

The Affective Experience of Novice Computer Programmers

Nigel Bosch¹, Sidney D'Mello^{1,2}, *Departments of Computer Science¹ and Psychology²,
University of Notre Dame, Notre Dame, IN 46556, USA*

Abstract. Novice students (N = 99) participated in a lab study in which they learned the fundamentals of computer programming in Python using a self-paced computerized learning environment involving a 25-min scaffolded learning phase and a 10-min unscaffolded fadeout phase. Students provided affect judgments at approximately 100 points (every 15 seconds) over the course of viewing videos of their faces and computer screens recorded during the learning session. The results indicated that engagement, confusion, frustration, boredom, and curiosity were the most frequent affective states, while anxiety, happiness, anger, surprise, disgust, sadness, and fear were rare. Confusion + frustration and curiosity + engagement were identified as two frequently co-occurring pairs of affective states. An analysis of affect dynamics indicated that there were reciprocal transitions between engagement and confusion, confusion and frustration, and one way transitions between frustration and boredom and boredom and engagement. Considering interaction events in tandem with affect revealed that constructing code was the central activity that preceded and followed each affective state. Further, confusion and frustration followed errors and preceded hint usage, while curiosity and engagement followed reading or coding. An analysis of affect-learning relationships after partialling out control variables (e.g., scholastic aptitude, hint usage) indicated that boredom ($r = -.149$) and frustration ($r = -.218$) were negative correlated with learning while transitions between confusion \rightarrow frustration ($r = .103$), frustration \rightarrow confusion ($r = .105$), and boredom \rightarrow engagement ($r = .282$) were positively correlated with learning. Implications of the results to theory on affect incidence and dynamics and on the design of affect-aware learning environments are discussed.

Keywords. Affect, computer science education, intelligent tutoring systems

INTRODUCTION

Computer science (CS) is a difficult degree to complete and has some of the highest attrition rates among undergraduate majors in the U.S. (Haungs, Clark, Clements, & Janzen, 2012). To address this issue, researchers have attempted to identify the factors that contribute to the eventual or failure in computer programming classes. Some of this research has focused on individual differences, such as mathematical ability, programming aptitude, and psychological traits of temperament and motivation (Alspaugh, 1972; Blignaut & Naude, 2008; Law, Lee, & Yu, 2010; Shute & Kyllonen, 1990). Many of these factors are somewhat influential in predicting a student's decision to enroll as well as their eventual success in computer programming courses. However, these trait-based attributes are very coarse grained and assume fixed dispositions. More fine-grained, person-in-context factors may provide additional insights for understanding outcomes in computer programming courses. The present paper focuses on one such factor – the affective states that students experience during their first encounter with computer programming.

Our working hypothesis is that affective factors play an instrumental role in the process of learning to program and can influence both immediate (failing to solve current problem) and long-term outcomes (failing an exam and dropping out of a CS course). A state of engaged concentration

(and perhaps flow) is hypothesized to be the ideal affective state for learning (Csikszentmihalyi, 1990). However, it is difficult to consistently maintain a state of engagement during computer programming because the experience is punctuated by failure and its resultant negative emotions. For example, confusion and frustration arise when output does not match expectations (confusion) or when the student gets stuck in a logical impasse (frustration). Persistent failure is associated with frustration (Burlison & Picard, 2004) and lower self-efficacy, which can lead to boredom (D’Mello & Graesser, 2012), and ultimately attrition (Larson & Richards, 1991).

The long-term goal of this research is to develop advanced learning environments for CS education. Various strategies, such as game-based learning (Min, Mott, & Lester, 2014) and adaptive materials (Weber & Brusilovsky, 2001) have been used to improve the learning experience for CS students. Given the importance of affect to learning (Pekrun & Linnenbrink-Garcia, 2014), one promising strategy is to develop interfaces that are mindful of student affect while they learn computer programming (D’Mello, Blanchard, Baker, Ocumpaugh, & Brawner, 2014; D’Mello & Graesser, 2015). However, much more basic research on students’ affect is needed before such affect-aware learning environments can be successfully engineered. As a step in this direction, the present study addresses five basic aspects of student affect during their first encounter with computer programming: 1) incidence of affective states; 2) co-occurring affective states; 3) transitions between affective states; 4) relationship between affect and interaction events; and 5) correlations between affect during scaffolded learning and later performance¹

Research Question 1 (RQ1). Affect Incidence

Affect has been well-studied during learning with technology. In a recent meta-analysis of 24 studies that involved learning with technology (D’Mello, 2013), engagement was consistently found to be very frequent across multiple learning contexts. Boredom, confusion, curiosity, happiness, and frustration occurred frequently in some studies while anger, anxiety, contempt, delight, disgust, fear, sadness, and surprise were infrequent. However, none of these studies concerned computer programming as the learning activity. Some researchers (e.g., Khan, Hierons, & Brinkman, 2007; Rodrigo et al., 2009) have studied affect during computer programming. For example, Rodrigo et al. (2009) studied the affective states of computer programming students and reported that flow (engagement) occurred most frequently, followed by confusion, neutral, and then frustration. The focus of these previous studies has been on coarse-grained affect reports and long-term relationships between affect and performance. Here, we examine fine-grained (15-second interval) student affect in an attempt to study affect incidence of novice students first encounter with programming.

RQ2. Affect Co-occurrence

Previous work has provided some important insights into the affective states that arise when students learn with technology (D’Mello, 2013). These studies monitored discrete affect (e.g., confusion, frustration, etc.) at multiple points in a learning session, but only one affective state was tracked at each time point (D’Mello, 2013). The implicit assumption here is that affective states individually occur rather than co-occur. We extend this work by studying affect co-occurrence, or when multiple

¹ This paper advances our previous work by expanding a previously collected dataset of 29 students with an additional 70 students for analyses (1) and (3) (Bosch & D’Mello, 2013; Bosch, D’Mello, & Mills, 2013), as well as adding new analyses to address (4) and (5). Analyses pertaining to (2) were presented in Bosch & D’Mello (2013) and are reproduced here for completeness.

affective states are experienced at the same time. It should be noted that previous research has explored affect transitions, where the emphasis is on the change from one affective state to another as discussed in more detail below (Baker, Rodrigo, & Xolocotzin, 2007; Bosch & D’Mello, 2013; D’Mello & Graesser, 2012). Co-occurrence is different because the emphasis is on multiple affective states that occur at the same time rather than in sequence. One exception is a study by Harley et al. (2012) which investigated co-occurring affective states in the domain of human anatomy education. They used commercial affect recognition software to measure affect. They found that happiness and sadness and sadness and disgust frequently occurred together. The co-occurrence of happiness and sadness is rather surprising and inconsistent with theory given that these affective states have opposite valence profiles (happiness is positive while sadness is negative) (Pekrun & Stephens, 2012). Similarly, sadness and disgust, though both negative, have opposing activation levels (sadness is a deactivating state while disgust is an activating state). These inconsistencies raise the question of whether the co-occurrence relationships uncovered might be attributed to inaccuracies in automated affect detection, which is a well-known problem in the field of affective computing (Calvo & D’Mello, 2010).

RQ3. Affect Transitions

This paper also explores the sequence of affective states throughout time by testing a theoretical model on affect dynamics that has been proposed for a range of complex learning situations (D’Mello & Graesser, 2012). The model (Fig. 1) posits four affective states that are crucial to the learning process: engagement, confusion, frustration, and boredom. It predicts an interplay between confusion and engagement, whereby a learner in the state of engagement may encounter an impasse and become confused. If an impasse is resolved the learner will return to the state of engagement. On the other hand, frustration is triggered when the source of the confusion is not resolved. Frustration can also lead to confusion if new impasses are encountered, but can transition into boredom when frustration is persistent. Further, boredom can transition back into frustration when learners are forced to persist in the learning session despite their boredom.

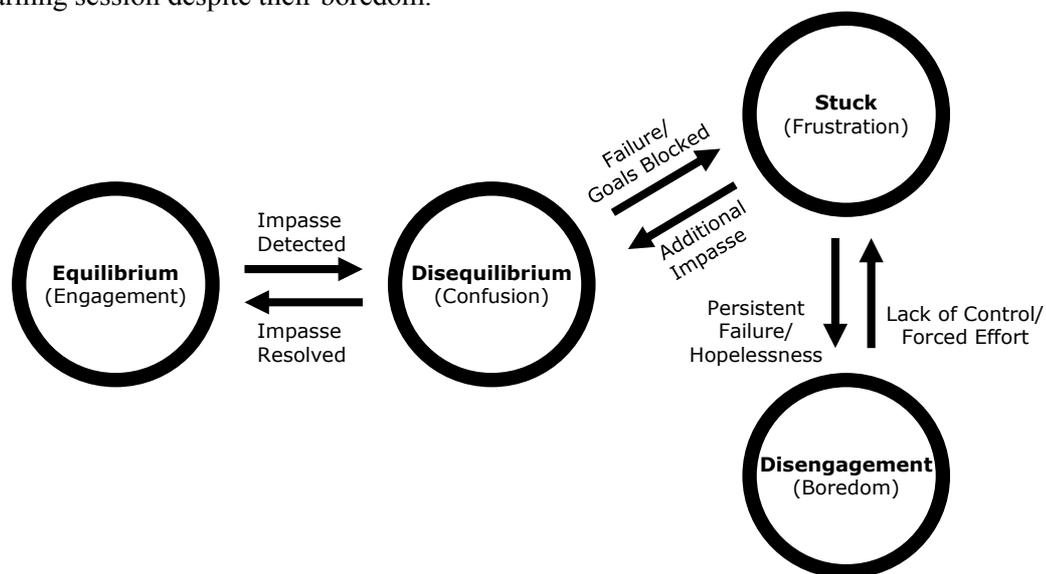


Fig. 1. Theoretical model of affect transitions.

