Affective Learning Companions: Increasing Affective State to Promote Long Term Learning

Salman Ahmad • Temitope Akinwade • Elizabeth Cohlman
Matthew Jones • Nitesh Mehta • Peter Woods
Advisor: Winslow Burleson

3 May, 2008

Table of Contents

Affective Learning Companions: Increasing Affective State to Promote Long Term Learning

Salman Ahmad Temitope Akinwande
Elizabeth Cohlman Matthew Jones
Nitesh Mehta Peter Woods

Advisor- Winslow Burleson
Assistant Professor of Human Computer Interaction
Arizona State University

Department of Computer Science and Engineering: Arizona State University

CSE 423 Capstone Project Final Report
Spring 2008
Instructor: Dr. Yinong Chen

ALC@asu.edu

Report Submission Date: May 3rd, 2008

Abstract

"Emotion is fundamental to learning: students with high intrinsic motivation often outperform students with low motivation." [1] There have been many studies demonstrating the link between affective and cognitive learning, but this link is often ignored in our, "one-size fits all," education system. This paper introduces a computational agent that can assist educational professionals with achieving a balance between a student's affective state and cognitive learning; an affective learning companion. The affective learning companion will ascertain a user's emotions and behave appropriately. Using research in emotion recognition, effective pedagogical feedback, and optimization, the companion will respond to the user in a way that improves the user's affective state in the short-term, and improves learning in the long-term. Using a platform consisting of three sensors (skin conductance glove, pressure sensitive mouse, and posture sensing chair) and facial recognition software, the system will be able to analyze the data streams and determine the user's current affective state, such as boredom, fatigue, self-confidence, motivation, etc.. Once the user's affective state is recognized, an on screen agent will respond in a manner similar to that of a real human companion. This project has enhanced the ability of current learning systems and provides an alternative way to represent material. This project has also advanced research in human centered computing (the field concerned with computing and how it relates to the experience of being human). The affective learning companion will add to the statistical and computational techniques available to estimate the structure that governs human behavior. This paper addresses the first stage in development of the affective learning companion. Our goals of developing the core of an affective learning companion have been reached, and further development will continue.

1. Introduction

"I can't do this". "I am not good at this". These are comments typically heard by students trying to learn, and often correspond with affective states of confusion, frustration, and hopelessness. Our education system has overemphasized conveying a great deal of information and facts, in a form that forgets about the natural learning process. These steps include making mistakes, recovering from them, figuring out what went wrong, and beginning again. "Learning naturally involves failure and a host of associated affective responses." [2] A skilled teacher can recognize various states of emotion in students and address these needs appropriately. The goal of this project is to give computers similar abilities. An affective learning companion will be able to assess a student's current affective state, and respond as an expert teacher would.

By re-engaging the bored student, encouraging the frustrated student, or just enhancing the student's natural desire of exploration, the affective learning companion seeks to improve a student's affective state in the short term, and ultimately increase the student's learning in the long term. Using emotion recognition, the system measures a student's emotion and state of engagement. In order to accomplish this, we have used a platform that consists of three sensors (a posture chair, a pressure sensitive mouse, and a skin conductance glove), and facial recognition software. Information from these data streams are combined and interpreted together. Using existing psychological research on how a person physically responds during certain affective states, we can determine within a reasonable range, the user's current state. The onscreen character will respond to the user in a manner consistent with human behavior, displaying its own emotion. Depending on the context, the character can also offer hints or suggestions, or show related examples.

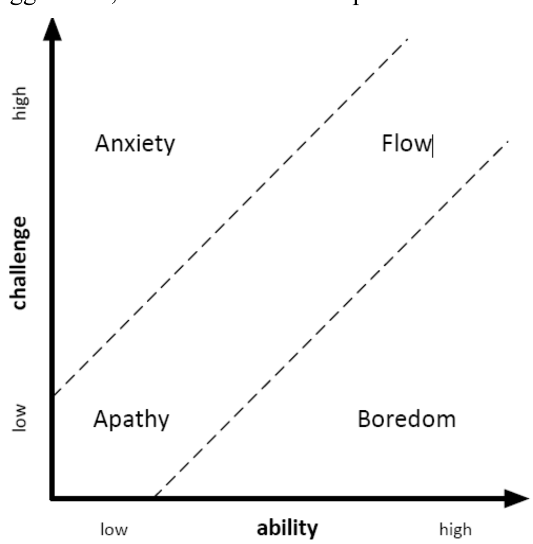


Figure 1.1: Flow experience described in terms of ability and challenge

During the 20th century, a psychologist by the name of Mihály Csíkszentmihályi brought about the idea of a flow experience. A flow experience occurs when an individual is completely engaged and absorbed in a given activity. It is a time of great satisfaction and results in a rewarding sense of accomplishment. A flow experience is often described in relation to ability and difficulty of a challenge (**Figure 1.1**). In relation to learning systems, most attempt to engage the user by moving along the vertical axis and lowering the challenge to meet the user's ability. In doing this, a user is not really learning, but just demonstrating what they already know. In order to truly learn, a user needs to persist longer on higher challenges. This is what the affective learning companion seeks to achieve by enhancing the learner's perception of their own ability.

This paper discusses studies that have been done on affective learning, and what computerized agents people respond best to. We will then discuss the hardware and software components that went into implementing the core of an affective learning companion. This will followed up by a discussion of how the system was tested, and recommendations for future development.

2. Related Work

Great efforts have been made in comparing affective learning and cognitive learning. Most of the researchers involved with the efforts recognize that affective learning is just as important as cognitive learning even though it has been relegated to the background for such a long time.

As R.W. Picard states in his article "Affective learning – a manifesto", cognitive learning, as opposed to affective learning, has been greatly overemphasized in the last half-century. Kaiser and Oertel elaborate that overall, today's computer systems still do not consider the emotions of the users a factor. It is important to combine affective and cognitive learning to produce meaningful classroom learning [3]. Stancato and Hamachek further state that although many researchers have tried to draw a distinct line between affective learning (feeling and attitudes students have for themselves) and cognitive learning (mental functions involved in thinking and understanding subject matter content), there nevertheless exists a link between a student's academic performance and their self image. The reason this link is not clearly seen, and ultimately a reason that affect is lagging behind cognizance, is its difficulty to measure a user's affective state.[4]

A lot of researchers have delved into the measurement of affect and this project incorporates a lot of ideas from those various researches. The widely accepted method of recording affect, namely questionnaires, is unreliable because it interrupts the process. It also requires the subject to be aware of his/her emotional state, and is almost wholly unreliable in the case of children.[4] A suggestion by Picard in "Affective learning –a manifesto", is that several points of references should be used in predicting affect and emotional response rather than relying on these questionnaires. Examples of these points of references are facial recognition, posture sensors, and pressure sensors. In the article "Evaluating affective interactions: Alternatives to asking what users feel", Picard and Daily detail two categories for evaluating affect body measures (changes in muscle activity), and task measures (how someone goes about solving a problem). They proposed that by also using task measures, which measure how a person reacts to stimuli in the test itself, measurements can be taken without having the user directly describe their experiences, and without requiring special sensors or sophisticated analysis. Furthermore, these measurements can be taken at the same time the user is interacting with a system, thus alleviating the need to interrupt the user or wait until after the interaction completes to evaluate their emotional response. Many affective learning agents have incorporated sensors, body measurement, and task measurements. Two examples are illustrated next.

