Emotional Experience of Students Interacting with a System for Learning Programming

Thomas James Tiam-Lee and Kaoru Sumi
Future University Hakodate
116-2 Kamedanakanocho
Hakodate, Hokkaido 041-8655

Abstract

This paper discusses the emotional experience of students while interacting with a system for solving programming exercises. We collected data from 73 university students who each used the system for 45 minutes. They were also asked to provide self-report affect judgments which describe the emotions that they experienced at different moments in the session. Using this data, we performed an analysis of the emotional experience of students while interacting with the system content, focusing particularly on the transitions across different emotions. We also analyzed the facial expressions, pose, and logs in relation to the various emotional states. We believe this study can contribute to the recognition of student affect in programming activity, and can potentially be used in a variety of applications such as intelligent programming tutors.

Introduction and Related Studies

Recently, there has been great interest in modelling the affective experience of students while engaging in learning tasks. Previous studies have found that positive emotions such as enjoyment are positively correlated with student achievement, whereas negative emotions such as boredom as negatively correlated with student achievement (Daniels et al. 2009). Emotions have also been shown to be associated with motivation and self-regulated learning (Mega, Ronconi, and De Beni 2014), as well as the quality of the learning experience (Cho and Heron 2015). These studies provide support for efforts in automatically recognizing the affective state of students while learning.

Good affective models of learning-specific emotions can open up opportunities for intelligent tutoring systems (ITS) by allowing them to respond not only to the cognitive state but also the affective state of the student. For example, if confusion is detected, the ITS may provide an intervention in the form of a hint. This is referred to as affective tutoring. Previous ITS such as AutoTutor (D'Mello and Graesser 2012a), MetaTutor (Jaques et al. 2014), and FERMAT (Zatarain-Cabada et al. 2014) have shown the potential of affective tutoring in improving students' learning across various domains.

However, understanding affect in complex learning tasks still proves to be challenging. One such task is computer programming. In learning programming, students spend a lot of time writing, testing, and debugging code. Interactions with the tutor agent typically occur in less frequency, and displays of affect through facial expressions tend to be more subtle as well. Despite this, previous studies have shown that students have a rich affective experience of learning-specific emotions while learning programming (Bosch, D'Mello, and Mills 2013; Bosch and D'Mello 2013) - information that could hold much potential for improving programming instruction.

A study by Bosch, Chen, and D'Mello showed that it is difficult to automatically recognize fixed-point affect judgments in programming sessions by using face features alone. One of the ways to address this is to combine face data with an understanding of the affective experience of students while doing the task to improve the recognition of affective states. In programming activity, there is a rich set of data from the interaction between the student and the system that could be used to help model affect.

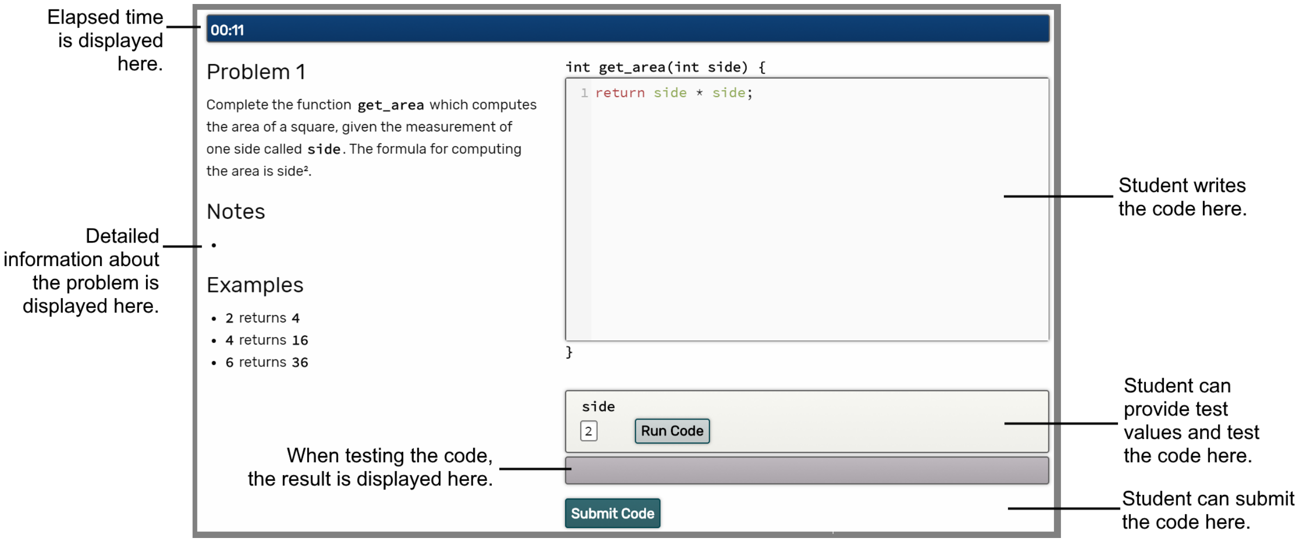
In this paper, we discuss a statistical analysis of the affective experience of students interacting with a system for programming practice. We focus our discussions on the distribution of the emotions experienced by the students, how these emotions transiton from one type to another, and which features are useful for recognizing these affective states.