Researchers have found some support for this model during learning with an ITS (D'Mello & Graesser, 2012), during self-guided undergraduate, masters, and doctoral research (Inventado, Legaspi, Cabredo, & Numao, 2012), and during interactions with narrative learning environments (McQuiggan, Robison, & Lester, 2008). We expect the theoretical model to apply to computer programming as well. We posit that encountering unfamiliar concepts, syntax and runtime errors, and other impasses can cause confusion. When those impasses are resolved, the student will be better equipped to anticipate and handle such impasses in the future. Alternatively, if the impasses persist, students may become frustrated and eventually disengage, entering a state of boredom. These possibilities will be tested in the present research.

RQ4. Transitions Between Affect and Interaction Events

We also examine sequences of affective states and interaction events in order to identify how particular interaction events (e.g., errors) influence specific affective states (e.g., frustration) and how affective states engender particular interaction events (e.g., hint request). Some previous work (Hosseini, Vihavainen, & Brusilovsky, 2014; Jadud, 2005; Rodrigo, Baker, Jadud, et al., 2009) examined interaction patterns during computer programming albeit without explicitly considering affect. For example, Rodrigo et al. (2009) analyzed student interaction patterns in a programming environment. Errors, such as consecutive source code compilations with the same error, were negatively related with performance as one might expect. D'Mello et al. (2009) studied transitions between affect and interaction patterns while students solved analytical reasoning problems. Their results indicated that students often became frustrated or bored (among other negative affective states) when provided with negative feedback, while happiness and eureka moments more often followed positive feedback. We apply a similar methodology in this paper, interleaving the affective states and interaction events by timestamp in an attempt to identify frequently occurring transitions between affective states and interaction events.

RQ5. Relationships between Affect and Learning

In our fifth question, we investigate the relationships between affective states and learning. Previous work with computer programming novices suggests that affective states are related to performance. Lee et al. (2011) found that confusion was negatively correlated with midterm exam score. Rodrigo et al. (2009) also found confusion and boredom were negatively related to midterm exam performance, while flow (engagement) was positively correlated with performance. More recently, Grafsgaard et al. (2012) collected several data sources while students conversed with a human tutor via a computer-mediated interface. Coarse-grained frustration reported by students was correlated ($r = .53$) with student confusion observed by the tutor. Additionally, tutor reports of student confusion and frustration were correlated ($r = .59$), and confusion was negatively correlated with posttest scores ($r = -.38$). In the present paper, we study how students affect experience (i.e., affect incidence, affect co-occurrence, and affect transitions) observed during a scaffolded learning phase correlate with performance on a subsequent fadeout phase after controlling for a number of factors (e.g., hint usage, demographics).

Current Study

The present study builds upon and extends previous research in this area (reviewed above) in the following three ways. First, previous work has taken an ecological approach to studying affect during computer programming in authentic learning contexts. This approach has obvious merits but is limited

with respect to the relatively coarse-grained nature of affect measurements. Taking a somewhat different approach, we track affect at a fine-grained level (every 20 seconds). Also, while much of the previous work has studied students enrolled in computer science classes, we focus on novices. This is because having basic computer programming skills is essential in the 21st century digital age. We accomplished this by carefully screening students to remove those with prior programming experience and those who are majoring in computer science. Third, our focus is on one-on-one human-computer programming experiences without interference, distractions, or social pressures that may apply when teachers or peers are involved in the learning process. This necessitated the high control of the lab, so we conducted a laboratory study in this early stage of the research. The hope is that insights gleaned from the present fine-grained lab study with non CS students can complement previous findings from coarse-grained ecological studies with CS students, thereby providing a more comprehensive understanding of students' affective experiences while learning computer programming.

METHOD

Participants

Participants (called students) were 113 students from a private Midwestern university in the United States. Fourteen students were removed because they reported having prior experience with computer programming and our intended focus was on novices only. Of the remaining 99 students, 49.5% were female and the mean age was 19.3 years ($SD = 1.12$ years). The students represented 25 majors including psychology, biology, architecture, marketing, and others, so there was considerable diversity (at least in terms of major) in the sample. Data collection took place over the course of two semesters. Data from cohort 1 ($N = 29$) was collected in Fall 2012 while data for cohort 2 ($N = 70$) was collected in Spring 2013. There were minor methodological differences between the two cohorts as detailed below.

Procedure

Students were individually tested in a two-hour session. The study consisted of three main phases (scaffolded learning; fadeout; and retrospective affect judgment) as discussed below. A webcam on the monitor recorded the face of students, while screen capture software recorded videos of the learning environment (see below). Students were not informed of the purpose of the research before beginning. Instead, they were informed that the goal was to test a new learning environment for novice computer programmers. Details of the purpose of the study were revealed to students only at the end of their two-hour session.

Learning Environment

Students were taught fundamentals of computer programming in the Python language, using a researcher-built computerized learning environment. Fig. 2 shows a screenshot of the learning environment used by students. Numbers overlaid in Fig. 2 indicate the different areas of the learning environment interface: 1) instructional text, 2) source code editing box, 3) hint display area, and 4) input/output console. Students in cohort 1 could freely interact with all four areas at the same time. However, cohort 2 students could only interact with only one area of the interface at a time. Specifically, each area could be made visible by clicking a button for that area, which would then hide the previously used area of the interface. This was done to disambiguate students' current interaction activity (i.e., determine if they were reading, viewing the hint, coding, or testing their code). Instructions for using the interface were provided to students. The learning environment kept logs of

interaction events including both student actions (e.g., key presses, button presses) and system actions (e.g., providing feedback on code correctness).

The screenshot shows a learning environment window titled 'CSL'. It is divided into several sections:

- 1**: A text area on the left containing instructional text about variables and Python code examples:

```
pi = 3.142
2 * pi * 3959
pi * 3959 * 3959
```

But you could also do the same thing for the radius, since it needs to be used several times:

```
pi = 3.142
r = 3959
2 * pi * r
```
- 2**: A code editor on the right with two lines of Python code:

```
1 side=556.6
2 side * side * side
```
- 3**: A 'Hint 1' section at the bottom left with the text: 'Set up the variable like this: side = 556.6 and remember that the formula for the volume of a cube is side³.' Below this is a 'Show hint 2/3' button.
- 4**: An output area at the bottom right showing the result of running the code: '172436661.496' followed by '***Script Ended***'. Below this is a 'Submit' button.

At the bottom right of the window, a status bar displays: 'You have 21 points so far' and 'This exercise is worth 3 points'.

Fig. 2. Screenshot of the learning environment used by students, with key areas numbered.

Learning Procedure

Students completed a 25 minute *scaffolded phase*, in which they had access to instructional materials, exercises to solve, and hints (both cohorts). This was followed by a 15-minute (cohort 1) or 10-minute (cohort 2). *fadeout phase*. The goal of the scaffolded phase was to provide foundational knowledge that could be applied in the fadeout phase while the goal of the fadeout phase was to assess learning.

Scaffolded phase. The scaffolded phase consisted of a set of 19 programming exercises and was timed at 25-minutes. Exercises covered syntax for arithmetic concepts (addition, multiplication, subtraction, exponentiation), geometry (volume, area, perimeter), and basic programming concepts (variables, reading from standard in, printing output, and integer vs. floating point numbers). Each exercise had a problem statement, an explanatory text, and a set of hints. Students needed to write working Python code to solve the problem in each exercise. The exercises were predominately math-based geometry problems with numeric inputs. This topic was chosen because it is often used in introductory programming courses. An example of an exercise is as follows: “Suppose you want to calculate the mileage you are getting in your car easily. Create a program to assist in this, first by

prompting for *Miles driven:* and then *Gallons of gas used:* Store each of these values in a variable and print out the resulting miles per gallon.” This exercise represents an incremental step from reading input and storing it as a variable (previous exercise) to reading two different inputs into different variables (current exercise).

Students were able to test their code with the interactive console, and submit code for automatic correctness checking when they were satisfied with their work. If a submitted solution was correct, the student would automatically be advanced to the next exercise. Otherwise, the learning environment would tell the student their solution was incorrect, and suggest using a hint or trying again. There was no limit on number of submission attempts allowed. Correctness was determined by comparing the output of the students’ code with the output of a predetermined correct solution, allowing acceptable variations such as different precision of π in geometry-based solutions. Additionally, solutions to exercises that required reading input were tested by automatically providing different input values and checking for corresponding correct outputs.

