Design of an Affective Agent for Aplusix

Thor Collin S. Andallaza[1]
tcandallaza@yahoo.com

Rina Joy M. Jimenez[1]
jimenezrjm@yahoo.com

ABSTRACT

Embodied Conversational Agents are computer interfaces that are capable of interacting with human users in a conversational manner. In the area of intelligent tutoring systems, ECAs hold great promise in enhancing the capabilities and performance of learners though simulating functions performed by human tutors such as providing feedback on the student's progress. One of the challenges in ECA design is that of establishing protocols for providing appropriate feedback to students. In this paper, we present two models of these protocols that we intend to use in the design and development of an ECA for Aplusix, an intelligent tutor for algebra.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents.*

Keywords

Affect, Aplusix, data mining, embodied conversational agent, intelligent tutoring systems, learning, log file analysis.

1. INTRODUCTION

An agent is a computer program embedded within a particular environment to achieve certain objectives such as providing assistance to users. It has the capability to perform autonomous action which allows it to perform these objectives [9]. Among the many types of agents, one of these is called an Embodied Conversational Agent (ECA).

ECAs are computer interfaces that are capable of behavior similar to that of humans. They can interact with a human user or with one another as human beings would in typical face-to-face conversations [3]. This behavior is achieved through the implementation of certain properties, such as recognition and generation of both verbal (through audible speech or via a keyboard) and nonverbal (facial expressions and gestures) input and output [3]. This level of interaction with human users provides the potential for ECAs to be used in a digital learning environment, allowing such applications to be able to address not only the cognitive development of students, but also aspects which can only be perceived through a humanlike interaction such as engaging in a conversation. These aspects include the students' moods, feelings, and any changes in behavior. One such learning environment is known as an intelligent tutoring system (ITS).

An ITS is a computer program that makes use of artificial intelligence to provide learners with individualized instruction. It acts as a support for the learner, such that it facilitates the learning process [11]. Previous work [6, 10] have shown that integrating agents into ITSs positively improve the learning experience of students.

Rebolledo-Mendez [10] implemented a motivational agent named Paul for Ecolab, an ITS that taught students topics on Ecology. By using Paul to model motivation and adjust motivational reaction, de-motivated, low, and average students were able to significantly increase their post-test scores. This was done by allowing Paul to motivate students in different ways: to exert more effort, to be more independent, or to be more confident [10].

There were, however, some limitations to the study. One of these was that the results were derived from a very small sample. Another limitation they indicated was that adapting feedback and character's reactions, in conjunction with a quiz, constitute only a first step in the study of motivating techniques in ITSs; thus, general guidelines could be used in order to improve student motivation [10].

In another study, Graesser et al. [6], they developed AutoTutor, an application which simulated the discourse patterns and pedagogical strategies of a typical human tutor. It was designed for college students in introductory computer literacy courses, who learned the fundamentals of hardware, operating systems, and the Internet. AutoTutor worked by initiating a conversation with the student. It appeared as a talking head that acted as a dialogue partner with the learner, who contributed to the conversation via input from the keyboard. One thing notable about the tutor was that it encouraged the learner to articulate answers that were lengthy and required deep reasoning – examples of which included answers to why, how, and what-if questions. There was a multi-turn dialogue involved between AutoTutor and the student, encouraging the student to construct the knowledge and discover what he or she had mastered, rather than bombarding the student with the information to master [6].

Results of the study showed that this strategy of AutoTutor was able to influence learning and mastery of students. Comparing students who used AutoTutor to those who only reread the topic and to the control group who did not reread, AutoTutor was able to help students answer more questions which were used in an actual computer literacy course, garnering a greater score than the two other groups [6].

Using the findings of these previous studies, we wanted to explore the effects of an ECA in another ITS. This ITS was Aplusix (Figure 1), an application that aimed to teach learners arithmetic and algebra [4]. The application allowed learners to solve algebraic expressions step-by-step as if writing with paper and pencil. It covered topics such as factorization, simplification, solving equations and inequalities, among others. Aplusix allowed students to select problem sets of a particular topic, and using its editor, they could solve each problem one step at a time until they were able to arrive at the expected solution. At any time, however, they were also able to exit the program, skip a problem, get hints to the current problem, or be shown the correct solution to the current problem [7].

