at the start of the exercise, the student is regarded as "thinking" but if a student pauses at the end then the student is confused and resulted in being off-task. Figure 5 shows the decision tree generated from the partial results we got from our second expert, Mrs. Arespacochaga. From this, we can see that a majority of clips classified as having on-task behavior resulted from three main attributes:

- The average time of each action across all actions performed is greater than 0.45 seconds.

Move	Step	Events
		The student performs **0 action(s)** prior to this clip.
2	1	The student moves to step 1 and performs: **Duplicate current step**. 9x-(9-(-2x+8)) **The step has equivalence** The problem is **not solved**
3		The student moves the cursor with: **Place cursor** after **3.70 second(s)**.
4		The student begins to delete with: **BackSpace** after **1.30 second(s)** for the next **3 turns** in **1.1 second(s)**. 9x-(92x+8) The step has a **non-equivalence** The problem is **not solved** and has **non-equivalences**
7		The student types: **+** after **1.60 second(s)**. 9x-(9+2x+8) The step has a **non-equivalence** The problem is **not solved** and has **non-equivalences**
8		The student begins to move the cursor with: **Right** after **1.00 second(s)** for the next **3 turns** in **0.3 second(s)**.
11		The student deletes with: **BackSpace** after **0.80 second(s)**. 9x-(9+2x8) The step has a **non-equivalence** The problem is **not solved** and has **non-equivalences**

Figure 3. A sample clip preprocessed for readability.

- The total time of actions before the student becomes inactive for the rest of the 20-second clip is greater than 10.7 seconds.
- The student inputs 6 numbers at most.

The first two features clearly reflect Mrs. Arespacochaga's thought-process of looking at when a student pauses. If a student pauses at the end, it lessens their actions taken within 20 seconds and thus reduces action time and students who pause at the start generally raises the average time across all actions taken.

The third feature however is not a reflection of this. Considering the use of Aplusix, the types of inputs the students have at their disposal compose of number inputs, symbol inputs, use of functions, cursor movements using either the keyboard or mouse, editing keys such as delete, cut, copy, paste, and so on. Since the exercises are generally composed of small numbers with 3 digits or less, it is not unusual if students would only type in 6 numbers or less. Looking at the tree, if the student did type in more than 6 numbers, the problem complexity determines if typing in more numbers within 20 seconds is viable or the student may have ended up doing trial-and-error or just playing around.

4.2.1 Features Used

Based on the feedback we received from our experts, we decided to use the following features for machine learning:

Problem difficulty and complexity: one of the more basic information required by our experts was what type of problem and how difficult the student was trying to solve. This is usually the bases on how "reasonable" were the pauses the student made or the confusion the student is displaying. Problem difficulty alone was not sufficient since more than 80% of the problems were of B1 – Expansion and Simplification. Problem complexity gives a numerical rating on how complicated the original problem looks as to possibly confuse the student.

Starting Turn: clips do not necessarily begin at the start of the exercise and sometimes contain actions that already find the students in the middle of solving problems. In conjunction with the problem difficulty, how reasonable the actions of the students are depends on this feature.

Action Count and time: these are two of the more basic information of the clip and counts how many actions the student did within the clip and the time between the first action to the last action.

Average times: this is the average time of each action across all actions within the clip.