In the article "Affective Learning Companions" by Winslow Burleson, an agent similar to our end goal is described. This was work done by Winslow Burleson while he was at MIT, which is a precursor to our project. Based on the hypothesis that a Learning Companion Architecture with the ability to sense the child's affect will have a greater impact toward learning than one that lacks this ability [1], the learning companion incorporates affective sensors such as cameras for facial expression, eye gaze detection, seat pressure pads to detect posture, galvanic skin response, and the state of the game. The intelligent tutor was used to help students in the tower of Hanoi game and thus they had to use the state of the game as a deciding factor on the performance of the student.

In the article "An affective agent-based virtual character for learning environments", by Boff, Reategui, and Viccari, another agent is presented. The agent architecture presented is multi-agent. It incorporates four different agents into the final design of the student agents. The four agents (social agent, mining agent, recommender agent, and the virtual character) all have varying functions in the total system.

The social agent is responsible for collecting data about the students' interactions and to provide such information for the mining agent. They classified the information collected by the social agent into six categories: Social Profile; Acceptance Degree; Sociability Degree; Mood State; Tutorial Degree; and Performance. Details of all these categories can be found in the main article. The mining agent is responsible for extracting profiles from the data collected and to store them in a knowledge representation mechanism called item descriptors. These descriptors store information about how the social-affective characteristics may affect a student tutoring ability. This is needed because the recommender agent's main purpose is to recommend student tutors to other students that request help from the system. The virtual character agent's goal is communicating with the user through a natural language mechanism, identifying when to recommend a tutor to a student needing assistance and triggering the recommendation process.

The architecture described is only possible and useful when there is a network of computers in a classroom. However, when the user of the learning companion is a single user, this architecture fails. Our project solves this problem because each companion is tailor made for the user and will be used individually. Furthermore, a database will be incorporated into the final design so that user interactions are stored and thus the agent gets better at predicting emotional state of the user over time. Another flaw of the design described above is that there is no sensor implemented. As other researchers have shown, using sensors can help with the accurate prediction of a user's emotional state and thus help the agent respond accordingly. Our project solves this because we will incorporate sensors into our design and thus predict the user's emotional state, and respond accordingly to what the user might be feeling at a particular time instance. For additional information on each of the articles mentioned above, refer to *Appendix A*.

3. Design and Implementation

This section contains an overview of the Affective Learning Companion system along with specific design and implementation details.

At a very abstract level, our system utilizes several sensors that are able to measure physical responses from the user. These sensors include a facial recognition camera, skin conductance glove, pressure mouse, and posture chair. Each of these elements is composed of one to many Piezo Film sensors. These sensors detect a change in voltage differential. The technical manual for these sensors can be found in *Appendix B*. The sensor data is sent to an application, known as the system server, through several small sensor boards called Arduinos. (Refer to *Appendix B* for data sheets on the Arduino USB board.) The system server receives the Arduino data via a serial port (which is emulated within the USB connection) and then executes software that is able to interpret and correlate the sensor data with human emotions. How the system does this is represented in Table **3.1** on the next page. After characterizing the user's behavior, the system server determines the best way to respond to the user and controls the agent accordingly. The manner in which the agent should react is still under development, and will leverage a variety of psychological theories along with data collected from study groups in the next phase of development. This general flow can be seen in **Figure 3.1** on the next page. Ultimately, this system will be deployed in an education environment ranging from an elementary school to a higher education facility.

The system server is the heart of our system. It is written in C and is completely multithreaded. Every Arduino board has a corresponding thread that reads the output generated by one or more sensors attached to the board. The system server is also responsible for launching and controlling the on-screen agent by using standard operating system calls. Since we realized that there may be features that need to be implemented in later versions of the system, we designed the system server to allow for customization down the road. This has been accomplished through the use of function pointers, macros, and makefiles. Essentially, if custom logic is needed, the developer just needs to add a new source file containing the new logic, create a function pointer to the new function using macros in the main program, and recompile using the makefile.

When creating the system, we had a few different agents from which to select. Originally, we debated between using a virtual agent (a computer character displayed on screen), or a physical agent (a small robot or toy that can physically move). The physical agent can be helpful in certain specific applications: for example, if the Affective Learning System is used as a motivational tool for exercise, having a physical agent that gets up and moves around may be more effective. However, the cost of a physical agent, lack of mobility, and difficulty of remote maintenance—coupled with the fact that our system is designed for academic education and not physical activity—caused us to favor a virtual agent. A more in-depth analysis of this decision making process can be seen in the **Modeling, Testing, and Quantitative Evaluation** section.

Table 3.1: This table shows how the different sensors can be used in correlation with human emotions

Sensor	*Description*
Skin Conductance	The skin conductance sensor will test the electrical conductance of the user's skin. This value will change in correlation to the user's state of arousal and hence can be used in conjunction with other sensors to determine the emotional state of the user.
Posture Chair	The posture chair has a variety of sensors to determine how the user is sitting. This can give important insights into how he is feeling. For example, if a user is relaxed, he may be reclining frequently. On the other hand, if he is nervous, he may be rocking back and forth.
Pressure Mouse	The pressure mouse will measure how much force the user is exerting on the mouse. This again can be correlated with certain emotions. For example, if he is angry, the user may apply a lot of pressure on the mouse.
Facial Recognition Camera	Coupled with facial recognition software provided by Google, the facial recognition camera will analyze the user's facial expression and infer his emotional state within ranges of confidence.

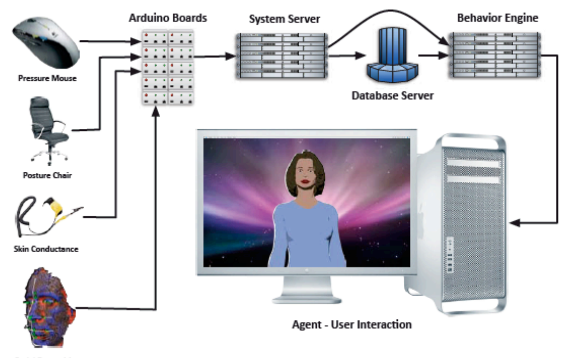


Figure 3.1: Overview of ALC

When choosing a virtual agent, the characteristics that we were interested in were the agent's emotional expressibility, the level of programmatic control, its ability to be expanded in the future, and support from its creators. There were two main virtual agents that were considered: Tim Bickmore's agent and Ken Perlin's agent. Bickmore's agent featured an aesthetically pleasing, full-bodied avatar that was part of a well refined system with many features, such as text to speech capabilities and a powerful scripting engine. Unfortunately, we did not have access to the source code, which limited our future expandability and ruled out the ability to customize the agent to our specific needs. Furthermore, we were limited by the scripting engine's limited set of built-in commands and did not have as much control over the avatar as we desired. This became problematic; we soon found that there were many emotions that could not be expressed. This limited emotional expressibility and expandability led to our rejection of Bickmore's agent. Instead, we decided to go with the agent created by Ken Perlin. Perlin's agent was not nearly as refined as Bickmore's (lacking a proper scripting engine and a full-bodied character); rather, it was just a face that had to be controlled from the keyboard. However, the emotional expressibility of this agent was impressive, as every aspect of the face is customizable allowing it to express the most subtle of emotions. Furthermore, Perlin was eager to provide support and disclosed the full source code, allowing us to expand the agent in the future. Ultimately, we realized that our access to the source code coupled with Perlin's support would enable us to overcome any shortcomings the agent may have in terms of missing features. All in all, because it met all of our design requirements, we selected Perlin's agent for our design.