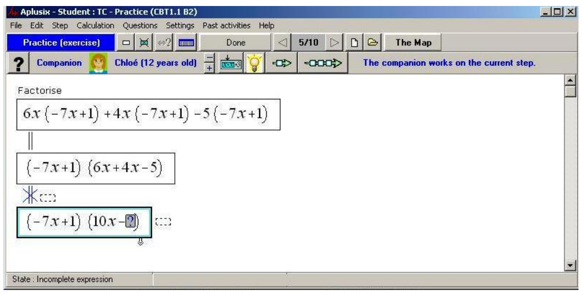


Figure 1. A screenshot of Aplusix.

Within the environment of Aplusix, an ECA could fulfill a number of roles to improve the learning process. One of the roles could be to detect and identify the current state of the student – whether the student was experiencing boredom, confusion, or flow.

Designing and implementing an ECA for an ITS such as Aplusix, however, is a challenge. One of the most difficult phases is designing the appropriate protocols as to when the agent can appropriately respond given different situations. Our research is an implementation of that phase, where we attempt to design these protocols using the interaction logs taken from previous studies with Aplusix. The results of this research will then serve to be a potentially stable stepping stone towards the development of a fully functional agent for algebra.

This research is a continuation of an ongoing research in detecting student affect in order to sustain motivation. Specifically, the research grounds itself on three previous studies [1, 6, 7], each investigating a particular concept, and using Aplusix as their learning environment. These concepts shall be discussed later on in the next section of this paper.

In addition, this research also includes ideas from Cassell et al. [3] regarding the ECA overall architecture. Figure 2 is a visual representation of the architecture and the interaction between components.

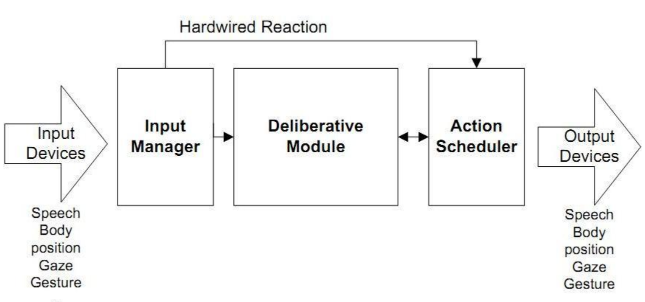


Figure 2. The ECA Overall Architecture [3].

The architecture consists of four major components: the Input Manager, the Hardwired Reactions, the Deliberative Module, and the Action Scheduler. The Input Manager is responsible for acquiring data from various input devices, converting input into forms that can be utilized by the system's other modules, and routing the results to the Deliberative Module [3]. In some cases, the Input Manager may reroute the results into the Action Scheduler via the Hardwired Reactions module in order to reduce system response time. This is true for user input that requires quick reactions, but not any reference to the discourse model [3]. The Deliberative Module serves as the action selector portion of the architecture, where at each moment in time, it will determine the contribution of the agent into the conversation. Finally, the Action Scheduler serves as the motor control of the agent, coordinating output actions at the lowest level [3]. We shall map out which components are addressed by the previous studies in the next portion of this paper.

2. PREVIOUS WORK

In this section, we will discuss previous research on Aplusix and affect which examined the relationships between student behaviors or affective states and learning. For our research, the studies by Lagud [7], Bate [1], and Lim [8] discuss topics that are considered most relevant to what we want to explore.

2.1 Affect and Learning

The study by Lagud [7] dealt with identifying the relationship between the affective and learning profiles of students while interacting with Aplusix. It defined the affective profile as a percentage in time where a student exhibited an affective state (an emotion, feeling, or mood) during an observation period, while the learning profile, which was of particular interest for our study, was based on the number of correct items solved, amount taken to solve each problem, as well as the highest difficulty attempted by the student [7].