System Design and Experimental Setup

In this section we discuss the system design and the experimental setup for this study. We recruited 38 students from Future University Hakodate in Japan and 35 students from De La Salle University in the Philippines to participate in this study. We chose to recruit participants from two different countries not only to increase the amount of data that could be collected but also to investigate if there are similarities or differences between the two groups. All students who participated in the study were enrolled in a freshman introductory programming course in their respective universities at the time of the experiment.

Each student interacted with a system in which they have to solve a series of coding exercises. A screenshot of the system is shown in Figure 1

In each exercise, they must write the body of a function



Elapsed time is displayed here.

00:11

Problem 1

Complete the function `get_area` which computes the area of a square, given the measurement of one side called `side`. The formula for computing the area is side².

Notes

-

Examples

- `2` returns `4`
- `4` returns `16`
- `6` returns `36`

Detailed information about the problem is displayed here.

```
int get_area(int side) {
return side * side;
}
```

Student writes the code here.

Student can provide test values and test the code here.

When testing the code, the result is displayed here.

Student can submit the code here.

Figure 1: Screenshot of the system used in the data collection

Table 1: Exercise List

No.	Problem Description
1	Return the area of a square given the length of the side.
2	Return the change given the price of the item, the number of items bought, and the amount paid by the customer.
3	Return the larger value between two integers.
4	Return the name of the winner of a rock paper scissors game given what each player played.
5	Return the age of the middle child given the ages of three brothers.
6	Return the total of all integers in a given array.
7	Given an array of integers, return the number of elements that are divisible by 3.
8	Return the sum of all the factors of a given integer.
9	Given an array of integers, return the number of times that the most frequently occurring element appears.

according to a given specification. For example, in one of the exercises, the function took in an integer value representing the length of the side of a square, and should return the area of the square. The students were not allowed to skip exercises until a correct solution had been provided. The order of the exercises were fixed for all of the subjects, and were arranged in increasing difficulty. Table 1 shows the exercises that were given to the students.

The students performed three types of interaction with the system throughout the session.

First, the student could edit the code. A code edit may be classified as an insertion (adding characters) or a deletion (removing characters). The system provides an interface for editing code similar to an integrated development environments (IDE).

Second, the student could test the code. This was done by providing sample values for each input parameter of the function, and then clicking the ”Run Code” button to execute the code. The system responded to this command by displaying the result of the execution on the screen. The result may either be a successful compilation, a compilation error, or a runtime error. In the case of a successful compilation, the return value of the function was displayed. In the other two cases, the Java error message was displayed instead.

Third, the student could submit the code. The system then automatically checked the code by running it on a set of predefined test cases and then comparing the result against the expected values. If the code passed all test cases, the system responded with a ”correct” message and displayed the next problem. If the code failed at least one test case, the system displayed a ”wrong” message. The student was not informed of the failing test cases nor the type of error that occurred, if any.

Each student used the system for 45 minutes, or until all the problems were solved correctly. Throughout the session, the system automatically logged information which comprised of (1) a video recording of the student’s face, (2) all code changes, and (3) all compilations and submissions.