Hints were available for each scaffolded exercise. Hints ranged from further instructional explanation of the key concept(s) in an exercise, code examples illustrating the concept(s), up to complete solutions for an exercise (bottom-out hint). Hints were made available after a time delay ranging from 45 to 90 seconds relative to the start of the exercise or the previous hint request. Time delays were based on the anticipated difficulty of the exercise and previous hints. Longer delays were used to require more processing of complicated concepts. The possible score for each exercise was set to be the number of hints for that exercise plus one. Using a hint resulted in a deduction of one point from the exercise. For example, an exercise with three available hints could be worth as much as four points (no hints used) or as little as one (all hints used).

Fadeout phase. Following the scaffolded phase, students completed a *fadeout programming phase*. The fadeout exercise made use of all major concepts that could be covered in the scaffolded phase. It was designed to be more difficult than novice students would be capable of solving, though they could make progress toward a solution. No hints or explanation were available during the fadeout phase in order to encourage unscaffolded problem solving and assess learning. Students in cohort 1 also completed a five-minute debugging exercise. However, the debugging exercise was removed from cohort 2 because it proved too short for meaningful analysis. The present analyses only focus on the 10-minute fadeout programming exercise and the 25-minute scaffolded phase since these were consistent across both cohorts.

Affect Judgments (Phase 3)

We measured students’ affective states using a retrospective judgment protocol (Rosenberg & Ekman, 1994), which is a validated offline affect-judgment technique that affords fine-grained affect measurement without any interruptions during the learning session (see review of affect annotation methods [Porayska-Pomsta, Mavrikis, D’Mello, Conati, & Baker, 2013]). The protocol commenced after the fadeout phase of the study. Students were shown synchronized videos of their own face and on-screen activity (from screen capture videos) and were asked to make judgments about what affective states they were experiencing at various points in the learning session. Thus, affective judgments were based on a combination of context (as given by screen capture video), facial cues, and memories of the learning session. Fig. 3 shows an illustration of the interface used for retrospective affect judgment.

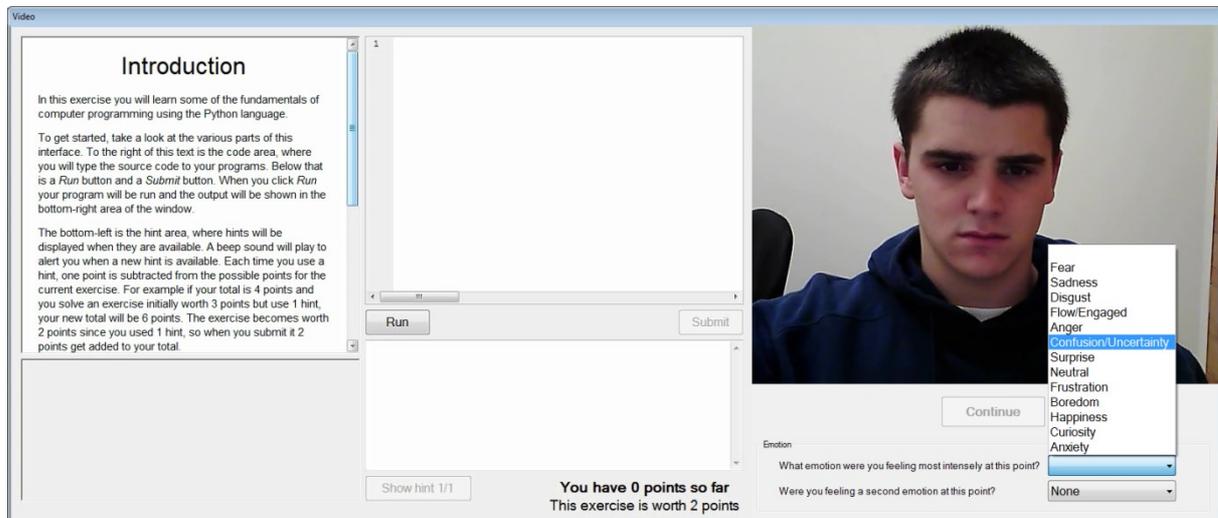


Fig. 3. Retrospective affect judgment interface.

Students were prompted to provide affect at 100 randomly chosen fixed points at which the videos automatically paused. Judgment points corresponded with to interaction events, such as key presses, running code, showing hints, and other such occurrences. Some periods of idle activity (longer than 30 seconds) were also chosen for affect judgments. In addition to the 100 fixed points, students could spontaneously pause the video streams and provide an affect judgment at any time.

Students selected their affective states at each point from a randomly ordered list comprised of anger, anxiety, boredom, confusion/uncertainty (henceforth abbreviated as confusion), curiosity, disgust, fear, frustration, flow/engagement (henceforth abbreviated as engagement), happiness, sadness, surprise, and the neutral state (defined as no apparent emotion). These states are largely derived from Pekrun's taxonomy of academic emotions (Pekrun & Stephens, 2012) and from previous work on affect during learning with technology (D'Mello, 2013). Students were required to choose a primary affective state at each judgment point. Students could also voluntarily provide a secondary judgment—a co-occurring affective state they were experiencing at that point.

It is important to mention three points pertaining to the affect judgment methodology. This procedure was adopted because it affords monitoring students' affective states at multiple points, with minimal task interference, and without students knowing that these states are being monitored while they complete the learning task. Second, this retrospective affect-judgment method has been previously validated (Rosenberg & Ekman, 1994). Analyses comparing these offline affect judgments with online measures including self-reports and observations by judges have produced similar affect profiles (Craig, D'Mello, Witherspoon, & Graesser, 2008; Craig, Graesser, Sullins, & Gholson, 2004). Third, the offline affect annotations obtained via this protocol correlate with online recordings of facial activity and body movements in expected directions (D'Mello & Graesser, 2010). Although no method is without its limitations, the present method appears to be a viable approach to track affect at a relatively fine-grained temporal resolution.

Assessing Performance and Learning

Students could complete as many exercises as possible within the time limit for the scaffolded phase before being automatically directed to the fadeout phase. On average, students completed 13.3 scaffolded exercises ($SD = 4.01$). The students' cumulative score (exercises completed + hints not used; see above) was used as a measure of performance in the scaffolded phase. The highest possible score was 67, while the lowest possible score was a 0. Mean scaffolded score was 41.4 ($SD = 12.6$). Scores for the fadeout phase were calculated differently since there was only one exercise and no hints. Instead, two trained judges considered the number of lines of code in a student's solution that corresponded semantically to lines in a "correct" solution (maximum = 11). The human judges independently scored every solution and resolved any differences via discussion. The mean fadeout score was 5.95 ($SD = 3.41$).

RESULTS AND DISCUSSION

The results are organized with respect to the five main research questions articulated in the Introduction. Due to their similarity and to increase the sample size, data from the two cohorts was pooled together for analysis.

Affect Incidence

A total of 9,696 affect judgments were obtained from the 99 students. The analyses proceeded by computing proportion scores for each student's *primary* affective state reports only; secondary affect reports are examined in the co-occurrence analyses presented next. The distribution of affect proportions violated assumptions of normality, so nonparametric tests are used for the analyses reported below. That being said, students were not reporting the same affective state every time because the maximum proportion score was 0.74 (one student reported neutral 74% of the time). Table 1 presents mean proportions of affect reports overall and across the two phases of the study.

Table 1. Mean (SD) proportion of affective reposts

Affective State	Overall	Scaffolded	Fadeout
Engagement	.244 (.176)	.210 (.174)	.329 (.266)
Confusion	.206 (.114)	.207 (.107)	.205 (.187)
Frustration	.120 (.088)	.118 (.096)	.123 (.129)
Boredom	.090 (.122)	.088 (.124)	.100 (.182)
Curiosity	.071 (.062)	.082 (.076)	.043 (.070)
Anxiety	.032 (.042)	.018 (.030)	.068 (.097)
Happiness	.023 (.031)	.030 (.042)	.006 (.016)
Anger	.017 (.032)	.018 (.037)	.013 (.035)
Surprise	.014 (.016)	.016 (.020)	.008 (.017)
Disgust	.008 (.019)	.009 (.021)	.007 (.020)
Sadness	.004 (.010)	.003 (.009)	.006 (.024)
Fear	.001 (.005)	.001 (.007)	.001 (.007)
Neutral	.168 (.156)	.199 (.175)	.089 (.163)

Overall affect. The results indicated that engagement, confusion, frustration, boredom, curiosity (henceforth referred to as frequent affective states) occurred at least 5% of the time and collectively accounted for approximately 73% of all affect judgments. The other affective states (anxiety, happiness, anger, surprise, disgust, fear, and sadness) were infrequent and summatively accounted for only 10% of the affect reports. Neutral (no affect) comprised 17% of the reports. Moreover, Wilcoxon signed rank tests (with a Bonferroni correction of $p < .0012$ [$.05 / [6 \text{ frequent} \times 7 \text{ infrequent affective states}]$] to account for multiple tests) indicated that each frequent affective state and neutral occurred at significantly higher rates than the less frequent affective states. This finding is in line with previous research suggesting that boredom, engagement, confusion, and frustration are the affective states that routinely occur during learning with technology while curiosity occurs frequently in some contexts (D’Mello, 2013). The subsequent analyses focuses on these five states and neutral.