The test was done on 140 first and second year high school students from four schools in Metro Manila and one school in Cavite. The students' age ranged from 12 to 15 years old, all of which were computer-literate but had never used Aplusix. They were grouped into 10, and for 42 minutes they used Aplusix under constant observation [7].

The results of the study revealed that students who attempted more difficult problems and took the least amount of steps and time solving them experienced the most flow, defined as a total immersion and focus in an activity [5], while those who attempted problems of lower difficulty and took the most number of steps and time solving experienced boredom and confusion the most. Because the study was able to confirm intuition with quantitative data [7], its findings could aid in identifying the affect of students who performed poorly while using Aplusix, and thus aid in the design of an agent through formulating appropriate cognitive and affective intervention.

There were some limitations to this study, however. For one, the profiles generated by the study were taken at a student and session level. In order for the profiles to be more useful and make the ECA implementation more effective, our research will attempt to generate finer-grained profiles through a different analysis approach, making the profiles much feasible for use in real time analysis of student interaction with the ITS.

2.2 Detecting Off-task Behavior

Another study by Bate [1] explored the possibility of automatically identifying off-task behavior when using an intelligent tutoring system such as Aplusix. The study aimed to achieve this objective through the analysis of Aplusix logs taken from previous studies, grouped them into twenty-second clips, and were then labeled by two experts as either on-task or off-task. The reasoning formulated from the expert labeling then became the basis for the machine learning in identifying off-task behavior through the use of a program called Wakaito Environment for Knowledge Analysis (WEKA).

Based on the results of the study, the features which were used to generate the model included problem difficulty, starting turn, action count and time, deletion, keyboard inputs and interaction, solution status, and progression [1]. These were the features used to generate two separate models for detecting off-task behavior, each one according to the reasoning of one of the experts. The study was not able to combine the models, however, due to the low agreement between the two experts [1].

2.3 ECA Design

Finally, a study by Lim [8] sought to find out what considerations were needed in designing a motivational pedagogical agent, where 'pedagogical' referred to the agent being integrated with an intelligent tutoring system or any computer-aided learning environment to facilitate learning [8]. It explored two different approaches to motivation, namely a motivational agent that was empathic and encouraging, and a motivational agent that was capable of collaboration apart from just empathic concern, thus was able to sustain a student through a period of "Stuck", which was identified in study conducted by Burleson [2] as the opposite of flow – a feeling of being out of control or with lack of concentration due to a sense of failure.

In order to answer the research questions indicated, Lim introduced Dante, a pedagogical agent created for the study. Dante was implemented on a "Wizard-of-Oz" mechanism, where a human operator observed end-user interactions with the system, and controlled the system to provide output when necessary. Two implementations of Dante were done to reflect the two different approaches discussed above, and both implementations were tested with a group of high school students while using Aplusix. Three groups were formed for the experiment: a control group having no agent, an experimental group with an empathic agent, and another experimental group with a collaborative and empathic agent. Data was gathered though the use of assessment forms [8].

Unfortunately, upon analysis of the results of the study, the experiment was not able to show a significant difference in the preference of students for a collaborative and empathic agent as compared to one that relies on an empathic approach alone. However, the study also concluded that there was a potential for motivational agents to develop more persistence in students when attempting to solve problems in an intelligent tutoring system.

The study concluded its discussion with recommending future work, such as including different looks for the agent, the implementation of an automated means of evaluating the student rather than a "Wizard-of-Oz" form of interaction, as well as quality feedback returned by the agent to the student [8]. The main limitation of Lim's study, however, was that the agent and its logic were never integrated.

2.4 Summary of Previous Work

In relation to the ECA architecture of Cassell et al. [3], the studies of Lagud [7] and Bate [1] mainly aided in the construction of the Input Manager and the Deliberative Module. Both studies dealt with providing intelligence for the agent by presenting key features within interaction logs which were indicative of a learner's current state – in flow or bored, on-task or off-task. In the same way, the studies also provided a foundation for automatically retrieving relevant input from Aplusix, which can then be processed into usable information – the primary task of the Input Manager.

