Examining the Intermediate Programmers Understanding of the Learning Process

David Simkins, Adrienne Decker
School of Interactive Games and Media and RIT Center for Media, Arts, Games, Interaction and Creativity (MAGIC)
Rochester Institute of Technology
Rochester, NY - USA
dwsigm@rit.edu, adrienne.decker@rit.edu

Abstract— **Within computer science education, we have spent considerable effort on the introduction to the discipline (particularly to programming) and the teaching of novice programmers. However, we do not often think about the teaching and learning for the intermediate students. Having data about student's perceptions coming into a second year data structures course, it became of interest to systematically analyze the data to see what if any interesting patterns or results we could see in this information. In this paper, we present the results of the analysis of the open-ended questions at the end of the survey. The students were asked to identify a concept that they found particularly difficult and to tell us why. They were also asked to describe how they achieved success in learning a concept that they initially found difficult. We developed a system of thematic codes using a grounded theory approach on a sampling of the data, which we then applied to the entire data set. We uncovered patterns and trends in the student responses. During our analysis, we have uncovered that students are expressing blame for the lack of learning on themselves and instructors, pointing to perceived failures in the classroom environment and course structure, and are showing evidence of the understanding of the learning process. We conclude with some overall recommendations for how this findings could be operationalized to better understand the process for this intermediate learner.**

Keywords—role play; soft skills; software engineering; game development lifecycle; business and legal concerns

I. INTRODUCTION

Intermediate students in computer science are caught between the commonly studied novices and the experts we expect them to become. We have spent considerable time in with novice programmers, examining the difficulties and trajectories of new programmers as they begin their study. However, we have spent significantly less time studying these intermediates to determine how they are proceeding, where they are having the most difficulty, and how they manage to overcome it. Through survey data given to intermediate students as they begin intermediate programming, this study examines the student's own understanding of their challenges on their way to their current level of knowledge. Through this admittedly retrospective self-report from intermediate students, we uncover not only the areas they perceive as difficult, but also the ways in which they believe knowledge is acquired, and who they believe has access to it.

II. BACKGROUND

The work on novice programmers entry into programming has been well established [1], but there is less work focused on intermediate programming students. Work focused on intermediate student programmers include qualitative analysis focused on teachers and instructor. A phenomenological study of instructors approaches to problems in teaching this population through themes of who and what, population and topic [2] A related student-focused study takes an interview-based approach to evaluate intermediate student experiences [3]. These articles provide deep insight into the experiences of instructors and students. This current work-in-progress is a companion to these works, providing a mixed-methods approach with a much larger sample that will ultimately be able to provide wider investigation of the experiences described. This is intentionally a ground-up, theory-building project that can independently speak to these data, and to a broad base of future approaches.

The scope of this work focuses on student responses. In an attempt to develop toward a large population survey that can be analyzed using our developing mixed-methods approach, a decision was made to focus on student experiences of difficulty. The work-in-process data sample is being expanded across multiple universities and departments, looking at intermediate programming students. Due to the nature of the data sources, researchers will not necessarily have access to potentially useful corroborating data sources, including individual student performance. Perhaps in future work we, or others, will find ways to establish additional sources to further triangulate these data, but for the moment this belongs to further inquiry.

III. METHODS

The study was drawn from survey data administered during the first days of the intermediate programming sequence (n=140). The initial section included a series of topics with likert responses to indicate level of difficulty. Analysis of these data lie outside the scope of this paper. This paper focuses on two of three free-response questions that followed the likert responses. This was followed by three free response questions. The first two are a pair, and the second is the target of the subject of this paper's analysis. The questions were:

1. Give an example of something you studied in your previous computer science courses (does not have to be from the list above) that you feel you still don't understand very well.
2. Why do you feel the topic you indicated was difficult for you to understand?

The responses to question 1 were collected into categories of similar answers. These were in almost all identifiable topics drawn from earlier computer sciences courses at the same institution. Where the topic could not be identified, the answer was omitted.

The responses to question 2 were analyzed independently from question 1. A first pass through a randomly selected subset of the data identified nine categories of interest.

1. Where did students place the blame for the difficulty of the topic?
2. Did the students show either a entity or incremental theory of intelligence in their responses?
3. Did the student indicate a behavioral failure in their approach to the problem?
4. What tools for learning were identified as useful or needed but lacking?
5. Did the student indicate a difficulty in the mode of instruction of the previous courses?
6. Did the student show sophisticated metacognitive awareness in their response?
7. Did the student experience a lack of comfort with the difficult topic?
8. Did the student express difficulty with the abstract nature of the problem?
9. Did the student express a difficulty in topic transfer?