Scaffolded vs. Fadeout phases. We compared the affective states reported during the two phases of the study (scaffolded and fadeout). Six Wilcoxon signed rank tests, one for each frequent affective state (and neutral), revealed that there were significant ($p < .001$) differences across phases for engagement, neutral, and curiosity. Results indicated there was more neutral reported in the scaffolded phase ($M = .199$, $SD = .175$) compared to the fadeout phase ($M = .089$, $SD = .163$), $Z = -6.06$, $p < .001$. Similarly, there was more curiosity reported in the scaffolded phase ($M = .082$, $SD = .076$) compared to the fadeout phase ($M = .043$, $SD = .070$), $Z = -5.50$, $p < .001$. However, there was more engagement in the fadeout phase ($M = .329$, $SD = .266$) compared to the scaffolded phase ($M = .210$, $SD = .174$), $Z = -4.45$, $p < .001$.

Affect Co-occurrence

Students were required to select the affective state they felt most strongly for each judgment (the primary state), but could also optionally provide a secondary affective state if they were experiencing more than one affective state. We examined co-occurring affective states by considering their secondary affect judgments in tandem with the primary judgments. Students that made no secondary judgments ($N = 23$) or made fewer than 10 secondary judgments ($N = 30$) were excluded. There were a total of 1,764 secondary affect judgments provided by the remaining 46 students. Table 2 shows the mean proportions of primary and secondary affect states for these students sorted by primary ratings. Only, anxiety, boredom, confusion, curiosity, engagement, and frustration were commonly ($> 5\%$) reported as secondary affective states. Thus, only these states were considered for subsequent co-occurrence analyses. Although, neutral was occasionally reported as a secondary judgment (5.4%), it was not considered in co-occurrence analyses because it is conceptually similar to a primary affective rating only. Considering only the frequent affective states and only students who reported at least 10 co-occurrences resulted in 1,303 pairs of ratings for subsequent analysis.

Table 2. Mean (*SD*) proportions of affective states reported.

Affective State	Primary	Secondary
Confusion	.301 (.160)	.222 (.156)
Engagement	.237 (.143)	.150 (.109)
Frustration	.167 (.149)	.186 (.085)
Curiosity	.085 (.092)	.099 (.076)
Boredom	.056 (.084)	.088 (.121)
Anxiety	.050 (.067)	.115 (.150)
Neutral	.038 (.055)	.053 (.074)
Surprise	.020 (.037)	.022 (.030)
Happiness	.019 (.033)	.017 (.030)
Anger	.019 (.046)	.023 (.037)
Sadness	.004 (.015)	.006 (.023)
Disgust	.004 (.010)	.014 (.041)
Fear	.001 (.004)	.004 (.022)

What pairs of affective states co-occurred? An association rule learning metric called Lift (Equation 1) (Tan, Kumar, & Srivastava, 2002) was used to compare the observed probability of two co-occurring affective states (numerator) with the probability of those states co-occurring due to chance (denominator). A Lift value higher than 1 indicates a pair of affective states co-occurred more frequently than expected by chance.

$$Lift = \frac{\Pr(X \text{ and } Y)}{\Pr(X)\Pr(Y)} \quad (1)$$

Lift was separately calculated for each student for all pairs of affective states that were frequently reported as both primary and secondary affective states. Table 3 shows the average Lift across all students for each pair of states. Only the confusion + frustration and curiosity + engagement affective state pairs occurred at levels above what was expected by chance (Lift = 1). It should be noted that these two co-occurring pairs are theoretically consistent, while pairs such as boredom + engagement or boredom + confusion do not make theoretical sense.

Table 3. Mean lift (*SD*) for every pair of frequently reported affective states.

	Anxiety	Boredom	Confusion	Curiosity	Engagement	Frustration
Anxiety	-					
Boredom	.280 (.480)	-				
Confusion	.550 (.428)	.600 (.474)	-			
Curiosity	.214 (.343)	.490 (.798)	.507 (.538)	-		
Engagement	.758 (.690)	.447 (.659)	.583 (.323)	1.33 (.984)	-	
Frustration	.495 (.488)	.569 (.659)	1.14 (.593)	.072 (.166)	.221 (.262)	-

Does one affective state in a co-occurring pair imply the other? The dependence of one affective state on the other in these co-occurring pairs may provide some additional information for interpreting their presence. To examine the dependence we used another association rule learning metric called confidence (Equation 2) (Tan et al., 2002). Confidence measures the probability of an affective state Y occurring, given the presence of another affective state X (i.e., to what extent does X imply Y).

$$\text{Confidence } (X \rightarrow Y) = \frac{\text{Pr}(X \text{ and } Y)}{\text{Pr}(X)} \quad (2)$$

The confidences of both possible orderings of the affective states in the two frequently co-occurring pairs were compared to determine if one state in the pair was more likely to imply the other state than vice versa. Table 4 presents the results of comparing the confidences for the two affective state pairs that occur more often than chance with paired-samples *t*-tests. We note that the affective states in a co-occurrence pair did not imply each other equally. Specifically, confusion was more likely to imply frustration than vice versa ($p < .001$) and curiosity was more likely to imply engagement than vice versa ($p < .001$).

Table 4. Comparisons of confidence for affective state pairs. Standard deviations are in parentheses.

Primary → Secondary	Mean (SD)	t
Confusion → Frustration	.419 (.225)	-5.71*
Frustration → Confusion	.672 (.263)	
Curiosity → Engagement	.495 (.359)	4.53*
Engagement → Curiosity	.259 (.190)	

Note. * $p < .001$

Affect Transitions

We previously introduced a theoretical model of affect dynamics that specified a number of transitions between affective states (see Fig. 1). To test this model, we used a previously developed metric (Equation 3) to compute the likelihood of the occurrence of each transition relative to chance (D’Mello, Taylor, & Graesser, 2007). This likelihood metric computes the conditional probability of a particular affective state (*next*), given the current affective state. The probability is then normalized to account for the overall likelihood of the *next* state occurring. If the affective transition occurs as expected by chance, the numerator is 0 and so likelihood is as well. Thus we can discover affective state transitions that occurred more ($L > 0$) or less ($L < 0$) frequently than expected by chance alone.

$$L(\text{Current} \rightarrow \text{Next}) = \frac{\text{Pr}(\text{Next}|\text{Current}) - \text{Pr}(\text{Next})}{1 - \text{Pr}(\text{Next})} \quad (3)$$

Transition likelihoods were computed across from time series of affect sequences (one per student) across both scaffolded and fadeout phases. We removed self-transitions (transitions from a state to the same state) before computing *L* scores. For example, a sequence of affective states such as *confusion, frustration, frustration, boredom* would be reduced to *confusion, frustration, boredom*. This was done because our focus is on transitions between different affective states, rather than on the persistence of each affective state (D’Mello & Graesser, 2012; Inventado et al., 2012). Furthermore, we only focus on transitions between states specified by the theoretical model (boredom, confusion,

engagement, and frustration), which also happen to be among the most frequent affective states in the present data. More specifically, the likelihoods were computed with respect to all affective states (except for removal of self-transitions), but we only analyze transitions involving the four affective states specified in the theoretical model.

We identified the transitions that occurred significantly more than chance ($L = 0$) by computing affect transition likelihoods for individual students and then comparing each likelihood to zero (chance) with a two-tailed one-sample t -test. Significant ($p < .05$) transitions are shown in Fig. 4 and are aligned with the theoretical model on affect dynamics. A Bonferroni correction was not applied because we were testing transitions involving states specified by a theoretical model (Fig. 1) rather than all possible transitions.

The results (see Fig 4 and Table 5) indicated that five of the six predicted transitions, engagement \leftrightarrow confusion, confusion \leftrightarrow frustration, and frustration \rightarrow boredom, were significant and aligned with the theoretical model. The predicted boredom \rightarrow frustration transition was not significant in the present data. Instead of transitioning to frustration, boredom was likely to transition to engagement even though the boredom \rightarrow engagement transition was not predicted by the theoretical model. It is possible that the nature of our computerized learning environment encouraged this transition more than expected. This might be due to the fast-paced nature of the learning session, which included 19 exercises and an in-depth programming task, so boredom might have quickly dissipated. Furthermore, students had some control over the learning environment in that they could use bottom-out hints to move to the next exercise instead of being forced to wallow in their boredom.