At the end of the coding phase, the session was automatically split into intervals. The boundaries of these intervals corresponded to key moments in the session, which included: program compilation (testing the program), program submission, and the beginning and ending of each typing sequence. A typing sequence refers to a sequence of code changes (insertions and deletions) with a maximum interval of 5 seconds in between. We chose this limit because sometimes students pause to do brief moments of thinking while they are typing. Intervals that were less than 5 seconds in length were merged with the succeeding interval until it was at least 5 seconds in length.

Table 2: List of Affective State Labels

Label	Definition
Engaged	You are immersed in the activity and enjoying it.
Confused	You have feelings of uncertainty on how to proceed.
Frustrated	You have strong feelings of anger or disappointment.
Bored	You feel a lack of interest in continuing with the activity.
Neutral	There's no apparent feeling.

Table 3: List of Action State Labels

Label	Definition
Reading	You are reading the problem.
Thinking	You are thinking about the next step you will do.
Writing	You are translating your ideas by writing them into code.
Finding	You are trying to determine what the error is or thinking about how to fix it.
Fixing	You are trying to change something in the code to fix the error.
Unfocused	You are not focused in the task and your mind is thinking about other things.
Other	The above labels do not apply.

To collect affective data on the programming session, each student was asked to provide self-report affect and action judgments on each interval. A maximum limit of 150 intervals was set for each student to keep the annotation task manageable. If the session contained more than 150 intervals, we randomly chose 150 intervals for annotation. For each interval, the student was asked to select an emotion label, describing the affective state that best described his or her experience during that interval, and an action label, describing the type of action he or she was doing during that interval. To minimize subjectivity in self-reports, we provided a clear definition for each label. We chose the emotions of engagement, confusion, frustration, boredom, and neutral for the affective state labels based on a previous study which showed that these were the common emotions experienced by novice programmers (Bosch, D'Mello, and Mills 2013).

Tables 2 and 3 show the emotion labels and action labels respectively, along with their definitions.

Data collection was performed from July to August 2018 in Future University Hakodate in Japan and De La Salle University in the Philippines. A total of 38 Japanese students and 35 students who were at the time taking up freshmen programming courses participated. We were able to collect a total of 49 hours, 25 minutes, and 17 seconds of session data. This comprised of 9,702 annotated intervals. The average number of annotated intervals collected per student is 132.9. The average length of an interval is 17.24 seconds, resulting in fairly fine-grained affect information.

Table 4: Distribution of Different Affective States

Emotion	Japanese	Filipino
Engaged	34.89%	36.09%
Confused	18.05%	19.51%
Frustration	16.20%	22.91%
Bored	7.94%	6.07%
Neutral	22.92%	15.42%

Results of the Analysis

In this section, we present the results of the analysis done on the data. Our analysis aimed to explore the following questions:

- What is the distribution of the affective states experienced by the students?
- What are the common transitions between affective states?
- Which events trigger transitions between affective states?
- Which features could be used to recognize the presence of affective states?

Affective State Occurrences

In this section we present results regarding the occurrence of the different affective states as reported by the students. Table 4 shows the distribution of the different affective states reported by the Japanese and Filipino students. This is based not on the number of intervals but on the total duration of the intervals.

A high level look at the distribution reveals similarities between the two groups. Engagement was the emotion that was reported the most, and comprised of approximately a third of the total duration of the session. Meanwhile, confusion and frustration each comprised of around a fifth of the total duration. In this experiment, Japanese students tend to report the neutral emotion more often than the Filipino students.

We also investigated if the student's performance had a relationship with the distribution of the reported affective states. To do this, we divided the students into 5 groups based on the number of problems they were able to solve in the session. Students who were able to solve more problems were considered to have a better performance than those solved less problems. We then computed for the distribution of the affective states in each group. Figure 2 shows the results for both groups.