id	sample_id	chunk_id	affect	engagement	comment start	end	elapsed	problem difficulty		problem complexity		starting move	action c
739	3	4	normal on	12:50:50	12:51:03	00:00:13		A1	6	2	10	19.6	2	1.96	1	0
741	3	4203	normal on	12:49:50	12:50:06	00:00:16		A1	13	19	10	19.1	2	1.91	1	0
743	3	4194	confused off	12:50:08	12:50:21	00:00:13		A1	6	6	46	12	18.2	1.66666666667	1.516666	
745	3	4237	confused on	12:59:08	12:59:58	00:00:50		A2	11	11	39	13	15.7	1.53846153846	1.207692	
747	3	4222	confused off	12:51:05	12:51:31	00:00:26		A2	10	10	67	5	18.8	4	3.76	1
789	3	1112	normal on	13:25:10	13:25:25	00:00:15		B1	15	2	3	3	6.66666666667	1	1	
821	3	3263	normal on	13:00:00	13:00:41	00:00:41		B1	7	2	18	18.2	1.11111111111	1.01111111111		
965	3	886	confused on	13:22:05	13:25:08	00:03:03		B1	9	9	441	6	12.7	3.33333333333	2.116666	
1027	3	1689	normal off	13:21:12	13:22:03	00:00:51		B1	33	896	41	15	0.487804878049	0.365853658537		
1031	3	813	confused on	13:09:44	13:10:06	00:00:22		B1	14	304	2	19.9	10	9.95	0	
1047	3	634	unknown off	13:10:08	13:10:30	00:00:22		B1	6	93	4	19.5	5	4.875	0	0
1133	3	2293	confused on	13:07:24	13:08:24	00:01:00		B1	6	8	16	16	18.5	1.25	1.15625	1
1139	3	3391	normal off	13:00:43	13:00:56	00:00:13		B1	6	2	11	0.6	1.81818181818	0.0545454545455		
1217	3	3356	normal unknown	13:10:32	13:11:25	00:00:53		B1	9	57	10	12.7	2	1.27	1	0
1309	3	4310	confused on	13:08:26	13:09:42	00:01:16		B1	12	65	16	14.7	1.25	0.91875	2	

Figure 4. This generated feature table was used for WEKA.

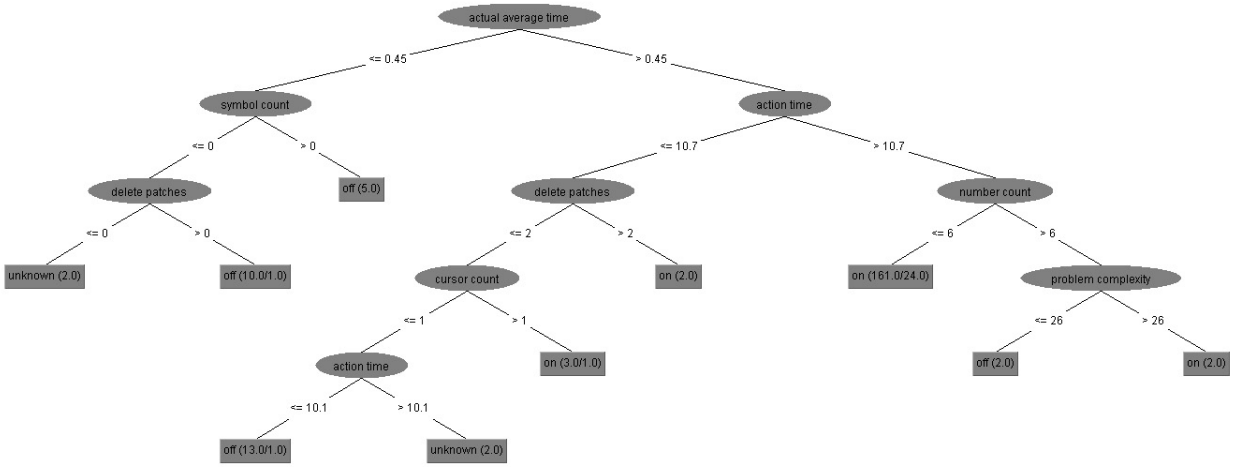

Figure 5. This decision tree represents the thought process of our expert.

Deletion: in keeping track of trial-and-error, we kept track of the deletion activity the students made and the activity of other actions in between deletions. Students who performed trial-and-error would have bursts of deletion with little activity in between bursts.

Activity: activity constitutes the various inputs the student made during the exercise. This includes number inputs, symbol inputs, letter inputs, cursor movement, editing functions such as cut and paste, and miscellaneous functions such as declaration of problem solved.