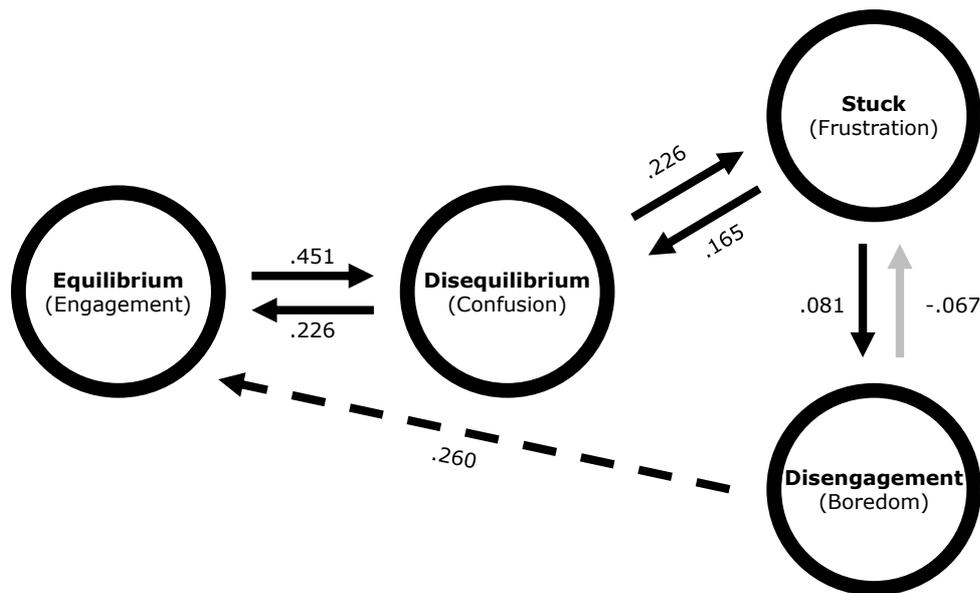


Fig. 4. Frequently observed affective state transitions. Edge labels are mean likelihoods L of affective state transitions. The grey edge represents a transition that was predicted by the theoretical model but was not significant. The dashed edge represents a transition that was not predicted but occurred in our data.

Table 5. Details of frequently observed affective state transitions.

Transition	Mean <i>L</i> (SD)	<i>t</i>
Engagement → Confusion	*.451 (.358)	12.0
Confusion → Engagement	*.226 (.289)	7.76
Confusion → Frustration	*.226 (.276)	8.16
Frustration → Confusion	*.165 (.387)	4.20
Frustration → Boredom	*.081 (.266)	3.02
Boredom → Frustration	-.067 (.343)	-1.79
Boredom → Engagement	*.260 (.466)	5.08
Boredom → Confusion	.057 (.507)	1.02
Confusion → Boredom	.027 (.219)	1.22
Engagement → Boredom	.022 (.231)	0.91
Engagement → Frustration	-.012 (.296)	-0.39
Frustration → Engagement	.036 (.348)	1.01

Note: * indicates $p < .05$

Interestingly, boredom was likely to transition to engagement (mean $L = .260$, $p < .05$) even though the boredom → engagement transition was not predicted by the theoretical model. It is possible that the nature of our computerized learning environment encouraged this transition more than expected. This might be due to the fast-paced nature of the learning session, which included 19 exercises and an in-depth programming task in a short 35-minute session. Furthermore, students had some control over the learning environment in that they could use bottom-out hints to move to the next exercise instead of being forced to wallow in their boredom, unlike a previous study that tested this model using a learning environment (AutoTutor) that did not provide any control over the learning activity (D’Mello & Graesser, 2012).

Transitions Between Interaction Events and Affective States

The analyses so far have examined affective phenomena (incidence, co-occurrence, and transitions) independent of the events occurring in the learning environments. Additional insights can be learned by considering the interaction events that precede and follow affective states. Toward this end, affective states were interleaved with the interaction events shown in Table 6 according to timestamp to provide a continuous sequence of interaction events and affective states. States (either interaction or affect) that repeated were coalesced to a single instance as in the affect-only transition analysis (e.g., *ShowHint*, *ShowHint* becomes simply *ShowHint*). This step was especially important because interaction events such as Coding (triggered with every key press in the code box) occur far more frequently than others.

The L metric was applied in order to compute transitions between interaction events and affective states. Student-level L values for each event-affect pair were compared to chance (zero) using a two-tailed independent samples t -test. Students ($N = 29$) from cohort 1 could not be used in this analysis because they did not have interaction states logged with enough context to disambiguate activities like

reading from hint viewing or thinking during coding. Thus, only the data from the 70 students in cohort 2 were used in this analysis.

We calculated transitions separately for scaffolding vs. fadeout phases because of the different interaction events in the two phases. For example, there was only one exercise in the fadeout phase, and no correct solution was generated, so events like ShowProblem and SubmitSuccess were not relevant in the fadeout phase. Additionally, hints were available in the scaffolding phase but not in the fadeout phase.

Table 6. Description of interaction events.

Interaction Event	Description
Coding	Editing or viewing the solution to the current exercise
Reading	Viewing the instructions and/or problem statement for the current exercise
ShowHint	Viewing a hint for the current exercise
ShowProblem	Starting a new exercise (occurs automatically after SubmitSuccess)
SubmitError	Code submitted for correctness checking and produced an error or wrong answer
SubmitSuccess	Code submitted and was correct
TestRunError	Code was run and encountered a syntax or runtime error
TestRunSuccess	Code run without syntax or runtime errors (but was not checked for correctness)

Affective states were interleaved with the interaction behaviors according to timestamps, providing a continuous sequence of interaction events and affective states as they occurred during the learning session. States (either interaction or affect) that repeated were coalesced to a single instance as in the affect-only transition analysis (e.g., *ShowHint*, *ShowHint* becomes simply *ShowHint*). This step is especially important when considering interaction behaviors because interaction behaviors such as Coding (triggered with every key press in the code box) occur far more frequently than others. *L* was then computed and the resulting transitions were compared using a two-tailed independent samples *t*-test against a test value of 0 to find transitions that occurred more frequently than expected by chance.

Transitions in the scaffolded phase. There were 14 total states (8 interaction events and 6 affective states), resulting in $14 \times 13 = 182$ potential transitions as self-transitions such as Coding \rightarrow Coding were not considered. Fig. 5 illustrates the significant transitions in the scaffolding phase at $p < .000275$ (.05/182 after applying a Bonferroni correction).

involving (a) confusion and frustration and (b) engagement, curiosity, and boredom. In particular, confusion and frustration were both preceded by an incorrect solution submission (SubmitError; $L = .07$, $p < .000275$ for confusion, $L = .09$, $p < .000275$ for frustration) and were followed by a hint request (ShowHint; $L = .07$, $p < .000275$ for confusion, $L = .09$, $p < .000275$ for frustration) or coding, which itself triggered confusion ($L = .06$, $p < .000275$) and frustration ($L = .04$, $p < .000275$). Reading was a precursor of confusion ($L = .05$, $p < .000275$) but not frustration ($L = -.01$, $p > .000275$). These transitions align with the aforementioned theoretical model of affect dynamics in that assimilation (i.e. Reading; $L = .05$, $p < .000275$ transitioning to confusion), generation (Coding; $L = .06$ to confusion, $L = .04$ to frustration, $p < .000275$), evaluation (SubmitError; $L = .07$ to confusion, $L = .09$ to frustration, $p < .000275$), and help-seeking (ShowHint; $L = .07$ from confusion, $L = .09$ from frustration, $p < .000275$) activities continually interact with confusion and frustration. On the other hand, curiosity ($L = .20$ to reading, $L = .22$ to coding, $p < .000275$), engagement ($L = .09$ from reading, $L = .51$ to coding, $p < .000275$), and boredom ($L = .07$ from reading, $L = .44$ to coding, $p < .000275$) were mainly associated with assimilation (Reading) and generation (Coding) activities but not with evaluation (SubmitError; all $p > .000275$) and help-seeking (ShowHint; all $p > .000275$) activities. Finally, the transitions to and from boredom may shed some light on the unexpected boredom to engagement transition that was contrary to the theoretically model (Fig. 4). Boredom transitioned into coding ($L = .44$, $p < .05$), which may have in turn led the student to become re-engaged rather than staying bored.

Transitions in the fadeout phase. Fig. 6 illustrates the frequently occurring transitions in the fadeout phase. The ShowHint and SubmitSuccess events could not occur in the fadeout phase, so 6 affective states and 6 interaction events yielded 132 ($12 \times (12 - 1)$) possible transitions. A Bonferroni correction was applied to test the significance of the fadeout transitions resulting in a significance threshold of .00038 (i.e., $.05/132$).