Noticeable trends on the distribution of affect reports could be observed based on the number of problems solved. Boredom and frustration decreased as the number of problems solved increased, while engagement increased along with the number of problems solved. Interestingly, a drop on the amount of engagement reports was observed in both groups on the 8-9 problems solved category. A probable cause of this is the increase in confusion and frustration due to the difficulty of the last two exercises offered by the system. These observations support previous literature that showed correlations between different types of emotions and

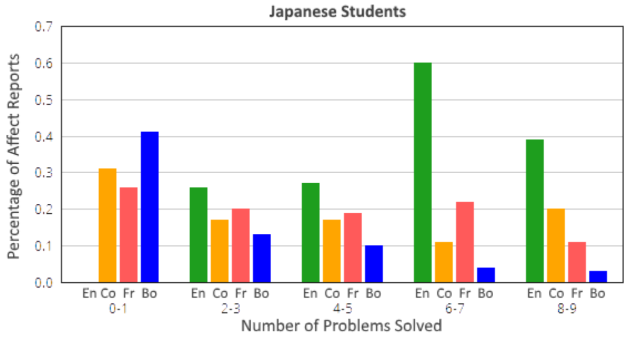


Figure 2: Distribution of Affective State Reports Grouped by Performance

Table 5: Frequency of Transitions Between Affective States (the row is the previous state and the column is the next state)

	Japanese					Filipino			
	En	Co	Fr	Bo		En	Co	Fr	Bo
En		70	17	9	En		54	11	8
Co	68		47	13	Co	68		30	17
Fr	21	35		10	Fr	15	26		17
Bo	8	10	11		Bo	9	8	13	

student performance, and highlight the importance of managing student affect in learning systems.

Affective State Transitions

In this section we present results on the transitions between different affective states. Table 5 shows the frequency of each transition between pairs of affect reports. We only considered intervals that that are immediately consecutive. The data shows that certain transitions occurred more often than others. For example, transitions from engagement to confusion and vice versa occurred in large frequency for both of the groups.

To verify which transitions were significant, we applied a scoring metric for transition likelihoods between affective states proposed by D'Mello (2012). We computed the likelihood score of a state following another as the conditional probability of the next state occurring after the current state normalized over the overall probability of the next state occurring. This likelihood score value has a range of $(-\infty, 1]$.

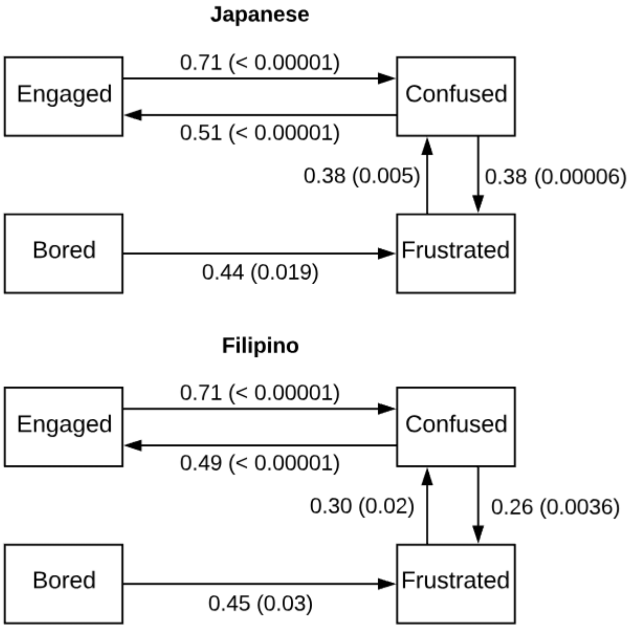


Figure 3: Significant transition likelihood scores between affective states. Edge values are the mean likelihood values and the values in parenthesis are the p values

A likelihood value > 0 means that the transition occurred above chance. We did not consider transitions to the same affective state. Likelihood score values were computed for every student and a two-tailed one sample t-test was performed. Significant transitions ($p \leq 0.05$) and their corresponding mean likelihood scores are shown in Figure 3.

Similar observations were found for both Japanese and Filipino groups. These results are consistent with the theoretical model of affect dynamics for complex learning proposed by D'Mello and Graesser (2012b). In this model, a student in the state of engagement may transition to state of confusion when a hurdle is encountered. Depending on whether the hurdle is resolved or not, the student may transition back to engagement in the case of the former, or transition to a state of frustration in the case of the latter. In the model, frustration may transition to boredom, but this was not observed at a significant level in our data. This may be because several of the students did not report boredom at all, making it difficult to establish a statistical significance.