The output of Lim's [8] study, however, was a split between the Deliberative Module and the Action Scheduler. For the Deliberative Module, the agent script which was generated in the study provided some form of logic for the agent, where each response corresponded to a particular facial expression on the part of the learner and the agent. The timing and synchronicity of the delivery, on the other hand, was a contribution to the creation of an Action Scheduler.

Given that all three studies provided foundations for the three major components of the architecture, synthesizing all of these into one output, therefore, may already constitute a fully-functional agent for Aplusix. For the purposes of this paper, however, we only discussed the preliminary work that was already done towards this goal, namely the generation of student models that will aid the agent in evaluating the affective state of a student when interacting with Aplusix. Further research needed on the other aspects not addressed was explained in the latter parts of this paper.

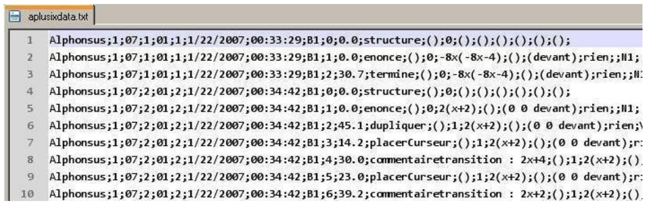
3. METHODOLOGY

As mentioned in the literature review of Lagud [7] study, one of the main limitations of the profiles that they constructed was that the grain size was coarse, particularly due to taking the profile at the student and session level.

In our research, we revisited the interaction logs generated by Aplusix during the previous study and approached the analysis and generation of the learner models in another perspective. Instead of the student and session groupings done in the previous study, the logs were grouped according to the difficulty levels set by Aplusix (e.g. B1, B2, A1, etc.), and based on the population size of each grouping, only those that were of ample size, with at least 30 entries, were selected. After which, we began analysis through the use of two distinct methods, both of which were utilized in Lagud's [7] study: standard deviations and terciles. Using these two methods, we generated the revised student models which the agent will use as a component of its intelligence.

3.1 Log Parsing

The Aplusix interaction logs used for this analysis were compiled into a single text file, each line containing a log of a specific student action. Figure 3 shows a screenshot of the log, while Table 1 describes the content of each line, as explained in Lagud's [7] study.



aplusixdata.txt

```
1  Alphonsus;1;07;1;01;1;1/22/2007;00:33:29;B1;0;0.0;structure;();0;();();();();();();
2  Alphonsus;1;07;1;01;1;1/22/2007;00:33:29;B1;1;0.0;enonce;();0;-8x(-8x-4);();(devant);rien;;N1;
3  Alphonsus;1;07;1;01;1;1/22/2007;00:33:29;B1;2;30.7;termine;();0;-8x(-8x-4);();(devant);rien;;N
4  Alphonsus;1;07;2;01;2;1/22/2007;00:34:42;B1;0;0.0;structure;();0;();();();();();();
5  Alphonsus;1;07;2;01;2;1/22/2007;00:34:42;B1;1;0.0;enonce;();0;2(x+2);();(0 0 devant);rien;;N1;
6  Alphonsus;1;07;2;01;2;1/22/2007;00:34:42;B1;2;45.1;dupliquer;();1;2(x+2);();(0 0 devant);rien;
7  Alphonsus;1;07;2;01;2;1/22/2007;00:34:42;B1;3;14.2;placerCurseur;();1;2(x+2);();(0 0 devant);r
8  Alphonsus;1;07;2;01;2;1/22/2007;00:34:42;B1;4;30.0;commentairetransition : 2x+4;();1;2(x+2);()
9  Alphonsus;1;07;2;01;2;1/22/2007;00:34:42;B1;5;23.0;placerCurseur;();1;2(x+2);();(0 0 devant);r
10 Alphonsus;1;07;2;01;2;1/22/2007;00:34:42;B1;6;39.2;commentairetransition : 2x+2;();1;2(x+2);()
```

Figure 3. A screenshot of the log compilation file.

Table 1. Description of Log Content [7].