When we first started using the agent, we had to add many features before we were able to integrate it with our system. Initially, the only way to control the face was through the keyboard. To circumvent this, we modified the source code and added a multithreaded API software layer. (Refer to *Appendix C. for the face API User's Guide.)* The API implements a set of basic behavior responses, such as sad or frightened, and can express different degrees of emotion, such as perturbed or furious. To implement the API, the program continually monitors its standard input stream for commands, which are passed in as strings, and triggers the corresponding response. Using this architecture, the system server can easily control the agent by simply writing to the agent's standard input.

Since our system will be deployed in multiple locations while it is still under development, a way to manage it remotely was desired. We plan to deploy the Affective Learning Companion system in three locations: the Exploratorium Museum in San Francisco, the Media Lab at MIT, as well as our development system at Arizona State University. To enable communication with these remote locations, our system uses a technology called .Mac, which is offered by Apple for use with Macintosh computers. With .Mac, we are able to have complete control of our remote systems from our development system at ASU. Through the file and screen sharing services provided by .Mac, we will be able to transfer files to our remote systems and perform any necessary configuration.

A breakdown of the hardware and software components of our system is shown in **Figure 3.2** below. As the figure shows, the main complexity in our system's implementation lies in its software, most notably in the system server and virtual agent.

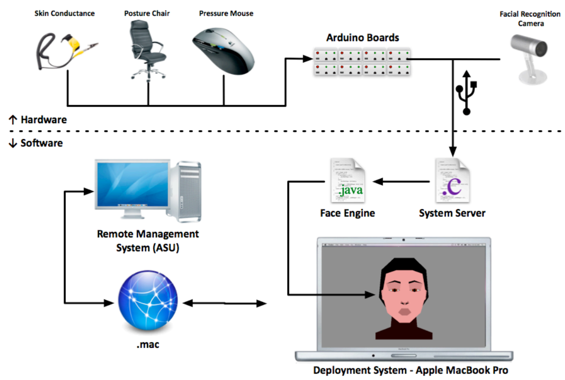


Figure 3.2: Hardware-Software breakdown of ALC system

4. Modeling, Testing, and Quantitative Evaluation

4.1 Evaluation Objectives

There were a couple of factors that went into consideration when evaluating our options for the hardware and software components. As per our design, we are going to be deploying our final product on a Mac Book pro. There are various tradeoffs that we needed to look at when making this decision. The Mac Book pro we were provided with has the following hardware specifications:

- 2.5 GHz Intel Core 2 Duo processor
- 230 GB Hard Drive
- 2 GB Ram
- 512 MB Graphics Card
- Built- in I-Sight Camera

We also needed to have text-to-speech capabilities. During out research, we found that the Mac Book Pro has a built-in text-to-speech program, but it is also much more realistic sounding then the alternative program provided by Microsoft. Since we are going to have three systems, one here at Arizona State University for development, one at the University of Massachusetts Amherst, and one at the Exploratorium Museum in San Francisco, we needed to have a way to remotely update any upgrades made. Apple provides software, .MAC, which enables us to manipulate the offsite systems as if they were on our own network. The cost of the Mac Book Pro certainly outweighs that of a PC laptop, but the performance and dependability tradeoffs make the expense worth it. The Mac Book Pro's strengths clearly lie in the fact that we were getting all this and more in a neat package. This clearly was the system of choice over a PC laptop.

Another area we need to evaluate was the selection of the best avatar. We were given two agents to choose from, one designed by Tim Bickmore, and another designed by Ken Perlin. As mentioned in the design section of this paper, the Bickmore's agent was a full-bodied character, but it could only process events that were pre-scripted by the designer. For our purposes, we needed something that could express multitudes of emotions. We also wanted to choose the character that user's are most likely to respond to. **Figure 4.1** shows the importance of choosing the right character. This figure shows the correlation between human likeness and likeability. It is important to note the concept of the "Uncanny Valley". If a computerized agent becomes too lifelike, it leads to an uncomfortable experience for the user. Since a computer agent does not have "life" in its eyes, it appears as if you are looking at a person in a coma. Both the agents we evaluated would provide the balance of human likeness we needed, but only Ken Perlin's agent would provide the emotional expressibility we needed. For these reasons, we chose to use the agent designed by Perlin. This agent uses OpenGL for its drawing routines, and has a powerful API that enables dynamic programmatic control of the face.

The next choice that needed to be made was what language to use for implementation of the system server. We needed to make sure that any language we choose was compatible with the Arduino boards, and had the libraries available to easily acquire data from the serial ports. We also wanted to try and keep the system server platform independent so that it could be used on multiple operating systems. For these reasons we choose C. C provided good performance and everything we needed for serial port communication, all with little overhead.

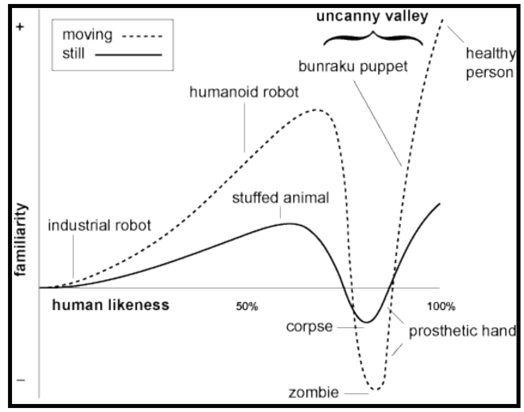


Figure 4.1: Uncanny valley and the relationship between human likeness and familiarity [8]

4.2 Conceptual Model

The overall system that we are implementing is quite complex, but for this development phase, our primary goal was to complete a proof of concept. During initial development, the system that we implemented consisted of a voltage differential sensor connected to an Arduino board. The Arduino board was then connected to the Mac book Pro via a USB port. The system server acquired the differential data (scaled from 0 to 255, size of an 8bit char) from the Arduino boards using serial port communication. The system server then directly interacts with the face engine using a command line interface. The system server was able to send commands to the agent API telling it what emotion to express. The next phase of development will consist of integrating the posture chair, pressure mouse, skin conductance glove, and facial recognition software with the current prototype. Once this level of communication is fully tested, we will implement the logic for decision making, and begin having the face react in a human manner. We also plan to implement a database for long term storage. This will give the system a way to predict future behavior for a long term user, and be able to collect data for human centered computing research. Refer to **Figure 4.1** for a high-level overview of how the system will be implemented.