Triggers of Affective Transitions

In this section, we present results on the events that trigger affective state transitions. We identified boundaries of the intervals that were associated with compilation and submission events. A compilation event refers to a point in the session where the student tested the code by providing sample inputs. This event could result to a compilation with no errors or a compilation with errors (syntax or runtime). On the other hand, a submission event could either result into a submission that passed or failed.

We computed the likelihood of each affective state to fol-

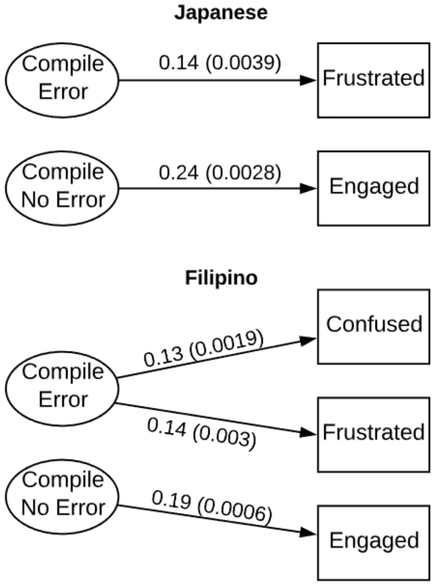


Figure 4: Significant transition likelihoods from compilation and submission events to affective states. Edge values are the mean likelihood values and the values in parenthesis are the p values

low each compilation or submission event. We did this for each student and performed a two-tailed one sample t-test to determine which transition likelihoods were significant. We did not consider submission passed events because there was a low number of occurrences of this event that was followed by an interval. We applied a Bonferonni correction resulting in $\alpha = 0.004$.

The results are shown in Figure 4. For the Japanese group, we found that there was a likelihood that a compilation error was followed by frustration in levels above chance. On the other hand, for the Filipino group, we found that compilation errors were likely to be followed by both confusion and frustration. In both groups, a compilation without any errors was likely to be followed by engagement.

We also performed frequency analysis of n-grams on the session data to determine common sequences of actions that lead to an affective state. We considered n-grams of length 4 to 6. Table 6 shows the common sequences. We considered a sequence to be common if it accounts for at least 2% of all sequences leading to the target affective state with the same n-gram length.

It can be seen that transitions in the affective state are often observed after a compilation or submission. This is expected because these actions are the types of interactions that the system responds or gives feedback to. The feedback likely triggers the change in affective state. Furthermore, it can be seen that engagement and confusion are associated with writing and thinking, while frustration is associated with finding and fixing bugs.

Table 6: Common Action Sequences That Lead to an Affective State (Frequency is occurrence over all n-grams of the same emotion and length)

Freq.	Sequence
12.92%	Writing → Thinking → **Engaged**
11.84%	Writing → Compile No Error → **Engaged**
9.83%	Thinking → Writing → **Engaged**
9.15%	Thinking → Compile No Error → **Engaged**
8.36%	Writing → Thinking → Writing → **Engaged**
8.75%	Writing → Thinking → Writing → Thinking → **Engaged**
14.9%	Thinking → Compile No Error → **Confused**
8.16%	Finding Bug → Compile No Error → **Confused**
13.86%	Compile No Error → Thinking → Compile No Error → **Confused**
10.18%	Thinking → Compile No Error → Thinking → Compile No Error → **Confused**
9.17%	Compile No Error → Thinking → Compile No Error → Thinking → Compile No Error → **Confused**
8.74%	Finding Bug → Compile Error → **Frustrated**
8.33%	Fixing Bug → Compile Error → **Frustrated**

Predictors of Affect

In this section, we present results on features that are useful for predicting affect. We investigated log-based features (code compilations, typing, etc.) and face-based features. For face-based features, we used OpenFace, a computer vision toolkit capable of head pose estimation, eye gaze estimation and action unit recognition from videos (Baltrusaitis et al. 2018).

