Publications

2017

A Human Factors Perspective on Learning Programming Languages using a Second Language Acquisition Approach

Rebecca Rohmeyer
Embry-Riddle Aeronautical University, rohmeyer@my.erau.edu

Paula Sanjuan Espejo
Embry-Riddle Aeronautical University, sanjuanp@my.erau.edu

Lulu Sun
Embry-Riddle Aeronautical University, sunl@erau.edu

Christina Frederick
Embry-Riddle Aeronautical University

Follow this and additional works at: https://commons.erau.edu/publication

Part of the Education Commons, and the Engineering Education Commons

A Human Factors Perspective on Learning Programming Languages using a Second Language Acquisition Approach

Rebecca Rohmeyer, Paula Sanjuan Espejo, Dr. Lulu Sun, and Dr. Christina Frederick

Embry-Riddle Aeronautical University

Abstract

In this day and age, knowing a programming language is an essential skill to have for those pursuing a career in any of the STEM fields. In most colleges and universities around the world, engineering and computer science students are required to take an introductory course in a programming language. However, many students find these courses intimidating and too challenging. This paper explores a psychological perspective on learning programming languages using Second Language Acquisition (SLA) theories. The paper also describes the basic function of how learning works and how SLA aids in the learning process. This paper also briefly discusses the Second Language Acquisition in a Blended Learning (SLA-aBLe) project, and how the use of SLA techniques facilitated students learning MATLAB. Demographic survey data and overall grade data from spring 2016 show that students in the SLA-aBLe courses received overall higher grades and felt less overwhelmed and intimidated.

Keywords

programming, hybrid, psychology, second language acquisition, MATLAB

Introduction

Programming is an essential skill to have for those who are studying to work in any of the STEM fields. In most colleges and universities around the world, it is required for engineering and computer science students to take at least one introductory course in a programming language, whether it is MATLAB, Java, or C/C++. However, students often find these classes intimidating and too challenging[1]. An explanation for this sense of insecurity is the way the students attempt to organize their knowledge. Research[2] was conducted to explain how students often attempt to store information of a programming language in terms of syntax. This research discusses that novice programmers (i.e. students) often organize what they learn in class in a syntax-oriented way, meaning they often do not understand the concept of the syntax and how it works in their programs, due to how the professor/teacher presents the material. Other research[3] suggests that this lack of understanding students experience is due to not having a clear mental model about how their program works in the real world. A mental model is essentially how someone explains

their thought process and how it operates in the real world through our experiences in life. On the other hand, experts tend to organize their knowledge in terms of concepts instead of specific syntax. This paper discusses the problems with how current programming classes are taught, the psychological theories of learning and learning languages, and techniques that could improve the way programming language classes are taught. This paper will also briefly describe our project, Second Language Acquisition in a Blended Learning Environment (SLA-aBLe) and our results.

Problems with current design of programming classes

Currently, there is much more emphasis on the outcome of the student's code rather than the process of which the student took to write their code[2]. Professors are often only interested if the code successfully runs rather than the design of the program and the steps the students took to get the outcome. This may cause students to be lazy with their algorithms, have messy code, and cause them to not fully understand how their code works. Another problem is that professors often assume that students can learn programming on their own through discovery of syntax and how the syntax works[2]. In other words, students are somewhat expected to figure it out on their own through trial and error. While this might work eventually, it is not the most efficient way to learn a new skill, especially for students who only have limited time per semester. Another frustrating reality for students is that they are often taught that programs are designed using an uncomplicated top-down design process, which only misleads students[2]. For these reasons, there needs to be a change in the way these programming language classes are taught.

Theories of learning and learning languages

Before moving forward to discussing new and improved approaches to teach programming languages, it is important to understand what learning is. It is also important to understand how people learn first and second languages, since programming languages can be compared to languages. Learning can be simply defined as "a persisting change in human performance or performance potential…which must come about as a result of the learner's experience and interaction with the world."[4]. For learning languages, there are two important terms to understand: language acquisition and language learning. Language *acquisition* is generally described as the innate, species-specific linguistic knowledge in one's mastery of a language, whether it be native or foreign[5]. Acquisition generally proceeds rapidly, occurs unconsciously and effortless, no negative evidence, etc[5]. Acquisition is similar to how children learn their first and second languages[6]. Children generally acquire knowledge not through formal teaching of grammar and vocabulary, but from being surrounded by others talking to them, such as baby talk[7].

On the other hand, language *learning* involves two psychological processes: explicit learning and implicit learning. Explicit learning happens when the learner is aware they are learning and modifying their knowledge. Implicit learning happens when there is a modification to the learners knowledge, but they are unaware of it occurring. Overall, learning occurs relatively slowly compared to acquisition, requiring more effort from the learner.

For those learning a new language, second language acquisition (SLA) techniques are used. Behaviorist theories in SLA techniques are one of the most popular types used currently in a

foreign language class in high school. For example, students are presented with audiolingual dialogues on a tape that students must memorize. These dialogues are then followed up with sentence structures, and vocabulary. Students are required to memorize the sentence structures and vocabulary through imitation and repetition[8]. Another theory of second language acquisition is the innatist point of view, which means that instead of having language dialogues from teachers, language is acquired through natural language from friends, family, or books. Also, output is not a concern since it is going to occur naturally, and errors are not corrected since the students will correct themselves over time. This could help students feel less stressed and intimidated[8]. However, this SLA technique may be too time consuming for a classroom setting. A third SLA theory is the interactionist point of view. This technique is similar to that of the innatist point of view. Both techniques rely on natural language with friends, family, and books. The difference with the interactionist point of view is that native speakers and target speakers are expected to interact to focus on communication. Both theories put little to no pressure on the learner, which can help with students' stress. It should be noted that language acquisition has yet to be fully understood, hence why there are an abundant of theories on this topic[9-13]. For that reason, only a select few theories were discussed in this paper.