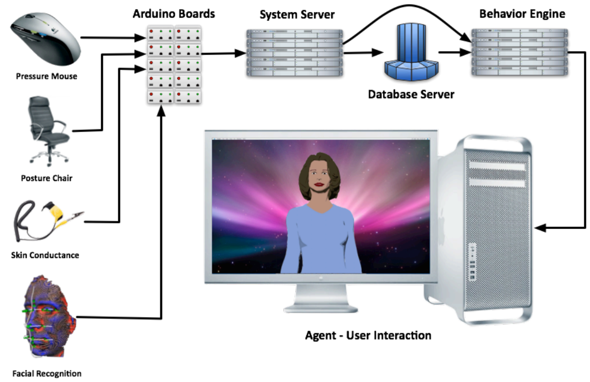


Figure 4.1: System overview including database implementation

4.3 Mathematical Model

One of the most important parts of our system is the Arduino boards. This element also has the largest impact on our computer resources. In order to test the impact the Arduino boards and the rate at which we polled the data had on the CPU usage, we examined the CPU usage at different samples per second. **Table 4.1** shows the values for the average CPU usage for each polling rate. **Figure 4.3** shows the graphical representation for this data. Note that these numbers represent the communication of data from the Arduino boards to the system server, not just the accumulation of data on the Arduino boards. Fitting the graph, we deduced that:

$$AverageCPUusage = \sqrt{PollingRate}$$

As one can see from the data, with about ten thousand (10,000) samples per second, we had an average CPU usage of about sixty-two percent (62%). These tests were run prior to any optimization in the system server on a Mac Book with the following specifications:

- 2GHz Intel Core 2 Duo Processor
- 1.25GB RAM

Since there will not be any other user applications running on our system, this use of resources is acceptable, but we hope that with future code optimization, we will see a reduction in the average CPU usage.

Table 4.1: Polling rate and average CPU usage of the Arduino boards

Polling Rate (sample / sec)	Average CPU Usage*
10000	62.20%
5000	54.40%
2500	49.80%
1000	42.50%
500	30.30%
100	22.60%

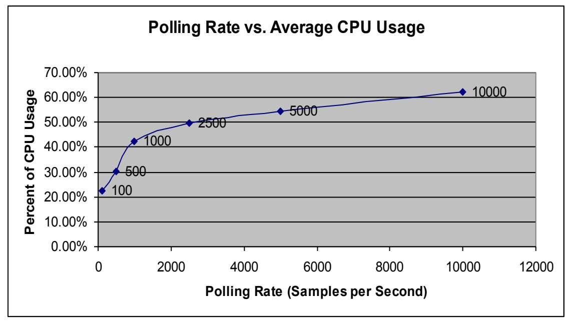


Figure 4.3: Graphical representation of polling rate vs. average CPU usage

4.4 Test Plans

4.4.1 Test Plan of Components

The original face application from Ken Perlin was not modified for our specific uses. In order to make sure it was suited for our application and there were no errors in the provided code, we carried out some initial testing. We divided this testing into four subsections as shown in **Table 4.2**. From this initial testing, we were able to determine that the application only met one of our four prime needs. In order for this application to successfully work in our system, we would need to develop the additional features. Once we developed these additional features, we completed unit testing on each component to make sure it was working as desired.

Table 4.2: Prime needs of face API

Feature	*Working (Yes/No)*
Key codes	Yes (One Exception)
Resizing of frame	No
Zooming of face	No
All emotions available	No

For the Ardunio board, we tested the serial port interface to make sure that it was functioning properly. We then monitored the serial ports from the system server to make sure we were receiving correct data from the boards and the data was being received consistently. More testing on the Arduino boards will be completed once we have the posture chair, pressure mouse, and skin conductance glove.

4.4.2 Testing of Overall System

Our current test plan only involved the hardware we had available to us. At the time of this paper, we only had one Arduino board and one voltage differential sensor. Our final system will consist of multiple sensors and three Ardunio boards. Our integration testing needed to make sure that data is received from the Ardunio board by the system server, and that the system server is communicating properly with the face API. In order to verify that all lines of communication were working properly, we set up a simple test. Anytime the sensor detected pressure, the system server would receive data and just change the emotion displayed by the face. We cycled through all the emotions we have currently developed to make sure that they are all displaying properly. The biggest challenge we had was adjusting the sensitivity of the sensor. If the threshold was too low, even a slight breeze or the jarring of the table would cause a change. If it was set too high, it would not recognize the change from just a person's touch. When the sensor detects a change in pressure, the output goes from high to low. In order to get the system server to only recognize changes from a person's touch, not in the surroundings, we needed to add a counter that would count the number of low inputs it received. We found that by setting this counter to look for twenty-five 0's in a row before making a change worked best. This required sustained pressure on the sensor, so no events would be triggered from jarring the table or a breeze in the room.

4.5 Experiment Design and Interpretation

Our research is not complete, and as such the experimental designs we have implemented are not finalized. However, the design of any of our experiments must take into consideration the goals we hope this system will facilitate. Our primary goal, as discussed earlier, is to facilitate learning in a subject by being aware of the subject's emotional state. The entire foundation of the Affective Learning Companion Project is built on the belief that affect, not cognizance, holds the key to learning.

We have developed a test plan for the system once we have all of the individual parts tested. Such a plan involves first getting the System Server and Face Code communicating, and testing how long the delay between the two is. The next step would be to tie the three sensors to their respective Arduino boards and, using a computer, test their input and output. Next, we would need to get the sensors, through the Arduino boards, talking to the System Server. Following this would be a long series of tests which will determine if user input (i.e. data from the sensors and Facial Recognition) is being correctly interpreted by the System Server and the data being sent to the Face Code. Finally, we would need a series of tests designed to make sure that the conclusions the System Server is drawing from the input (i.e. how the Face reacts given a set of circumstances) are correct within a well-defined range.

Since we have not finished with the entirety of our project, at this point we must discuss what we plan to do once construction and testing of the system is finished. As stated earlier, we plan to set up three Companions at different locations. Using technical input from the two universities and user response forms from the museum, we hope to compile enough data to fine-tune the Companion's decision-making algorithm to make it as lifelike as possible. In summary, the experiments we have conducted have been very limited in scope. We plan to conduct more exhaustive testing once we have a more fully-realized Affective Learning Companion and a system in place by which we can make updates to the prototype system and receive feedback.

4.6 Ethical Implications

The ethical implications of an affect-aware system are quite broad, with potential for a great deal of concern. One such dilemma, modified from [9], says that a system which is aware of a user's emotions might use the state of the user's affect against them. One example says that a soldier using an affect-aware training program might encounter a system which is aware of what tasks the soldier is most apprehensive of, and purposely tell the soldier to do them.[9] This could very easily hinder the soldier's learning process. Our Learning Companion could very well be modified to do this, and such an action would present an ethical problem for both the user of the system and its creator.

5. Conclusion and Summary

This design phase has been spent developing a proof of concept for an affective learning companion. We have established communication from a sensor to an Arduino board, which is then read by the system server using serial port communication. The system server is then able to control our agent via the face API. The face API is able to display multiple emotions and different ranges of those emotions. We have been behind schedule due to delay in receiving the promised sensors and facial recognition software. We have been reassured that these sensors will arrive shortly. We designed the proof of concept so the new hardware will be easily integrated with the rest of the system. The next design phase will consist of integrating the sensors as they arrive, implementing a database for long-term data storage, working on the logic for emotion recognition and how the agent should respond, and integrating the completed affective learning companion with a learning system.