Data	Description
1	School of student
2	Run number
3	Student number
4	Set number
5	Problem number within set
6	Absolute problem number – problem number relative to all problems answered by the student
7	Date of attempt
8	Time started
9	Problem level
10	Step number
11	Duration of step
12	Action performed
13	Error committed
14	Etape (Fr.) – stage or phase of solution
15	Current state of the mathematical expression
16	Etat (Fr.) – current state or condition of solution
17	Location of cursor
18	Selected values within the solution
19	Equivalence – indicates if the equation is correct or not
20	Resolution – indicates if the problem has been solved or not

In order to efficiently obtain and process relevant information from these logs, we decided to create a program using Java to automatically read from the text file and extract the current step count, the current duration taken, the types of action done, the type of equivalence observed, and the type of resolution observed.

Once the program identified that the incoming log was of another problem attempt by a student, it took all the information obtained from the previous lines and stored them in a DataEntry object, which was a profile of an attempt of a student at a particular problem. The DataEntry object was then sorted into DataEntryArray objects, where each was a set of DataEntry objects of a particular problem type. The program continued this process until it reached the last log given in the text file, where it then returned the total number of entries stored per problem type. The program then enabled the user to print the summarized profile into individual text files, which contained the profiles of each problem attempt done for that specific problem type.

3.2 Log Analysis

The text files of the selected problem types were compiled into a Microsoft Excel file for the analysis of the profiles. We divided the users into groups using two different criteria based on the study by Lagud [7]: standard deviations and using terciles. In both methods, we focused on only two properties of each attempt profile – number of steps and duration, again based on the study done by Lagud [7].

The standard deviation method required us to compute for the means and standard deviations of the steps and the duration for each problem type. The resulting values from the calculations became the basis of the learner models of each problem type using this method.

On the other hand, the terciles method of analysis required that, for each problem type, the sample be divided almost equally into three groups – the first tercile (which contained smaller values), the second tercile, and the third tercile (which contained the larger values). Again, using the values for the number of steps and duration, we obtained for each of the three groups a minimum, maximum, and the mean value.

In order to get the maximum for each problem type, this was done by first dividing the population size by 3, and with the resulting value x, we obtained the x-th least value as the maximum of the first tercile. The second tercile's maximum value therefore was the $2x$-th smallest value, and the largest value in the sample became the maximum of the third tercile.

On the other hand, the minimum values were as follows: the minimum of the sample is the minimum of the first tercile. For the second tercile, we obtained the y+1-th smallest value of the sample, where y was the population size of the first tercile. Finally, the y+z+1-th smallest value was the minimum of the third tercile, where y was the population size of the first, and z was the population of the second. From this information, we generated another version of the learner models per problem type.

4. DATA ANALYSIS AND DISCUSSION

This section discusses the resulting learner models that were generated from the Aplusix interaction logs from Lagud's [7] study, using a per-problem type analysis. We will first describe results of the parsing of the Java program created for this research. Afterwards, we will present the analysis done to generate the logs. Finally, we will present the learner models resulting from the analysis.

4.1 Log Parsing Results

Based on the resulting output of the program, problem type B1 contained the most number of attempts, resulting in 1366 entries, while problem types A4 and F5 were only attempted once. As stated in our Methodology, we only selected problem types that contained at least 30 entries to ensure an ample population size for the analysis portion of the study. The problem types selected for the analysis were A1 (52 entries), B1 (1366 entries), B2 (790 entries), B3 (174 entries), C1 (186 entries), and C2 (46 entries).

4.2 Log Analysis Results

The log analysis using standard deviation and terciles revealed different forms of the models which can be used by the agent to evaluate the current affective state of the student.

4.2.1 Standard Deviation (SD)

For the standard deviation method, the idea was as follows: when a student attempts to solve a particular problem type, the agent will load the model for that problem type and compare the resulting values from the attempt of the student. The ideal range for the attempt value x is therefore $m-s<x<m+s$, where m is the mean value, and s is the standard deviation value. Any value for x which is out of this range will trigger the agent to fire an intervention.

However, the results of this test revealed that the values for the standard deviations turned out to be very large – some of which were larger than their corresponding mean values. What this meant, therefore, was that the values for each problem type consisted of a wide range of values, most of which were far from the mean. Table 2 and 3 below shows the resulting models from the number of steps and duration respectively, along with the percentage of the population which were either above or below one standard deviation.