The categories were established using a bottom up coding scheme, the entire set was coded [4]. In each case the unit of analysis was the student's entire response. The researchers evaluated the entire data set collaboratively and deliberatively [5]. That is, they read each unit of data together, discussed its applicability to each of the nine categories and determined together which if any of the categories were appropriate. Each category was tested as if they were independent, that is making no assumptions about their dependence or independence at this point, so the response could apply to none, some or all of the variables. In the context of this approach, the researchers reached saturation [6][7], where no additional codes were added despite additional data. Given the discrete and pre-existing corpus of data, and a desire to allow for mixed methods analysis of the data, the entire data set was analyzed even after data saturation was achieved.

While in later implementations of this coding scheme it will be wise for multiple researchers to conduct independent analysis of the codes, with an inter-rater reliability test to determine the coherence of the codes across researchers. This is a complex coding scheme, with more categories than normal, and this pilot data was used to develop a coherent understanding using the entire corpus of currently existing data to create as complete as possible a coding schema within the categories through saturation. The coding set developed itself becomes an important product of research, useful for future evaluation of similar questions.

Each of the categories derived from the analysis of data is relevant to the general questions of computer science education, and additional background information may help to establish their importance within the scope of computer science education.

A. Blame

Codes identify where the student places blame for the difficulty of the subject. Where the student places blame can help us indicate where we should target future interventions, providing more support to either mitigate or eradicate the source of the difficulty.

B. Theory of intelligence

A great deal of work has been conducted recently in the identification of student theories of intelligence. A theory of intelligence is understood as the student's belief in where intelligence originates. Research in this are has identified two primary theories of intelligence, entity theories and incremental theories [8]. In entity theories, intelligence is thought to be inherent in the person, fixed or minimally malleable. Though the student must still learn, what they are able to learn is determined by who they are. In incremental theories of intelligence, the source of intelligence is thought to be through gradual accumulation. Through effective work, intelligence is developed over time, and anyone or almost anyone is capable of developing greater intelligence were they to go through the same work. While this certainly mirrors a nature and nurture debate, both views are agnostic to views of where identity comes from (society, genetics, etc.) or how developmental opportunities arise (context, exposure, individual effort, etc.). These should not therefore be seen as, for example, the genetic versus the developmental view, but instead in keeping with the subjective nature of the question studied, an indication of epistemological disposition.

Research into student theories of intelligence have shown inverse correlations between entity theories of intelligence and long-term success in learning [9].

C. Student behavior

Students engage in a variety of behaviors that can be supportive or detrimental to their success in college. Computer science students are no exception. In this category, codes were assigned when students self-identified problems in their own behavior that led to difficulties understanding the concept.

D. Tools for learning

This category included codes identifying tools for learning that students reported were helpful or, more often, where their absence or insufficient utilization was harmful to the student's ability to learn the concept.

E. *Mode of instruction*

This category included codes identifying modes of instruction that students reported were helpful or, more often, where the absence or insufficient utilization was harmful to the student's ability to learn the concept.

F. *Metacognition*

Some of the students showed a significant understanding of the problem and the difficulties they experienced in solving the problem. This category is for instances where the researchers see in the student's response that the student understood the concept they found difficult and were able to articulate the cognitive difficulty they experienced in approaching the problem. In other words, the students showed in their response a sophisticated analysis of their own cognitive process when approaching the problem.

G. *Comfort*

This category includes instances where students expressed a level of affective discomfort with the concept, and there is a strong suggestion or indication that this affective state contributed to the student's difficulty with the concept. While other categories might include understanding in both cognitive [10] and affective domains [11], there is typically a skew toward cognitive. This category is relevant entirely to the affective domain of learning.

H. *Abstract*

This category includes instances where students expressed that the difficult lay, at least in part, in the abstract nature of the concept. This is a cognitive expression of difficulty, as differentiated from an affective difficulty with abstraction, which would be comfort, though it is possible for a response to include both.

I. *Transfer*

This category includes instances where the student expresses difficulty in transfer of understanding from one associated concept to the concept they identify. This category is not just for indications that the student had difficulty transferring a concept, but for instances where the student expresses that transfer is involved.

IV. Results

For each category, codes were developed using bottom-up approaches derived from grounded theory [12]. This approach is influenced by thematic analyses [13]. The thematic codes developed from this bottom-up analysis were derived until code saturation was reached, the point at which no new codes were emerging from the data. The result of coding is indicated below by category and code, with percentages of responses for which each code applied. In a few cases there is only one applicable code, in which case the percentage by the category indicates that the category present in the response. As with categories, when more than one code is relevant to the student's statement within a category, all relevant codes apply, so percentages will often by greater than 100%.