Our recommendations for the next design phase are to incorporate elements that are currently missing, such as the sensors and the database.

This design phase, we struggled at first with getting precise requirements from our advisor. We learned that being prepared with questions at each meeting helped maintain focus on getting requirements clarified. There is still a lot to be learned since the project will continue and the major parts of development have yet to begin.

6. Acknowledgements

We would like to thank Dr. Yinong Chen and Jay Elston for the guidance they have given us this semester. We would also like to thank our advisor, Prof. Winslow Burleson, for constantly following up on everything we needed from the other university involved in the project. Lastly, we would like to thank Rebecca Cohlman for saving us a ton of money by printing our posters for the demonstration.

Each team member has carried their weight on this project. Nitesh, Tope and Matthew spent hours in the lab working with the face API. They created all the emotions that it can display, changed a considerable amount of hard-coded information to constants to make communication with the system server easier, and developed functions to manipulate the frame size and camera zoom. Salman, Peter, and Elizabeth worked on the system server and Arduino board communication. The Arduino board needed to be tested, and then communication set up between the system server and the face API. We have to generate behavior scripts, such as an introduction for the face, and also worked together for integration testing.

For the papers and presentations, each team member took a section to focus on. Each team member completed quality individual work by the deadline we had set. For the presentations, Salman and Peter did a wonderful job of putting everyone's sections into a cohesive and professional looking slide show.

7. References

[1] W. Burleson, "Affective Learning Companions: Modeling and Supporting Emotion During Learning- Project Summary," Nov. 2006.

[2] R. Picard, J. Cassell, B. Kort, R. Reilly, T. Bickmore, A. Kapoor, S. Mota, C. Vaucelle, "Affective Learning Companion." http://affect.media.mit.edu/projectpages/lc

[3] F. Stancato, A. Hamachek, "The interactive nature and reciprocal effects of cognitive and affective learning." 2001.

[4] R.W. Picard et al., "Affective Learning- A Manifesto." BT Technology Journal, vol. 22, pp. 253-269, 2004.

[5] E. Boff, E. Reategui, R. Viccari, "An affective learning agent-based virtual character for learning enviroments".

[6]K. Oertel, R. Kaiser, "Emotions in HCI – An Affective E-Learning System"

[7] R.W. Picard, S.D. Daily, "Evaluating affective interactions: Alternatives to asking what users feel".

[8] Mori, M. (1970). *The Uncanny Valley*. Retrieved May 1, 2008, from AndroidScience.com: http://www.androidscience.com/theuncannyvalley/proceedings2005/uncannyvalley.html

[9] Reynolds, Carson, and Rosalind Picard. "Ethical Evaluation of Displays that Adapt to Affect" http://affect.media.mit.edu/pdfs/04.reynolds-picard-ad.pdf

8. About the Authors

Salman Ahmad is a Computer Systems Engineering undergraduate student at Arizona State University. He graduated top of his class at Corona del Sol High School in Tempe, AZ and has upheld a 4.0 GPA in his studies at ASU.

Salman brings to our project programming experience both in working as a team and individually—his work at ASU's University Technology Office and at Freescale Semiconductor will be a significant asset to the ALC team.

Temitope Akinwande spent his high school years in Nigeria, and is now well on his way toward earning his BSE in Computer Systems Engineering at Arizona State University. A member of the Tau Beta Phi Engineering Honor Society and the Phi Eta Sigma National Honor Society, Temitope has maintained a 4.0 GPA throughout his three years at ASU. Temitope brings extensive knowledge of C, C++, Java, and many other languages to the ALC team, and will provide invaluable when it comes to programming know-how.

Elizabeth Cohlman is a Computer Systems Engineering student at Arizona State University, and is serving as our team leader. Her previous work experience at Godaddy Software, Volt Technical Resources, Arizona Public Service, and General Dynamics C4 Systems has given her a definitive advantage when it comes to organizing and managing the ALC team. Elizabeth's work experience has given her technical prowess and quick decision-making abilities which have proven invaluable with regards to overseeing the activities of the ALC team.

Matthew Jones is currently earning his BSE in Computer Systems Engineering at Arizona State University, and brings valuable experience to the ALC team. His extensive knowledge of Microsoft Visual Studio and related technologies, as well as database interaction, give him an edge technically in our team. Matthew has worked at Honeywell developing and deploying applications as an intern, which gives him vital experience working with a team on programming-related projects.

Nitesh Mehta is a Computer Systems Engineering undergraduate student at Arizona State University, having completed his high school studies in India. He brings to the ALC team significant programming experience, including everything from low level MIPS assembly language to high level knowledge of LISP and Prolog. Nitesh has experience working at the W.P. Carey School of Business in the IT department, giving him valuable knowledge of team dynamics which will greatly contribute to the ALC team.

Peter Woods is pursuing a BSE in Computer Systems Engineering at Arizona State Unviersity. Working for upwards of two years as an intern at Freescale Semiconductor, Peter brings extensive knowledge of Java application development and team dynamics to the ALC team. Peter has been programming since high school and knows many languages, giving him the ability to quickly prototype and deploy code in many different environments and across different architectures.

APENDIX A: Summary of Related Work

Pedagogical Agents as Learning Companions: The Role of Agent Competency and Type of Interaction
By: Yanghee Kim, Amy L. Baylor, PALS Group

This article describes an experiment involving 72 undergraduate students in an introductory computer-literacy course who were randomly assigned to one of four pedagogical agents: Low-Proactive, Low-Responsive, High-Proactive, and High-Responsive. The Low and High describe the competency of the agent i.e. how knowledgeable the agent is. The Responsive and Proactive describe the interaction of the agent. The proactive agent used its initiative to suggest various actions to the learner while the responsive agent waited till the learner asked for its help.

The result of the experiment showed that students in the proactive condition scored significantly higher than students in the responsive condition no matter what the competency of the agent was. Also, it was shown that students in the high-competency condition scored significantly higher than students in the low-competency condition.

In the section of the article titled "Instructional Design Issues", the researchers suggested that regarding agent competency, the agents should be designed as highly competent for learning contexts in which instructional goals focus on knowledge and skill acquisition. Thus the combination of High competent and Proactive agents will give the learner better knowledge and skill in the subject area. However, low competent agents according to the research helped improve the self-esteem of the learner and thus the self-efficacy because they feel better about themselves when the agent doesn't behave like it knows everything and this gives them the confidence and encourages them to continue the task.

Affective Learning – A Manifesto
By: R.W. Picard et al

The article is summarized below

- States that cognitive learning, as opposed to affective learning, has been greatly overemphasized in the last half-century
- Suggests that *affect*, emotional states, is just as important to learning as *cognizance*, or awareness of facts.
- States that the widely accepted method of recording affect, questionnaires, are unreliable because
  - They interrupt the process
  - They require the subject to be very aware of his/her own emotional state, which is by no means guaranteed
  - And they are almost wholly unreliable in the case of children
- States that a reason for affect lagging behind cognizance is its difficulty to measure
- Suggests that using several points of reference, i.e. facial recognition, posture sensing, pressure sensing, etc. can help predict affect and emotional response.
- Cites studies which show that postural movements of a child can be used to determine the child's affect nearly 80% of the time.
- Suggests that changing emotional states changes the manner in which we think.
- Suggests that affective learning can hold greater potential for learning, as an affective model may be able to get a child to like learning, rather than force them to.
- Suggests a manner of teaching in which the agent is viewed as a, "co-learner," who is figuring out the problem at the same time as a student; this is as opposed to a teacher, who already knows all the answers.