Status: status constitutes the different the states the students' solutions were during the exercise. We kept track of the number of help requests made, if the solution was abandoned, was the student able to solve it or partly solved it, and if the student came across equivalences in between steps, and finally how many steps the student went through within the time span of the clip.

5. FURTHER CONSIDERATIONS

Even though our preliminary model is only based on at least half of our total sample population, we have deemed it to be sufficient in meeting our expectations since we can compare similarities on how its feature structure compares with our expert's feedback. However, as a partial result there may still have been some unique instances our experts have yet encountered that could greatly alter how the decision-making process is made. Furthermore, the operational determination and extraction of the features may still have room for improvement. As it is, our features are composed mainly of statistics concerning each clip, and these were decided upon according to the feedback received from our experts on how they based their decisions. A possible consideration in changing our features could include keeping track of transitions made in terms of actions taking place after another particular action and so on. As we continue to receive feedback from our experts during the labeling process, we are sure to update the features we want to record from each clip and further develop our model.

6. ACKNOWLEDGMENTS

Support for this project was provided by the Science Education Institute - Department of Science and Technology (SEI-DOST) through the Engineering Research and Development for Technology (ERDT) program. We would like to thank Dr. Cornelia Soto and Mrs. Ria Arespacochaga for their cooperation in this research as our experts. This research undertaking was made possible by the Philippines Department of Science and Technology Engineering Research and Development for Technology Consortium under the project "Multidimensional Analysis of User-Machine Interactions Towards the Development of Models of Affect".

7. REFERENCES

[1] Baker, R.S. Corbett, A.T., Koedinger, K.R., and A.Z. Wagner (2004), "Off-Task Behavior in the Cognitive Tutor Classroom: When Students 'Game the System'", Proceedings of ACM CHI 2004: Computer-Human Interaction 383-390

[2] Baker, R., Corbett, A. T., and A.Z. Wagner (2006), "Human Classification of Low-Fidelity Replays of Student Actions", Proceedings of the Workshop on Educational Data Mining, Jhongli, Taiwan, pp.29-36.

[3] Hulshof, C. D. (2004), "Log File Analysis", Encyclopedia of Social Measurement

[4] Koedinger, K.R., Anderson, J.R., Hadley, .W.H., And M.A. Mark (1997), "Intelligent tutoring goes to school in the big city", International Journal of Artificial Intelligence in Education, 8, 30-43

[5] Murray, Tom (1999), "Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art" International Journal of Artificial Intelligence in Education, 10, 98-129

[6] Rodrigo, Ma. Mercedes T., Ryan S.J.D. Baker, Sidney D'mello, Ma. Celeste T. Gonzalez, Maria C.V. Lagud, Sheryl A.L. Lim, Alexis F. Macapanpan, Sheila, A.M.S.

Pascua, Jerry Q. Santillano, Jessica O. Sugay, Sinath Tep, and Norma J.B. Viehland (2008) "Comparing Learners' Affect While Using an Intelligent Tutoring System and a Simulation Problem Solving Game", Proceedings of the 9th International Conference on Intelligent Tutoring Systems, pp 40-49

[7] Rowe, Jonathan P., McQuiggan, Scott W., Robison, Jennifer L. (2009), and James C. Lester, "Off-Task Behavior in Narrative-Centered Learning Environments"

[8] Zhou, Yujian and Martha W. Evens (1999), "A Practical Student Model in an Intelligent Tutoring System", In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence.* Chicago, IL, 1999, pp. 13-18

[9] Koedinger, K. R. & Anderson, J. R. (1993). Effective use of intelligent software in high school math classrooms. In Proceedings of the *World Conference on Artificial Intelligence in Education*, (pp. 241-248). Charlottesville, VA: Association for the Advancement of Computing in Education.

[10] Rowe, Jonathan P., Mcquiggan, Scott W., Robison, Jennifer L., and James C. Lester, "Off-Task Behavior in Narrative-Centered Learning Environments",

[11] Walonoski, Jason A. and Neil T. Heffernan (2006), "Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems" *Intelligent Tutoring Systems*, Volume 4053/2006, 382-391, June 21, 2006