Table 2. Learner Models (SD method, Number of Steps)

Problem Type	Mean	Standard Deviation	% of the Population above or below 1 SD
A1	18	23	9.62
B1	74	88	10.25
B2	75	64	15.82
B3	80	96	12.64
C1	18	10	19.35
C2	44	42	10.87

Table 3. Learner Models (SD method, Duration (in seconds))

Problem Type	Mean	Standard Deviation	% of the Population above or below 1 SD
A1	37.24	59.14	9.62
B1	195.55	1344.74	0.44
B2	114.33	114.29	10.13
B3	163.79	208.57	10.92
C1	26.61	22.73	15.59
C2	82.04	90.78	16.67

4.2.2 Terciles

The tercile method for analysis, on the other hand, was another design that the agent could use to evaluate the current affective state of the student. The idea consisted of using the same step and duration values generated by the student when attempting a particular problem type. This time, however, it will determine if it falls under a particular category – the first, second, or third tercile – using the values obtained from the analysis as thresholds for these categories. The agent fires an intervention each time it identifies that the student is moving from one category onto another.

Table 4. Learner Models (Terciles method, Number of Steps)

Problem Type	Tercile Size	Min	Max	Mean
A1	19	3	9	6
	16	10	13	11
	17	15	143	37
B1	468	3	34	22
	446	35	66	49
	452	67	1087	153
B2	267	3	43	27
	259	44	78	58
	264	79	588	139
B3	58	3	32	12
	58	33	72	50
	58	76	543	178
C1	67	3	14	11
	60	15	18	16
	59	19	83	28
C2	15	3	23	15
	17	25	39	32
	14	41	212	90

Table 5. Learner Models (Terciles method, Duration (in seconds))

Problem Type	Tercile Size	Min	Max	Mean
A1	17	2.2	9.5	6.588
	17	9.6	24.2	16.718
	18	24.4	310.3	85.572
B1	455	0.2	56.9	35.482
	455	57.1	130.1	84.698
	456	130.7	49172.6	465.902
B2	263	0.1	57.4	35.801
	263	59.6	116.7	84.186
	264	137.1	1354.6	222.595
B3	58	0.1	59.1	25.388
	58	59.6	129.7	93.848
	58	137.1	1275.2	372.147
C1	62	0.5	15.6	11.715
	62	15.7	25.4	19.203
	62	25.4	154.8	48.916
C2	15	1.7	31.2	19.32
	15	32.4	60	43.08
	16	63.7	377	177.369

From the results of the analysis, the population of each tercile ended up to be almost equal in size as was intended, most especially when we analyzed the logs using duration. Table 4 and 5 presents the values from the models generated.

4.2.3 Interpretation of Results

The findings from Lagud's [7] study indicated that students who experienced flow required the least amount of steps or time to solve a problem, while those who took the most steps and time at an attempt experienced the most boredom and confusion. Following this logic, the attempts which fell under the above average group (m-s for standard deviation, first tercile for terciles) could generally indicate that the students experienced flow during that particular attempt. Likewise, those who fell under what could be called the below average group ($m+s$ for standard deviation, third tercile for terciles) could generally be an indication that the students experienced boredom and confusion the most during those attempts. It was these indications which could be used by the agent to fire appropriate interventions, most especially whenever it identified that a student at a particular attempt at a particular problem exhibited trends which fell under the below average group.

Extreme values that fell under the above average group, however, might also indicate boredom and confusion. Examples of these were students who took less than three steps and only a fraction of a second at a particular problem, which most likely indicated that the student gave up on the problem early in the attempt.

5. FURTHER STUDIES

From the analysis done on the interaction logs to generate the student models, we now have a starting point in developing the intelligence of an affective agent for Aplusix. We used the interaction logs to determine patterns which can be understood by the agent in order to identify a change in the student's current affective state, and when necessary, fire an intervention upon recognition of such an event.