The Interactive Nature and Reciprocal Effects of Cognitive and Affective Learning
By: Frank Stacato and Alice Hamachek

This article discusses the importance of combining both affective and cognitive learning to produce meaningful classroom learning. Many researchers have tried to draw a distinct line between affective learning, which focuses on the feelings and attitudes that students have for themselves, and cognitive learning, which centers on the mental functions involved in thinking, knowing, and understanding subject matter content. This article proposes that there is a link between a student's academic performance and their self image. "Increased confidence in the self enables the student to secure the necessary energy and motivation to find solutions where otherwise he/she would give up very quickly. " This article also argues that a student's self image is a frame of reference that provides meaning to the cognitive information, and without the presence of meaning, the student would be unable to grasp the cognitive information that appears irrelevant. Some researchers would argue that putting an emphasis on affective learning means just making the learner feel good about their self, and not doing any thing that would hurt their confidence (i.e. not giving failing grades). The authors argue that this is not what is meant by affective learning. Affective learning is enhancing one's image of their own ability, motivating them to persist in difficult situations. When a learner continues to work through a difficult problem despite setbacks, cognitive learning takes place.

Affective Learning: Sustaining Motivation to Learn through Failure and a State of 'Stuck'
By: W. Burleson and R. Picard

Professor Burleson's paper talks on how the prerequisite to becoming an expert is the ability to persevere and remain motivated through failure. Many researchers creating learning systems have taken the approach of manipulating the task in terms of difficulty, focus, and other parameters in an effort to sustain users' motivation. There are various scenarios where this approach is impractical, undesirable, or simply impossible. The main point is that the task-manipulation approach misses the important opportunity of helping users develop skills to deal with failure and frustration. But the system we shall be working involves an approach that uses affective agents to help users develop metacognitive skills such as affective self-awareness for dealing with failure and frustration.

An important element of our approach is the use of one or more affective agents as peer learning companions to facilitate development of empathetic relationships with learners. This paper describes work in progress exploring how characteristics of affective agents can influence perseverance in the face of failure. Users sometimes get discouraged by the difficulty of a problem that they face and they get "stuck" in a negative state which basically discourages learning and gives them a sense of "negative" time (i.e. a never ending task/taking forever). This feeling of being out of control, a lack of concentration and other factors coupled together result in mental fatigue and distress caused due to the engagement with the activity. Hence, when a user is learning, it is proposed by Kort, Reilly and Picard, that there exists a flow which begins with anticipation, expectation and exploration and learners can develop new skills if they keep moving through but a negative corollary to this state of flow is the negative state of stuck and hence the article name. Flow and Struck are like yin and yang and are constantly balancing each other out but a conscious awareness of the state of stuck tends to diminish happiness and is an obstacle to the process of learning. Therefore to achieve a good balance between flow and struck and to maximize learning a system of an affective partner is proposed one that is more like a peer than a tutor. The objective of such an agent is to motivate the learner whenever it senses any sort of frustration or anxiety rather than let the user get stuck in a negative state.

In conclusion, developing learning experiences that facilitate self-actualization and creativity is among the most important goals of our society in preparation for the future. To facilitate deep understanding of a new concept -- to facilitate learning -- learners must have the opportunity to develop multiple and flexible perspectives. The process of becoming an expert involves failure, understanding failure, and the motivation to move onward. Meta-cognitive awareness can play a role in developing an individual's ability to persevere through failure. We have presented a strategy for using affective agents to help sustain motivation through failure and a state of "Stuck." Many have taken the approach of manipulating the task; we take the approach of assisting users to modulate the effects of their own affective state. This strategy is likely to be particularly effective in assisting learners to pursue difficult and un-manipulated tasks that arise frequently in their everyday lives.

Emotions in HCI – An Affective E-Learning System
By: Robin Kaiser and Karina Oertel

The researchers in this article have started work on an affective learning system. This article outlined the preliminary steps for creating an Affective Computer System.

The major points are summarized below.

- Affective Computing has been defined as the interplay between emotions and computers

- Overall, today computer systems still do not consider the emotions of users
- The researchers have worked on developing a system to detect human emotions
- The system uses a variety of input devices including of a glove, a chest belt, and a data collection unit
- The system used here uses what is called the Russell model of emotions. It uses a valence-arousal relation to characterize human emotions.
- The advantage here is that several emotions are represented with this relation. The emotions have a surrounding "region" and if the response falls within that area, that emotion is evaluated to be exhibited
- The system here was designed to target only certain emotions, typically negative emotions like anger or frustration.
- The system tried to detect those emotions and provide means to alleviate them. They offered interesting mechanism to achieve this result including using a hammer to break the computer.
- Ultimately, the results in the affective learning group were better than the control group and the project does show significant potential

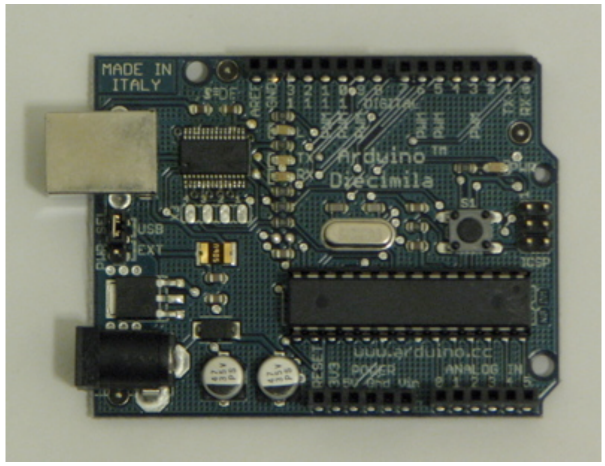
Evaluating affective interactions: Alternatives to asking what users feel
By: Rosalind Picard and Shaundra Bryant Daily

This article discusses using behavior-based models when evaluating affective interactions, as opposed to asking users how they feel or what they were thinking after the fact. It details two categories of evaluation: body measures, such as changes in muscle activity, and task measures, such as how someone goes about solving a problem. The ability to detect a user's emotional state has greatly improved over the last few years, with things like face recognition and more advanced sensors which can pick up on a person's emotions without feeling intrusive for the user. Furthermore, these measurements can be taken at the same time the user is interacting with a system, thus alleviating the need to interrupt the user or wait until after the interaction completes to evaluate their emotional response. Task measures of affect measure how a person reacts to stimuli in the test itself. For example, will a user do better if a test is well formatted and easy to understand as opposed to someone taking a test that is poorly formatted or confusing? In a similar vein, the user's perception of how long a task took to complete can be weighed against the actual amount of time the task took and used to determine their overall success at completing the task. Users in a study tended to underestimate the time it took to complete an easy or straightforward task, whereas those who were frustrated or given a harder task overestimated the time it took to complete the task. These task measures are much less understood than the body measures which were also discussed, however they can be used to determine how a user reacted to a certain test or task without having them directly describe their experiences, and without requiring special sensors or sophisticated analysis. By measuring the affective interactions using both physical, body measures as well as psychological, task measures, a user's gamut of experiences when interacting with a computer system can be evaluated with a degree of accuracy unobtainable using classic methods.