Action units are a taxonomy of fundamental actions of facial muscles used in previous studies for emotion recognition (Ekman and Friesen 1975). An example of an action unit is raising the inner brow or raising the cheek. OpenFace has shown good inter-rater agreement with baselines set by human annotators across multiple datasets in AU detection (Baltrušaitis, Mahmoud, and Robinson 2015). Table 7 shows a list of the features that we considered.

In this analysis, we treated each interval as a separate instance, with the affect report as the class label. To determine which features are useful for recognizing affect, we used RELIEF-F feature ranking to rank features based on how discriminative they were against the closest neighboring instance with a different class. We identified the features that scored high in this ranking, and then further made statistical analyses on these features. The following subsections discuss these.

Log-Based Features We performed paired Wilcoxon signed-rank tests to determine if action state significantly co-occur with various affective states. We found that engagement co-occurs significantly more with writing ($\mu = 0.49$) than thinking ($\mu = 0.25, p = 0.0000009$), and co-occurs significantly more with thinking more than the other actions.

Table 7: List of Features

Feature	Description
Log-based Features	
insert	No. of insertions in the code
remove	No. of deletions in the code
type	No. of insertions and deletions in the code
compile_err	No. of compilations with syntax error
compile	No. of compilations without syntax error
Pose-based Features (from OpenFace)	
pose_Tx	Location of the head in x axis in mm
pose_Ty	Location of the head in y axis in mm
pose_Tz	Location of the head in z axis in mm
pose_Rx	Rotation of the head in x axis in radians
pose_Ry	Rotation of the head in y axis in radians
pose_Rz	Rotation of the head in z axis in radians
gaze_angle_x	Gaze angle x in world coord. in radians
gaze_angle_y	Gaze angle y in world coord. in radians
Face-based Features (from OpenFace)	
AUs	Intensity of different action units

Confusion co-occurs significantly more with thinking ($\mu = 0.4$) more than writing ($\mu = 0.16, p = 0.0000035$), finding ($\mu = 0.21, p = 0.0023$) and fixing ($\mu = 0.18, p = 0.0002$). Meanwhile, frustration co-occurred with finding ($\mu = 0.2$) and fixing bugs ($\mu = 0.2$) more than it co-occurred with writing ($\mu = 0.17$), but the difference is not significant.

Document insertions occurred significantly more when students were engaged ($\mu = 0.65$) then when they were confused ($\mu = 0.34, p = 0.000000011$), frustrated ($\mu = 0.35, p = 0.000046$) or bored ($\mu = 0.25, p = 0.00078$). Document deletions occurred significantly more when students were engaged ($\mu = 0.13$) than when they were confused ($\mu = 0.09, p = 0.0063$) or bored ($\mu = 0.07, p = 0.0033$). Overall, document changes occurred significantly more when students are engaged ($\mu = 0.77$) then when they are confused ($\mu = 0.44, p = 0.000000047$), frustrated ($\mu = 0.55, p = 0.0055$) or bored ($\mu = 0.38, p = 0.0015$). These findings suggest that document changes were indicative of engagement, and supports our previous study in which confusion was classified using hidden Markov models with log-based features (Tiam-Lee and Sumi 2018).

AU04 - Brow Lowerer AU04 is an action unit referred to as the "Brow Lowerer". As the name implies, it refers to the lowering of the eyebrow. Figure 5 shows some examples of this action unit.

AU04 was ranked highly in RELIEF-F feature ranking for classification tasks for engagement, confusion, and frustration. Students exhibited this action unit in the data when furrowing the brow, and also when looking down to the keyboard repeatedly when typing. In our data, there was an increased observation of AU04 on Japanese students because they tend to look down on the keyboard more than the Filipino students.

Upon further analysis, it can be seen that the mean intensity of AU04 increases in moments of confusion and frustration for both typing and non-typing intervals (see Table 8). This finding supports previous studies that have associated AU04 with confusion and frustration (Bosch, Chen, and D'Mello 2014; Grafsgaard, Boyer, and Lester 2011).