Blame		
	Student	25.00%
	Instructor	27.14%
	Problem	25.00%
	Course Structure	6.43%
	none	23.57%
Theory of Intelligence		
	Entity	7.14%
	Incremental	12.14%
	none	80.71%
Student Behavior		
	lack of effort	4.29%
	lack of motivation	0.71%
	studied the wrong chapter	0.71%
	skipped class	0.71%
	none	93.57%
Tools for Learning		
	Practicing	10.71%
	Lecture	6.43%
	Examples	5.71%
	Definitions	2.14%
	Studying	1.43%
	Reading	0.71%
	Online Resources	0.71%
	Textbook	0.71%
	none	74.29%
Mode of Instruction		
	Insufficient Explanation	12.86%
	Not enough time during semester	8.57%
	Insufficient scaffolding	4.29%
	Concept not made Relevant	4.29%
	Course sped up at the end	1.43%
	Instructor unclear	0.71%
	When covered in the semester	0.71%
	none	65%
Metacognition		3.57%
Comfort		5.71%
Abstract		14.29%
Transfer		2.14%

V. Conclusion

Analysis of this data continues to be a work in progress. The creation of categories, and codes within categories, will allow future research to target these categories and codes with a much larger sample size to see if these trends persist across populations of mid-level programming students.

The current state of the data has a number of interesting suggestions. Student explanation of difficulty contains a warranted claim that places blame more than 75% of the time. Within that, the students seem equally likely to blame themseles, their instructor, or the problem itself.

Despite the need to warrant in terms of blame, there is little tendency to provide a warrant based either on the student's own innate capacity or the student's lack of work, entity and incremental theories of intelligence respectively. Relatively few of the students identified any fault in their own behavior, even when they blamed themselves (comparing blame of student 25% with total blame on student behavior, 6.43%).

Adequate lectures and clear and numerous examples both seemed to be missed by students when they reported a lack, and the instructor's insufficient explanation seemed to have a substantial effect on student perceptions of success.

The abstract nature of the concepts student struggled with was noted in the data, but not overwhelmingly so. Students also did not often expressly indicate an affective discomfort with the concepts.

Given the fairly straightforward question asked, it is perhaps interesting to see strong evidence of metacognition and transfer at all, though both are in relatively low percentages.

Though there are some interesting suggestions in the raw percentages, there is a deeper dive into the data which would be fruitful. A cross-section of data to determine if there are clusters of responses would be useful, and further study may show that some of the codes are likely less independent than the methodology initially assumes.

There is a great deal of research on entity and incremental theories of intelligence, but additional understanding of correlative features of these theories in students would help to identify and change student practices before they lead to future problems.

Overall, this pilot study has opened the door for future work on mid-level programming students and will help develop tools for researching these populations, with the goal of identifying problem areas and developing means of improving student learning.

References

[1] Pears, A., Seidman, S. Malmi, L, Mannila, L., Adams, E., Bennedson, J., Devlin, M., & Paterson, J. A Survey of Literature on the Teaching of Introductory Programming. In *ITiCSE-WGR '07 Working group reports on ITiCSE on Innovation and technology in computer science education*, New York, 2007, 204-223.

[2] Bergland, A., Eckland, A., Pears, A., East, P., Kinnunen, P., Malmi, L., McCartney, R., Mostrom, J-E., Murphy, L., Ratcliffe, M., Schulte, C., Stamouli, I., & Thomas, L. Learning computer science: perceptions, actions and roles. *European Journal of Engineering Education 34, (4),* August 2009, 327-338.

[3] Pears, A-K., Berglund, A., Exkerdal, A., & Pears, A. Second Year Computer Science and IT Students' Experience of Participation in the Discipline. In the proceedings of Koli Calling 2015, November 19-22, 2015, Koli, Finland.

[4] Strauss, A. & Corbin, J. M. Grounded Theory in Practice. Thousand Oaks, CA: Strauss.

[5] Paulus, T., Woodside, M. & Zeigler, M. Extending the Conversation: Qualitative Research as a Dialogic Process. The Qualitative Report 13(2), June 2008, 226-243.

[6] Glaser, B. & Strauss, A. The Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago: Aldine, 1967.

[7] Miles, M. B. & Huberman, A.M..Qualitative Data Analysis (2nd edition). Thousand Oaks, CA: Sage Publications, 1994.

[8] Dweck, C. S. Self-Theories: Their Role in Motivation, Personality, and Development. Philadelphia, PA: Model Press. 2000.

[9] Cutts, Q., Cutts, E., Draper, S., O'Donell, P. & Saffrey, P. Manipulating Mindset to Positively Influence Introductory Programming Performance. In Proceedings of the 41st ACM Technical Symposium on Computer Science Education. New York, NY, 431-435. 2010.

[10] Anderson, L. W. & Krathwohl, D. P. A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy for Educational Objectives. New York: Pearson.

[11] Krathwohl, D.R., Bloom,B.S. & Masia, B. B. .Taxonomy of educational objectives, Book II. Affective domain. New York, NY. David McKay, 1964.

[12] Glaser, B. & Strauss, A. The Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago: Aldine, 1967.

[13] Cresswell, J. W., *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches.* Los Angeles, CA: SAGE, 2009.