Appendix B. Hardware Data Sheets

Arduino Communication Board

Product Name: Arduino Diecimila



Overview

The Arduino Diecimila is a microcontroller board based on the ATmega168 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

"Diecimila" means 10,000 in Italian and was named thusly to mark the fact that over 10,000 Arduino boards have been made. The Diecimila is the latest in a series of USB Arduino boards; for a comparison with previous versions, see the index of Arduino boards.

Schematic & Reference Design

EAGLE files: arduino-diecimila-reference-design.zip

Schematic: Arduino-Diecimila-schematic.pdf

Note that R2 is not mounted and that R3 has been replaced by a 100 nano-farad capacitor.

Summary

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6

DC Current per I/O Pin	40 mA
Flash Memory	16 KB (of which 2 KB used by bootloader)
SRAM	1 KB
EEPROM	512 bytes
Clock Speed	16 MHz

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader). It has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library).

Programming

The Arduino Diecimila can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The ATmega168 on the Arduino Diecimila comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the ATmega168 through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Diecimila is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. Version 0009 of the Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Diecimila is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Diecimila. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

Input and Output

Each of the 14 digital pins on the Diecimila can be used as an input or output. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. Pins 3, 5, 6, 9, 10, and 11 can provide PWM output; for details see the analogWrite() function. If anything is connected to pins 0 and 1, it will interfere with the USB communication, preventing new code from being uploaded or other communication with the computer.

The Diecimila has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and some low-level code.

See also the mapping between Arduino pins and ATmega168 ports.

Communication

The Arduino Diecimila has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

A SoftwareSerial library allows for serial communication on any of the Diecimila's digital pins.

The ATmega168 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega168 datasheet.

Power

The Arduino Diecimila can be powered via the USB connection or with an external power supply. The power source is selected by the PWR_SEL jumper: to power the board from the USB connection, place it on the two pins closest to the USB connector, for an external power supply, the two pins closest to the external power jack.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. A low dropout regulator provides improved energy efficiency.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

USB Overcurrent Protection

The Arduino Diecimila has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Diecimila PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Piezoelectric film properties

Measurement Specialties, Inc.
Sensor Products Division
950 Forge Avenue
Norristown, PA 19403

Tel: 610.650.1500
FAX: 610.650.1509
Internet: www.msiusa.com
e-mail: sensors@msiusa.com

Piezo film is a flexible, lightweight, tough engineering plastic available in a wide variety of thicknesses and large areas. Its properties as a transducer include:

- Wide frequency range—0.001 Hz to 109 Hz.
- Vast dynamic range (10-8 to 106 psi or µ torr to Mbar).
- Low acoustic impedance—close match to water, human tissue and adhesive systems.
- High elastic compliance
- High voltage output—10 times higher than piezo ceramics for the same force input.
- High dielectric strength—withstanding strong fields (75V/µm) where most piezo ceramics depolarize.
- High mechanical strength and impact resistance (109—1010 Pascal modulus).
- High stability—resisting moisture (<0.02% moisture absorption), most chemicals, oxidants, and intense ultraviolet and nuclear radiation.
- Can be fabricated into unusual designs.
- Can be glued with commercial adhesives.

One major advantage of piezo film over piezo ceramic is its low acoustic impedance which is closer to that of water, human tissue and other organic materials. For example, the acoustic impedance (ZO = ρ υ) of piezo film is only 2.6 times that of water, whereas piezo ceramics are typically 11 times greater. A close impedance match permits more efficient transduction of acoustic signals in water and tissue.

Piezo film does have some limitations for certain applications. It makes a relatively weak electromechanical transmitter when compared to ceramics, particularly at resonance and in low frequency applications. The copolymer film has maximum operating/storage temperatures as high as 135oC, while PVDF is not recommended for use or storage above 100 bC. Also, if the electrodes on the film are exposed, the sensor can be sensitive to electromagnetic radiation. Good shielding techniques are available for high EMI/RFI environments.

Table 1 lists typical properties of piezo film. Table 2 provides a comparison of the piezoelectric properties of PVDF polymer and two popular piezoelectric ceramic materials.

Piezo film has low density and excellent sensitivity, and is mechanically tough. The compliance of piezo film is 10 times greater than the compliance of ceramics. When extruded into thin film, piezoelectric polymers can be directly attached to a structure without disturbing its mechanical motion. Piezo film is well suited to strain sensing applications requiring very wide bandwidth and high sensitivity. As an actuator, the polymer's low acoustic impedance permits the efficient transfer of a broadband of energy into air and other gases.

Table 1. Typical properties of piezo film

Symbol	Parameter		PVDF	Copolymer	Units
t	Thickness		9, 28, 52, 110	<1 to 1200	μm (micron, 10^{-6})
d_{31}	Piezo Strain Constant		23	11	10^{-12} $\frac{m/m}{V/m}$ or $\frac{C/m^2}{N/m^2}$
d_{33}			-33	-38	
g_{31}	Piezo Stress constant		216	162	10^{-3} $\frac{V/m}{N/m^2}$ or $\frac{m/m}{C/m^2}$
g_{33}			-330	-542	
k_{31}	Electromechanical Coupling Factor		12%	20%	
k_t			14%	25-29%	
C	Capacitance		380 for 28μm	68 for 100μm	pF/cm^2 @ 1KHz
Y	Young's Modulus		2-4	3-5	10^9 N/m^2
V_0	Speed of Sound	stretch:	1.5	2.3	10^3 m/s
		thickness:	2.2	2.4	
p	Pyroelectric Coefficient		30	40	10^{-6} C/m^2 °K
ε	Permittivity		106-113	65-75	10^{-12} F/m
$ε/ε_0$	Relative Permittivity		12-13	7-8	
$ρ_m$	Mass Density		1.78	1.82	10^3kg/m
$ρ_e$	Volume Resistivity		$>10^{13}$	$>10^{14}$	Ohm meters
$R_□$	Surface Metallization Resistivity		<3.0	<3.0	Ohms/square for NiAl
$R_□$			0.1	0.1	Ohms/square for Ag Ink
tan $δ_e$	Loss Tangent		0.02	0.015	@ 1KHz
	Yield Strength		45-55	20-30	10^6 N/m^2 (stretch axis)
	Temperature Range		-40 to 80...100	-40 to 115...145	°C
	Water Absorption		<0.02	<0.02	% H_2O
	Maximum Operating Voltage		750 (30)	750 (30)	V/mil(V/μm), DC, @ 25°C
	Breakdown Voltage		2000 (80)	2000 (80)	V/mil(V/μm), DC, @ 25°C

Table 2. Comparison of piezoelectric materials

Property	Units	PVDF Film	PZT	$BaTi0_3$
Density	10^3kg/m3	1.78	7.5	5.7
Relative Permittivity	$ε/ε_0$	12	1,200	1,700
d_{31} Constant	(10^{-12})C/N	23	110	78
g_{31} Constant	(10^{-3})Vm/N	216	10	5
k_{31} Constant	% at 1 KHz	12	30	21
Acoustic Impedance	(10^6)kg/m^2-sec.	2.7	30	30

Operating properties for a typical piezo film element

The DT1 element is a standard piezo film configuration consisting of a 12x30 mm active area printed with silver ink electrodes on both surfaces of a 15x40 mm die-cut piezo polymer substrate.