We note fewer significant transitions in the fadeout phase compared to the scaffolded phase. We suspect that two factors led to the sparseness of fadeout graph shown in Fig 6. First, as discussed above, there were two fewer interaction events, leading to fewer possible transitions. Second, the fadeout phase was only 10-minutes long, resulting in fewer affect observations and a smaller sample size for some transitions. For example, some students reported no boredom during the fadeout phase, leading to a reduced sample size, which provides less statistical power. This might also explain why two expected transitions, TestRunError \rightarrow frustration ($L = .03$) and SubmitError \rightarrow frustration ($L = .04$), were positive but not significant. Nevertheless the key pattern evident in the fadeout phase involves the following cycle: Coding \rightarrow TestRunSuccess \rightarrow SubmitError \rightarrow Coding. This cycle aptly illustrates the exceedingly difficulty of the fadeout exercise, where students were able to run their code without syntax or runtime errors, but could not get the correct answer.

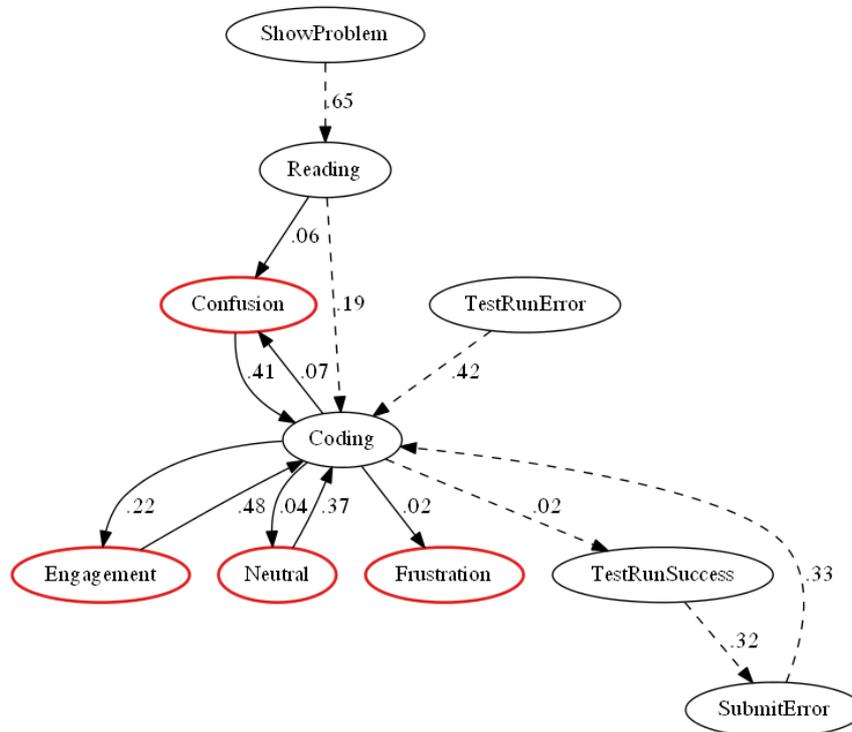


Fig. 6. Transitions between affect and interaction events in the fadeout phase. Solid lines indicate transitions including affect. Dashed lines indicate transitions not involving affective states. Numbers represent L for transitions.

Correlations Between Affect and Learning

Our final analysis focused on understanding the relationship between affective phenomena and learning outcomes. Specifically, we correlated affective phenomena (incidence and transitions) observed in the scaffolded phase with performance during the fadeout phase. The latter was taken to be a measure of learning because it involved unscaffolded coding of a complex novel problem that required application of previously learned concepts.

A number of analytic decisions need to be clarified before presenting the results. First, we partialled out demographics (gender) and scholastic aptitude (self-reported SAT scores that are shown to correlate with actual scores - Cole & Gonyea, 2010) as these variables are known to correlate with performance. Second, we also partialled out the overall score and the number of hints used in the scaffolded phase in order to target unique variance (net of scaffolded performance) in fadeout performance. Third, co-occurring affective states were not considered in these correlations because co-occurrences were derived from both scaffolded and fadeout phases combined in order to maximize the sample size.

The first set of analyses (see Table 6) consisted of partial correlations between affect incidence during scaffolded phase (proportional occurrence of frequent affective states and neutral) and fadeout score (learning measure) after controlling for gender, SAT (Scholastic Aptitude Test, a standard test for university admission in the USA), scaffolded score, and hints used during scaffolded phase. Due to small sample size, we consider correlations around .100 (consistent with a small effect size; Cohen,

1988) as suggestive trends rather than focusing on significance. Consistent with expectations, boredom and frustration were negatively correlated with fadeout score. Engagement, confusion, and neutral showed positive but weak trends.

Table 6. Correlations between scaffolded affect and fadeout performance.

Affective State	Partial Correlation
Boredom	-.149
Confusion	.087
Curiosity	-.026
Engagement	.093
Frustration	-.218
Neutral	.091
Affect Transition	
Engagement → Confusion	.058
Confusion → Engagement	-.007
Confusion → Frustration	.103
Frustration → Confusion	.105
Frustration → Boredom	-.039
Boredom → Engagement	.282

Next we studied correlations between significant affective transitions in the scaffolded phase and fadeout performance. Here, proportions of individual affective states were also partialled out in addition to gender, SAT, and scaffolded performance and hint usage. For example, proportions of engagement and confusion in the scaffolded phase were partialled out for the engagement → confusion transition. The resulting partial correlations are in Table 6. We note that confusion → frustration (partial $r = .103$) and frustration → confusion (partial $r = .105$) transitions positively correlated with performance. These transitions are indicative of students being in the throes of problem solving where they experience impasses, challenges, and failure. The boredom → engagement transition was also positively correlated with fadeout performance, indicating that the ability to re-engage from boredom is positively predictive of performance.

GENERAL DISCUSSION

Computer programming is a challenging but essential skill for computer science education. Understanding the experience of novice students will be helpful for developing adaptive computerized learning environments. This paper takes a step in this direction with an emphasis on student affect. We performed fine-grained analyses on student affect during their first programming lesson in order to advance basic research and apply any insights gleaned to develop automatic interventions that respond to affect in addition to cognition. Our emphasis was on identifying frequent affective state and understanding how these states are related to each other, to events in the learning session, and to

performance in the learning task. In this section we discuss our main findings with respect to the five research questions posed, and discuss implications, limitations, and future work.

Main Findings

Our first research question concerned the incidence of affective states. A recent meta-analysis found that engagement occurred more frequently than chance during learning with technology (D’Mello, 2013), while confusion, frustration, boredom, curiosity, and happiness varied across studies. Affective states such as contempt, anger, and others were infrequent. In the current study, we found that engagement, confusion, frustration, boredom, and curiosity were the dominant affective states reported by novice programming students. This finding aligned with previous research outside of programming and suggests that future research should primarily focus on these states.

Our second research question concerned co-occurring affective states. We discovered that the co-occurrence was infrequent in general. When affect did co-occur there were two stable co-occurrence patterns: confusion + frustration; curiosity + engagement. These findings suggest that there might be a need to revise the aforementioned theoretical model to incorporate co-occurrence relationships between confusion and frustration and between curiosity and engagement.

Our third research question concerned transitions between affective states. We tested a theoretical model of affective dynamics during complex learning (D’Mello & Graesser, 2012). The model focuses on the role of impasses in triggering confusion and other affective states. Impasses commonly arise in computer programming, particularly when novices encounter unfamiliar concepts, syntax errors, and unexpected output. The model posits that unresolved impasses can lead to frustration, which can and eventually lead to boredom. The observed affect transitions largely aligned with this theoretical model, although there were two exceptions (i.e., no evidence for the expected boredom → frustration transition and evidence for the unexpected boredom → engagement transition). This suggests that our theoretical model might need to be revised to incorporate a possible re-engagement link from boredom in lieu of the boredom to frustration link. Fig. 7 presents an updated theoretical model incorporating the new affect transition as well as co-occurrences.

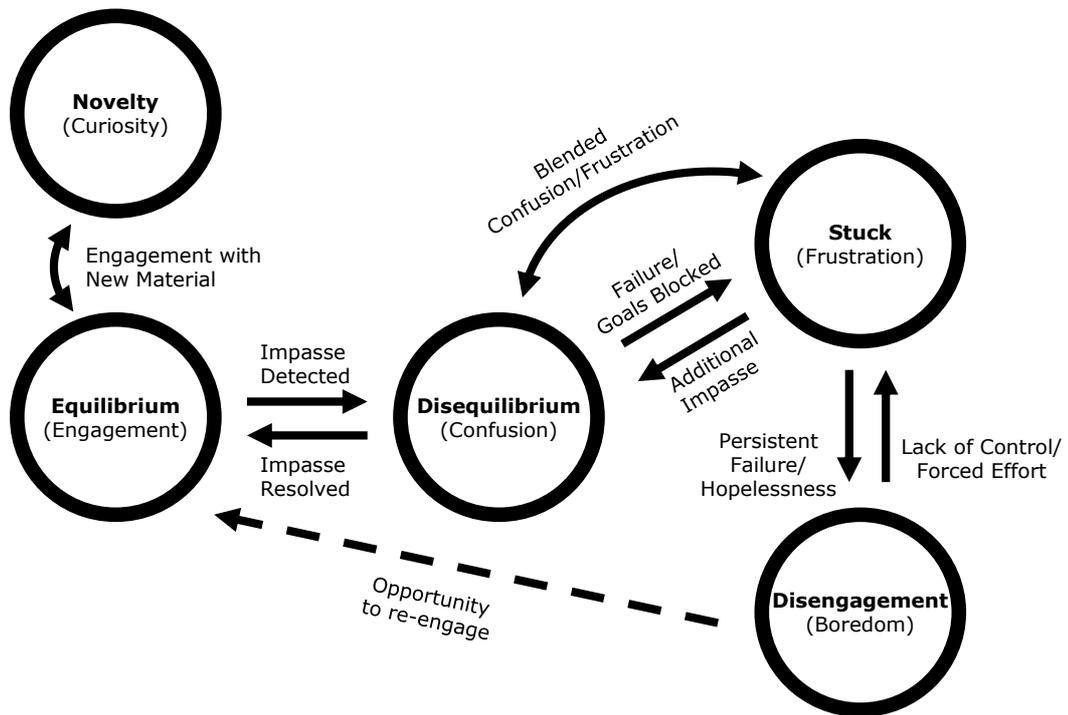


Fig. 7. Updated model of affect transitions and co-occurrence based on findings. Dashed line represents new transition between affective states. Arcs represent co-occurring affective states.