In our ongoing work, we will integrate this into an implementation of the agent and test these models in order to observe how the models affect the both the decision-making ability of the agent, as well as its impact on the students who receive the agent's interventions. We will further analyze the logs, investigating other factors such as which keystrokes may automatically trigger an intervention regardless of number of steps and duration, and the like.

While this study was able to show when the agent can appropriately intervene based on the models generated, how the agent intervenes was not discussed in the course of this paper. Thus, another ongoing research related to this study is determining the appropriate responses of the agent to students when such interventions are necessary. These will be done by using the agent script generated by Lim [8], as well as additional responses from another research regarding person-to-person tutoring. In that study, we shall observe video clips of human tutors with their tutees, observing which particular events trigger the human tutor's interventions, what they say to the tutee in the intervention, and their facial expression as they deliver it. In addition, further research will also be done on creating the physical look of the agent, using the design of Dante, the agent created by Lim [8], as reference.

6. ACKNOWLEDGMENTS

We would like to thank Dr. Ma. Mercedes Rodrigo for her valuable input and guidance throughout this research. We also extend our appreciation to Ms. Jessica Sugay and the Ateneo Laboratory for the Learning Sciences for their continuing support and unique contributions to the study. In the same way, we are grateful to the Department of Science and Technology Philippine Council for Industry, Energy and Emerging Technology Research and Development (PCIEERD) for the grant entitled "Development of Affect-Sensitive Interfaces" which provides the funds that make this research possible. Finally, we would like to thank Mr. Allan Edgar Bate, Ms. Maria Carminda Lagud, and Ms. Sheryl Ann Lim whose studies served as the foundation of this research.

7. REFERENCES

[1] Bate, A.E. 2010. *Automatic detection of student off-task behavior while using an intelligent tutor for algebra*. Masteral Thesis. Ateneo de Manila University.

[2] Burleson, W., and Picard, R. 2004. Affective agents: Sustaining motivation to learn through failure and a state of stuck. In *Proceedings of the Workshop on Social and Emotional Intelligence in Learning Environments in conjunction with the 7th International Conference on Intelligent Tutoring Systems* (Maceio-Alagoas, Brasil, August 31, 2004).

[3] Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsson, H., and Yan, H. 2000. Human conversation as a systems framework: Designing embodied conversational agents. In *Embodied Conversational Agents*. The MIT Press, USA.

[4] Chaachoua, H., Nicaud, J., Bronner, A., and Bouhineau, D. 2004. APLUSIX, a learning environment for algebra, actual use and benefits. *ICME 10: 10th International Congress on Mathematics Education* (Copenhagen, Denmark, July 4 - 11, 2004).

[5] Csikszentmihalyi, M. 1990. *Flow: The psychology of the optimal experience*. Harper & Row, NY.

[6] Graesser, A., Chipman, P., King, B., McDaniel, B., and D'Mello, S. Emotions and learning with autotutor. *Proceedings of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts that Work* (2007).

[7] Lagud, M.C. 2010. *The affective and learning profiles of students while using and intelligent tutoring system for algebra*. Masteral Thesis. Ateneo de Manila University.

[8] Lim, S.A. 2010. *Towards a framework for developing motivational agents in intelligent tutoring systems*. Masteral Thesis. Ateneo de Manila University.

[9] Padgham, L., and Winikoff, M. 2004. *Developing intelligent agent systems.* Wiley, UK.

[10] Rebolledo-Mendez, G., du Boulay, B., and Luckin, R. Motivaing the learner: An empirical evaluation. *Intelligent Tutoring Systems* (2006).

[11] Rosi, M. Stankov, S., and Glavinic, V. Intelligent tutoring systems for asynchronous distance education. In *Proceedings of the 10th Mediterranean Electrotechnical Conference* (Lemesos, Cyprus, May 29 - 31, 2000).

8. AUTHORS' INSTITUTIONAL AFFILIATIONS

[1] Ateneo Laboratory for the Learning Sciences, Department of Information Systems and Computer Science, Ateneo de Manila University, Loyola Heights, Quezon City, Philippines, +63 (2) 426-6001 loc 5666, http://penoy.admu.edu.ph/~alls