1. Electro-Mechanical Conversion

(1 direction) 23 x 10-12m/V, 700 x 10-6N/V
(3 direction) -33 x 10-12m/V

2. Mechano-Electrical Conversion

(1 direction) 12 x 10-3V per microstrain, 400 x 10-3V/μm, 14.4V/N
(3 direction) 13 x 10-3V/N

3. Pyro-Electrical Conversion

8V/ o K (@ 25 o C)

4. Capacitance

1.36 x 10-9F; Dissipation Factor of 0.018 @ 10 KHz; Impedance of 12 KΩ @ 10 KHz

5. Maximum Operating Voltage

DC: 280 V (yields 7 μm displacement in 1 direction)
AC: 840 V (yields 21 μm displacement in 1 direction)

6. Maximum Applied Force (at break, 1 direction)

6-9 kgF (yields voltage output of 830 to 1275 V)

Appendix C. Face API User's Guide

Basics

The face we decided to implement was given to us by Ken Perlin. It's been written entirely in JAVA but also uses OPEN GL to do polygon rendering. Listed below are the files that make up the face code and their uses are as follows:

- \beam\face\src\face\Face.java – The most crucial component of our program. This function is responsible for initializing the face and how to orient it with regards to a specific emotion. It also entails what shapes and colors each polygon should be and it's 3-D orientation in space.
- \beam\face\src\face\InputThread.java – This was created by us to send commands to the face via the command line. This class helped us to make an abstract interface to our face API which would enable the System server to communicate with it by sending commands through the command line. It calls the various emotion functions in our Face.java file.
- \beam\face\src\face\FaceRenderer.java – This file is being used to position our API. It renders the frame within which the face is contained and it is also responsible for the size of the actual face. This class has the ability to zoom in and out of our agent using the "camera" functionality. It is important with respect to the fact that it has the main() function in it.
- \beam\face\src\face\Graphics.java – This is the class that's doing the actual polygon rendering. It has functions within it to draw and fill shapes such as rectangles, ovals, custom polygons, etc.
- \beam\face\src\face\ImprovMath.java – We really haven't touched this class at all. We believe it is being used to add some smoothness to the face and to make it look nicer.
- \beam\face\src\face\KeyMapper.java – This a really small class and is used to map all the keys of a standard keyboard into two strings data0 and data1 respectively.
- \beam\face\src\face\Matrix3D.java – This class is used to create 3D points and homogeneous vectors, and also to create transformation matrices with these. There are methods to rotate, translate, and scale transformations, and to apply transformations to vectors. We can also get and set the elements of matrices and vectors.

Key functionality

Listed below are the capabilities of our face and what it can do:

- Our face has 21 static emotions such as look happy, look sad, think, angelic etc. All we did to make these emotions was to orient the different aspects of the face such as the eyes, nose, and brows and so on.
- In order to make our face look more realistic we have added two animations to it. These include the laugh and talk functions which basically make the face laugh (or talk) for a specified amount of time for example 3 seconds.
- One of the other challenges we faced for our project was the fact that a user emotion generally has various degrees to it. A person can either be happy or be feeling ecstatic. In order to model this human behavior we implemented degrees of emotions to our face. We chose three basic emotions which every human encounters namely happy, angry and frustrated, and added different shades to these emotions. The net result was that we added 5 degrees of emotions to these three specific human behaviors which enabled us to more closely reflect an actual human.

Basic commands

Each aspect of the face has numbers (1-5 except Mouth, 1-4) which detail to what level the change should be taken. The following is a list of those commands and their descriptions:

brows 1	// Brows very low	brows 4	//Brows high
brows 2	// Brows low	brows 5	// Brows very high
brows 3	//Brows middle (reset)		

lids 1	// Close eyelids
lids 2	// Squint
lids 3	// Open (reset)
lids 4	// Wide
lids 5	// Humongous
lids 6	// Eyelids down
lids 7	// Eyelids middle (reset)
lids 8	// Eyelids up
eyes 1	// Eyes look far left
eyes 2	// Eyes look slight left
eyes 3	// Eyes look straight ahead
eyes 4	// Eyes look slight right
eyes 5	// Eyes look far right
gaze 1	// Eyes look down
gaze 2	// Eyes look slight down
gaze 3	// Eyes look straight ahead
gaze 4	// Eyes look slight up
gaze 5	// Eyes look up
face 1	// Face look actor's right
face 2	// Face looks slight right
face 3	// Face look straight ahead
face 4	// Face look slight left
face 5	// Face look actor's left
head 1	// Head tilt down
head 2	// Head tilt slight down
head 3	// Head centered
head 4	//Head tilt slight up
head 5	// Head tilt up
tilt 1	//Head tilt actor's right
tilt 2	//Head tilt slight right
tilt 3	//Head tilt centered
tilt 4	//Head tilt slight left
tilt 5	//Head tilt actor's left
mouth 1	//Mouth open big
mouth 2	//Mouth open medium
mouth 3	//Mouth open small
mouth 4	//Mouth closed
lips 1	//Lips very wide
lips 2	//Lips wide
lips 3	//Lips reset (normal)
lips 4	//Lips kiss
lips 5	//Lips pucker
smile 1	//Lips frown
smile 2	//Lips small frown
smile 3	//Lips normal
smile 4	//Slight smile
smile 5	//Smile
sneer 1	//Lips thin
sneer 2	//Lips purse
sneer 3	//Lips normal
sneer 4	//Small sneer
sneer 5	//Big sneer

Emotion shades:

angry 1	//Brows down
angry 2	//Mad
angry 3	//Angry
angry 4	//About to explode
angry 5	//Scream
frust 1	//Thinking
frust 2	//Concentrate
frust 3	//Scared/Worried
frust 4	//Getting angry
frust 5	//Borderline insane
happy 1	//Content
happy 2	//Pleased
happy 3	//Joy
happy 4	//Excited
happy 5	//Chuckle

Base emotions:

angry	// Same as angry 4
wink	
sleepy	
bored	
sad	
frightened	
ecstatic	// Same as Happy 5
angelic	
arrogant	
mischief	
tired	
scream	//Same as angry 5
think	//Same as frust 1
shy	
annoyed	
confused	//Same as frust 3
concentrate	//Same as frust 2
frustrated	//Same as frust 4

reset // Resets face

Animations:

laugh # [1]

talk #

Compiling instructions

1. Copy the beam folder to your base windows directory (in our case that was the C drive).
2. Open the command line using the Windows Key + R and then typing “cmd” or from the start menu go to Accessories and then the Command Prompt.
3. Change your active working directory to your base windows directory (C: drive for us) by using the “cd..” command.
4. Type in “compile.bat” to compile all the files.
5. Type “run.bat” to run the Face API.
6. Once the face is up you should have a command line cursor waiting for your input. Use this to send the various commands as listed in the “table of face commands”.

[1] (# can be any number within the range of an integer i.e. -2,147,483,648 to +2,147,483,647)