Our fourth research question focused on contextualizing the affective transition by incorporating interaction events into the analysis. We expected to find positive affective states such as engagement and curiosity following successful interactions such as TestRunSuccess, and vice versa for negative affective states. We found that all key affective states were related to knowledge assimilation (reading) and construction (coding) activities but only the confusion and frustration accompanied failure (Submit Error) and subsequent help seeking behaviors (ShowHint). In general, this analysis led to a more nuanced understanding into antecedent-consequent relationships between affective states, systems actions, and student actions.

Our fifth research question concerned correlations between affect and learning. We expected alignment with previous computer programming education research, where negative affective states including boredom, confusion, and frustration negatively correlated with performance while engagement positively correlated with performance (Rodrigo, Baker, Jadud, et al., 2009; Rodrigo & Baker, 2009). Our results confirmed that boredom and frustration during the scaffolded learning phase negatively correlated with performance on the fadeout phase. As expected engagement and boredom-engagement transitions during scaffolding was also positively correlated with learning (performance on the fadeout phase). Importantly, confusion and reciprocal confusion-frustration transitions during scaffolding positively correlated with fadeout performance. This is consistent with impasse-driven theories of learning, which suggest that confusion provides an opportunity to learn and challenging impasse resolution activities that accompany confusion (and can even lead to frustration) can be beneficial to learning (D’Mello & Graesser, 2014a, 2014b; D’Mello, Lehman, Pekrun, & Graesser, 2014; VanLehn, Siler, Murray, Yamauchi, & Baggett, 2003).

Implications for Intelligent Learning Environments

Our findings can inform the development of more effective education technologies for computer programming. One way to increase the effectiveness of these technologies is to design them to be responsive to student affect (D’Mello, Blanchard, et al., 2014; D’Mello & Graesser, 2015). Affect-aware learning technologies require affect detection in order to determine what states to track and when to intervene. Our findings on affective incidence suggest that these technologies should focus on engagement, confusion, frustration, boredom, and curiosity. Three of these four states (confusion, frustration, and boredom) and experiences as being negatively valenced, so it might be particularly important to focus on those states.

Our findings on co-occurring affective states can be used to inform affect detectors about which states are likely to be confused together (i.e., confusion-frustration; curiosity-engagement). In particular, might the somewhat lower accuracies (see Calvo & D’Mello, [2010]; S. D’Mello & Kory [2012] for reviews) of state-of-the-art affect detection systems be attributed to co-occurring affect? Should these affect detectors focus on detecting *affective blends*? If so, what is the appropriate response? Should an affect-aware learning technology respond to confusion, frustration, or both, if these states co-occur?

The results on transitions between affective states and interaction events are important because they provide insight into the events that precede and follow affective states. Affect-aware learning technologies for computer programming may be able to leverage this information in many ways. For example, interaction events can be used to develop log-file based affect detectors that can complement face-based affect detection (Bosch, Chen, Baker, Shute, & D’Mello, in press). They can be used to design affect-aware interventions, such as recommending a hint when excessive frustration is detected. Additionally, knowledge of affective states and events may lead to better curriculum development for computer programming education. For example, events such as submission errors (which correlate with frustration) could be monitored for different programming exercises in order to determine which would be likely to lead to excessive frustration.

Implications of the findings for other domains are less clear. On one hand, the results frequently aligned with findings from other domains involving complex problem solving (learning computer literacy with an ITS). The results were also instrumental in advancing theory on affect and learning and theories are intended to be generalizable. On the other hand, the results might not generalize to other contexts, such as reading a computer programming text because one might not observe the same levels of confusion and frustration in a text comprehension task. In essence, further research is needed to test generalizability more explicitly.

Limitations and Future Work

There are some limitations with the present study that need to be addressed in future work. One set of limitations stems from how the data were collected. First, self-reports are biased by the honesty of the students, so future studies should consider alternate methods in addition to or in lieu of self-reports. Possible methods include online observations (Ocumpaugh, Baker, & Rodrigo, 2015), video coding by trained judges (Graesser et al., 2006), or sensor-based affect measurement (Calvo & D’Mello, 2010). Second, the sample size was small, which limited the statistical power required to detect smaller effects. Third, the students were sampled from a single university, so the results might not generalize to the larger population of novice computer programmers. Fourth, data collected in a lab study may not generalize to more realistic educational scenarios. Future work might benefit from data collection

in an ecologically valid learning experience (e.g., when students complete their own programming homework).

An additional limitation of this study is the potential for the results being specific to the learning environment. In particular, the incidence of curiosity and engagement varies across scaffolded and fadeout phases. This finding can be attributed to key differences in the activities and affordances across these phases. It also suggests that differences are to be expected when different learning environments are considered because these will likely involve different activity types and interface affordances (e.g., access to hints, availability of feedback). Therefore, future work should include plausible variations in learning environments and instruction formats to further explore the potential relationships between those factors and student affect.

Future work should also consider more nuanced relations between affect, interaction events, and learning than the partial correlations reported in this paper. For example, moderation analysis might be used to uncover possible moderators (e.g., individual differences such as gender or SAT score) of the affect-learning relationship. Similarly, separately examining high and low performing students might yield different relationships between affect and learning. Students might also be grouped based on patterns in their affect over time and then analyzed separately. For example, as seen in Table 1, there was a tendency for students to report more engagement in the fadeout phase. However, this effect might not be observed in all students. Finally, a fine-grained exercise-level analysis might also yield insights about what materials or concepts are more difficult to grasp than others.

Concluding Remarks

A working knowledge of computer programming might soon be as critical of a skill as reading or writing in the digital age. But learning computer programming is an intellectually challenging and difficult endeavor – factors that yield to a complex interplay between affect and cognition. The present research focused on developing a better understanding of the affective experience of novices who are attempting to learn computer programming for the first time. The next step is to leverage the insights gleaned in this research to develop more effective next-generation learning technologies for computer programming education.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation (NSF) (ITR 0325428, HCC 0834847, DRL 1235958) and the Bill & Melinda Gates Foundation. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

REFERENCES

- Alsbaugh, C. A. (1972). Identification of Some Components of Computer Programming Aptitude. *Journal for Research in Mathematics Education*, 3(2), 89–98.
- Baker, R., Rodrigo, M. M. T., & Xolocotzin, U. E. (2007). The dynamics of affective transitions in simulation problem-solving environments. In A. C. R. Paiva, R. Prada, & R. W. Picard (Eds.), *Affective Computing and Intelligent Interaction* (pp. 666–677). Berlin Heidelberg: Springer.
- Blignaut, P., & Naude, A. (2008). The influence of temperament style on a student's choice of and performance in a computer programming course. *Computers in Human Behavior*, 24(3), 1010–1020.