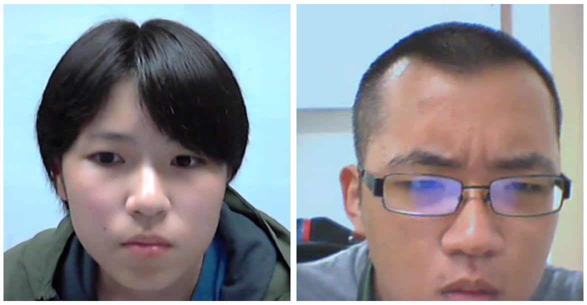


Figure 5: Displays of AU04 (Brow Lowerer). The left image is a Japanese student with a slight AU04 display, while the right image is a Filipino student with a stronger AU04 display.

Table 8: Mean Intensity Display of AU04 in Typing and Non-typing Intervals

	Japanese	Filipino
Engaged, not typing	0.64	0.21
Confused, not typing	0.66	0.22
Frustrated, not typing	0.98	0.31
Engaged, typing	0.89	0.21
Confused, typing	0.65	0.29
Frustrated, typing	1.35	0.35

We also performed a Wilcoxon signed-ranked test to determine if there was a difference in the mean intensity of AU04 on intervals of frustration compared to other affective states, and found that there was indeed a significant difference. The mean intensity display of AU04 in intervals of frustation across all groups is 1.24, while the mean intensity of the display of AU04 in all other intervals is 0.99, with a p value of 0.0065, indicating a significant difference.

Head Rotation Standard Deviation Another feature that was ranked highly is the standard deviation of the head location with respect to the camera. We found that the mean standard deviation of the head location tends to be higher across both groups in intervals of boredom. This suggests that there is a bigger range of head movement when students are bored. The mean standard deviation values of the head location features are shown in Table 9.

Discussion

In this study, we looked into the affective experience of students while using a system for programming practice. This system did not provide any learning interventions such as learning prompts or hints to help the students. It only provided a basic interface to facilitate solving the programming exercises. Thus, it could be said that the environment used in our experiment was similar to that of students doing programming practice on their own without any guidance. This

Table 9: Head Location Average Standard Deviation Across Different Emotions

	Engaged	Confused	Frustrated	Bored
Japanese Group				
location X	12.10	15.75	15.00	**20.87**
location Y	10.16	10.55	11.58	**15.97**
location Z	13.91	15.67	16.67	**23.27**
Filipino Group				
location X	14.14	14.06	14.34	**17.25**
location Y	9.80	9.02	9.01	**10.68**
location Z	10.75	11.73	11.96	**15.74**

was different from previous similar studies in which learning interventions and prompts were given to elicit emotions.

Despite this, we found that students still experienced learning-specific emotions all throughout the sessions, and that they transition between these emotions. We were able to confirm transitions in the theoretical model of affect dynamics for complex learning tasks, which show that engagement generally transitions to confusion when hurdles are encountered, and confusion can either transition back to engagement if the confusion is resolved, or transition to frustration if not resolved.

On average, we found that frustration accounted for around a fifth of all the emotions experienced. This could have been potentially addressed if confusion could be resolved through tutor intervention. This is supported by our findings that students who performed better (i.e. solved more problems) experienced less negative affective states like frustration and boredom, and at the same time experienced more engagement. This shows that there is potential for intelligent programming tutors to improve the learning experience of students.

Transitions between affective states were often observed during code compilations and submissions. These are also the points in the session where the system gives feedback (i.e., displays if the output of the code or displays if the submission passed or failed). This implies that changes in the affective state could be more easily triggered by system feedback. And if appropriate interventions could be displayed, intelligent programming tutors could potentially control transitions of negative affective states to more positive ones.

We also looked into the features that could be useful for predicting affect, and found statistical evidence that associated certain log-based and face-based features in recognizing certain emotions. We found that document changes (typing), compilations, AU04 (lowering of the brow), and head location standard deviation could be useful features for predicting affect. Face-based features alone are difficult to use in affect recognition, as was shown in previous studies, but combining them with log-based features and a mdoel of affect occurrence and transition could potentially improve performance.

We conducted the same experiment on two different universities, and we were able to achieve very similar reuslts, adding support to the idea that these observations on the affective experience of students are consistent across different environments.