- Bosch, N., Chen, H., Baker, R., Shute, V., & D'Mello, S. (in press). Accuracy vs. availability heuristic in multimodal affect detection in the wild. In *Proceedings of the 17th International Conference on Multimodal Interaction*. New York, NY: ACM.
- Bosch, N., & D'Mello, S. (2013). Sequential patterns of affective states of novice programmers. In E. Walker & C. K. Looi (Eds.), *Proceedings of the First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)* (pp. 1–10).
- Bosch, N., D'Mello, S., & Mills, C. (2013). What emotions do novices experience during their first computer programming learning session? In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED 2013)* (pp. 11–20). Berlin Heidelberg: Springer-Verlag.
- Burleson, W., & Picard, R. W. (2004). Affective agents: Sustaining motivation to learn through failure and a state of stuck. In *Social and Emotional Intelligence in Learning Environments Workshop In Conjunction with the 7th International Conference on Intelligent Tutoring Systems, Maceio-Alagoas, Brasil*.
- Calvo, R. A., & D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1), 18–37.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences (2nd ed.)*. Hillsdale, NJ: Erlbaum.
- Cole, J. S., & Gonyea, R. M. (2010). Accuracy of self-reported SAT and ACT test scores: Implications for research. *Research in Higher Education*, 51(4), 305–319.
- Craig, S., D'Mello, S., Witherspoon, A., & Graesser, A. (2008). Emote aloud during learning with AutoTutor: Applying the Facial Action Coding System to cognitive–affective states during learning. *Cognition & Emotion*, 22(5), 777–788.
- Craig, S., Graesser, A., Sullins, J., & Gholson, B. (2004). Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media*, 29(3), 241–250.
- Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. New York: Harper and Row.
- D'Mello, S. (2013). A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. *Journal of Educational Psychology*, 105(4), 1082–1099.
- D'Mello, S., Blanchard, N., Baker, R., Ocumpaugh, J., & Brawner, K. (2014). I feel your pain: A selective review of affect-sensitive instructional strategies. In R. Sottolare, A. Graesser, X. Hu, & B. Goldberg (Eds.), *Design Recommendations for Intelligent Tutoring Systems - Volume 2: Instructional Management* (pp. 35–48).
- D'Mello, S., & Graesser, A. (2010). Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-Adapted Interaction*, 20(2), 147–187.
- D'Mello, S., & Graesser, A. (2012). Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2), 145–157.
- D'Mello, S., & Graesser, A. (2014a). Confusion. In R. Pekrun & L. Linnenbrink-Garcia (Eds.), *International handbook of emotions in education* (pp. 289–310). New York, NY: Routledge.
- D'Mello, S., & Graesser, A. (2014b). Inducing and tracking confusion and cognitive disequilibrium with breakdown scenarios. *Acta Psychologica*, (151), 106–116.

- D'Mello, S., & Graesser, A. C. (2015). Feeling, thinking, and computing with affect-aware learning technologies. In R. A. Calvo, S. D'Mello, J. Gratch, & A. Kappas (Eds.), *Handbook of Affective Computing* (pp. 419–434). New York, NY: Oxford University Press.
- D'Mello, S., & Kory, J. (2012). Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies. In *Proceedings of the 14th ACM international conference on Multimodal interaction* (pp. 31–38). New York, NY, USA: ACM.
- D'Mello, S., Lehman, B., Pekrun, R., & Graesser, A. (2014). Confusion can be beneficial for learning. *Learning and Instruction, 29*(1), 153–170.
- D'Mello, S., Person, N. K., & Lehman, B. (2009). Antecedent-consequent relationships and cyclical patterns between affective states and problem solving outcomes. In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. Graesser (Eds.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (pp. 57–64). Amsterdam: IOS Press.
- D'Mello, S., Taylor, R. S., & Graesser, A. (2007). Monitoring affective trajectories during complex learning. In *Proceedings of the 29th annual meeting of the cognitive science society* (pp. 203–208). Austin, TX: Cognitive Science Society.
- Graesser, A., McDaniel, B., Chipman, P., Witherspoon, A., D'Mello, S., & Gholson, B. (2006). Detection of emotions during learning with AutoTutor. In R. Sun & N. Miyake (Eds.), *Proceedings of the 28th Annual Meetings of the Cognitive Science Society* (pp. 285–290). Austin, TX: Cognitive Science Society.
- Grafsgaard, J. F., Fulton, R. M., Boyer, K. E., Wiebe, E. N., & Lester, J. C. (2012). Multimodal analysis of the implicit affective channel in computer-mediated textual communication. In *Proceedings of the 14th ACM international conference on Multimodal interaction* (pp. 145–152). New York, NY, USA: ACM.
- Harley, J. M., Bouchet, F., & Azevedo, R. (2012). Measuring learners' co-occurring emotional responses during their interaction with a pedagogical agent in MetaTutor. In S. A. Cerri, W. J. Clancey, G. Papadourakis, & K. Panourgia (Eds.), *Intelligent Tutoring Systems* (pp. 40–45). Berlin Heidelberg: Springer.
- Haungs, M., Clark, C., Clements, J., & Janzen, D. (2012). Improving first-year success and retention through interest-based CS0 courses. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 589–594). New York, NY, USA: ACM.
- Hosseini, R., Vihavainen, A., & Brusilovsky, P. (2014). Exploring problem solving paths in a Java programming course. In M. Coles & G. Ollis (Eds.), *Psychology of Programming Interest Group Annual Conference 2014* (pp. 65–76).
- Inventado, P. S., Legaspi, R., Cabredo, R., & Numao, M. (2012). Student learning behavior in an unsupervised learning environment. In *Proceedings of the 20th International Conference on Computers in Education* (pp. 730–737). Singapore: National Institute of Education.
- Jadud, M. C. (2005). A first look at novice compilation behavior using BlueJ. *Computer Science Education, 15*(1), 25–40.
- Khan, I. A., Hierons, R. M., & Brinkman, W. P. (2007). Mood independent programming. In *Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore!* (pp. 28–31). New York, NY, USA: ACM.
- Larson, R. W., & Richards, M. H. (1991). Boredom in the middle school years: Blaming schools versus blaming students. *American Journal of Education, 99*(4), 418–443.
- Law, K. M. Y., Lee, V. C. S., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education, 55*(1), 218–228.

- Lee, D. M. C., Rodrigo, M. M. T., Baker, R., Sugay, J. O., & Coronel, A. (2011). Exploring the relationship between novice programmer confusion and achievement. In S. D'Mello, A. Graesser, B. Schuller, & J. C. Martin (Eds.), *Affective Computing and Intelligent Interaction* (pp. 175–184). Berlin Heidelberg: Springer.
- McQuiggan, S. W., Robison, J. L., & Lester, J. C. (2008). Affective transitions in narrative-centered learning environments. In B. P. Woolf, E. Aïmeur, R. Nkambou, & S. Lajoie (Eds.), *Intelligent Tutoring Systems* (pp. 490–499). Berlin Heidelberg: Springer.
- Min, W., Mott, B., & Lester, J. (2014). Adaptive scaffolding in an intelligent game-based learning environment for computer science. In *Proceedings of the Workshop on AI-supported Education for Computer Science (AIEDCS) at the 12th International Conference on Intelligent Tutoring Systems* (pp. 41–50).
- Ocuppaugh, J., Baker, R., & Rodrigo, M. M. T. (2015). Baker Rodrigo Ocuppaugh Monitoring Protocol (BROMP) 2.0 Technical and Training Manual. In *Technical Report*. New York, NY: Teachers College, Columbia University. Manila, Philippines: Ateneo Laboratory for the Learning Sciences.
- Pekrun, R., & Linnenbrink-Garcia, L. (Eds.). (2014). *International handbook of emotions in education*. New York, NY: Routledge.
- Pekrun, R., & Stephens, E. J. (2012). Academic emotions. In K. R. Harris, S. Graham, T. Urdan, S. Graham, J. M. Royer, & M. Zeidner (Eds.), *APA educational psychology handbook, Vol 2: Individual differences and cultural and contextual factors* (pp. 3–31). Washington, DC, USA: American Psychological Association.
- Porayska-Pomsta, K., Mavrikis, M., D'Mello, S., Conati, C., & Baker, R. (2013). Knowledge elicitation methods for affect modelling in education. *International Journal of Artificial Intelligence in Education*, 22(3), 107–140.
- Rodrigo, M. M. T., & Baker, R. (2009). Coarse-grained detection of student frustration in an introductory programming course. In *Proceedings of the Fifth International Workshop on Computing Education Research* (pp. 75–80). New York, NY, USA: ACM.
- Rodrigo, M. M. T., Baker, R., Jadud, M. C., Amarra, A. C. M., Dy, T., Espejo-Lahoz, M. B. V., ... Tabanao, E. S. (2009). Affective and behavioral predictors of novice programmer achievement. *SIGCSE Bulletin*, 41(3), 156–160.
- Rodrigo, M. M. T., Baker, R., Sugay, J. O., & Tabanao, E. S. (2009). Monitoring novice programmer affect and behaviors to identify learning bottlenecks. In *Philippine Computing Society Congress 2009 Research-in-Progress Section*. Dumaguete City.
- Rosenberg, E. L., & Ekman, P. (1994). Coherence between expressive and experiential systems in emotion. *Cognition & Emotion*, 8(3), 201–229.
- Shute, V. J., & Kyllonen, P. C. (1990). *Modeling Individual Differences in Programming Skill Acquisition* (Technical Paper No. AFHRL-TP-90-76) (p. 34). Brooks AFB, TX: Air Force Human Resources Laboratory.
- Tan, P.-N., Kumar, V., & Srivastava, J. (2002). Selecting the Right Interestingness Measure for Association Patterns. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 32–41). New York, NY, USA: ACM.
- VanLehn, K., Siler, S., Murray, C., Yamauchi, T., & Baggett, W. B. (2003). Why do only some events cause learning during human tutoring? *Cognition and Instruction*, 21(3), 209–249.

Weber, G., & Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12, 351–384.