That being said, there are some differences between the two groups observed in our data. For example, in our experiment the Japanese students reported the ”neutral” (no apparent feeling) emotion more than the Filipino students, despite the same definition of the affective state labels being provided to the two groups. It is difficult to say whether this was because the Japanese students really felt less emotions or because they tend to be more reluctant to report their emotions.

Another noticeable difference that can have implications in the implementation of ITS is that the Japanese students tend to have higher intensities of AU04 (brow lowerer) compared to the Filipino students. Upon closer inspection, this was because the Japanese students tend to look down at keyboard more while typing, causing their eyebrows to move downwards, which was being detected as AU04. Considerations like this have to be made when designing systems for practical use.

Conclusion

In this paper, we presented an analysis of the affective experience of students while interacting with a system for programming practice. We believe that our findings can provide insights in the development and implementation of affect-aware intelligent tutoring systems for programming.

Acknowledgments

The authors would like to thank Mr. Fritz Kevin Flores, Mr. Manuel Carl Toleran, and Mr. Kayle Anjelo Tiu for facilitating the data collection sessions in De La Salle University in the Philippines.

References

Baltrusaitis, T.; Zadeh, A.; Lim, Y. C.; and Morency, L.-P. 2018. Openface 2.0: Facial behavior analysis toolkit. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, 59–66. IEEE.

Baltrušaitis, T.; Mahmoud, M.; and Robinson, P. 2015. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 6, 1–6. IEEE.

Bosch, N., and D'Mello, S. 2013. Sequential patterns of affective states of novice programmers. In *The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)*, 1–10.

Bosch, N.; Chen, Y.; and D'Mello, S. 2014. Its written on your face: detecting affective states from facial expressions while learning computer programming. In *International Conference on Intelligent Tutoring Systems*, 39–44. Springer.

Bosch, N.; D'Mello, S.; and Mills, C. 2013. What emotions do novices experience during their first computer programming learning session? In *International Conference on Artificial Intelligence in Education*, 11–20. Springer.

Cho, M.-H., and Heron, M. L. 2015. Self-regulated learning: the role of motivation, emotion, and use of learning strategies in students learning experiences in a self-paced online mathematics course. *Distance Education* 36(1):80–99.

Daniels, L. M.; Stupnisky, R. H.; Pekrun, R.; Haynes, T. L.; Perry, R. P.; and Newall, N. E. 2009. A longitudinal analysis of achievement goals: From affective antecedents to emotional effects and achievement outcomes. *Journal of Educational Psychology* 101(4):948.

D'Mello, S., and Graesser, A. 2012a. Autotutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2(4):23.

D'Mello, S., and Graesser, A. 2012b. Dynamics of affective states during complex learning. *Learning and Instruction* 22(2):145–157.

D'Mello, S. 2012. Monitoring affective trajectories during complex learning. In *Encyclopedia of the Sciences of Learning*. Springer. 2325–2328.

Ekman, P., and Friesen, W. V. 1975. Unmasking the face: A guide to recognizing emotions from facial cues.

Grafsgaard, J. F.; Boyer, K. E.; and Lester, J. C. 2011. Predicting facial indicators of confusion with hidden markov models. In *International Conference on Affective Computing and Intelligent Interaction*, 97–106. Springer.

Jaques, N.; Conati, C.; Harley, J. M.; and Azevedo, R. 2014. Predicting affect from gaze data during interaction with an intelligent tutoring system. In *International Conference on Intelligent Tutoring Systems*, 29–38. Springer.

Mega, C.; Ronconi, L.; and De Beni, R. 2014. What makes a good student? how emotions, self-regulated learning, and motivation contribute to academic achievement. *Journal of Educational Psychology* 106(1):121.

Tiam-Lee, T. J., and Sumi, K. 2018. Adaptive feedback based on student emotion in a system for programming practice. In *International Conference on Intelligent Tutoring Systems*, 243–255. Springer.

Zatarain-Cabada, R.; Barrón-Estrada, M. L.; Camacho, J. L. O.; and Reyes-García, C. A. 2014. Affective tutoring system for android mobiles. In *International Conference on Intelligent Computing*, 1–10. Springer.