New approaches to teaching programming languages

Now that we understand techniques of acquiring a second language, it is appropriate to discuss how those techniques may come useful for future programming classes. Since learning a program is analogous to learning a second language, second language acquisition techniques are going to be discussed. Research suggests that blended environments (combining different types of teaching styles and techniques) aid the learning process for students[15-19]. Teachers have been utilizing blended learning environments by supplementing in-class traditional material with e-learning activities[8] for decades. Using computers in the classroom is nothing new[8]. However, with the emergence of Web 2.0, there are many new opportunities for teachers to take advantage of to include in their blended learning environments[8]. There is also evidence that the use of cognitive frameworks enhances the learning process and can improve engagement for students[14].

Second Language Acquisition in a Blended Learning Environment (SLA-aBLe) at Embry-Riddle Aeronautical University

The project titled Second Language Acquisition in a Blended Learning Environment (SLA-aBLe) is concerned with redesigning the way that the Introduction to MATLAB (EGR115) classes are taught. The traditional MATLAB classes are hybrid classes, meaning some of the lectures are videos that the students watch on their own time and take online quizzes. This project wants to enhance these videos by utilizing cognitive frameworks to improve engagement and the learning experience. Also, the videos were made to be more interactive by including quizzes along the way to make sure the students are getting some practice before moving on the next major topic. There is some research that suggests having short, multiple-choice quizzes can help students score higher on exams[16]. We were curious to see the comparisons of perceived workload, learning outcomes, and overall engagement in the non SLA-aBLe sections and the SLA-aBLe sections. The SLA-aBLe sections tested the use of cognitive frameworks can improve the engagement and how effectively the students learn the course material[20]. For more

information of this study, please refer to the paper "A Second Language Acquisition Approach to Learning Programming Languages"[20].

Results of SLA-aBLe Pilot Study 1

For the first pilot study, which was conducted in Fall 2015 and Spring 2016, there were 3 instructors, each teaching at least 1 SLA and 1 non-SLA section. Four measures to compare non-SLA-aBLe sections and SLA-aBLe sections: NASA TLX, end of course evaluations, grades, and the use of videos. The students also took a demographic survey. For the NASA TLX, which measures cognitive, physical, temporal, performance demands, perceived effort, and frustration, SLA-aBLe had significantly higher scores for effort/importance of class, competence and usefulness, and enjoyment and competence. Also, SLA-aBLe students felt significantly less pressure/tension than the non-SLA-aBLe. This suggests that those students in the SLA-aBLe sections felt less pressure, put more effort into the class, and enjoyed the class more overall than the non-SLA-aBLe sections. While not statistically significant, SLA-aBLe students had on average a higher overall grade in the class ($M = 78.96$, $SD = 13.61$) than the non-SLA-aBLe students ($M = 76.36$, $SD = 18.54$). A demographics survey was also conducted, which was interested in languages that the students know and if they have any experience with programming languages. There were a total of 136 students who took part in this survey. The results show that 82.84% of them chose English as their native/first language. The students were also asked if they speak any other languages. A little more than half said no, at 53.38%, while 46.62% of the students indicated that they do speak other languages. Table 1 below shows the descriptive statistics of what other languages students know and their skill level. Next, the students were asked if they took any programming language courses in the past. Of the 136 students, 68.15% said no, while 31.85% said they have. Table 2 below shows the descriptive statistics of what programming languages they know and their skill level.

Table 1

Second Language	Not At All Fluent (%)	Not Very Fluent (%)	Moderately Fluent (%)	Somewhat Fluent (%)	Very Fluent (%)
English	0	1.92	1.92	9.62	86.54
Chinese	71.43	7.14	0	0	21.43
German	64.71	23.53	11.76	0	0
Spanish	9.30	34.88	34.88	6.98	13.95
Vietnamese	100	0	0	0	0
French	32.14	35.71	21.43	3.57	7.14
Arabic	84.62	0	0	0	15.38

Korean	85.71	14.29	0	0	0
Portuguese	84.62	15.38	0	0	0
Other	38.46	15.38	7.69	3.85	34.62

Table 2

Program Language	Low Skill Level/Novice (%)	Moderately Low Skill (%)	Moderate Skill Level/Trained Novice (%)	Moderately High Skill (%)	High Skill Level/Expert (%)
MATLAB	44	28	20	8	0
Fontran	93.3	0	6.67	0	0
Java	58.62	17.24	10.34	10.34	3.45
C/C++	72.73	9.09	18.18	0	0
Visual Basic	77.78	5.56	16.67	0	0
Python	59.26	18.52	14.81	7.41	0
Other	73.68	5.26	15.79	5.26	0

Results of SLA-aBLe Pilot Study 2

In the Fall 2016 semester, another pilot study was conducted utilizing the same measurements to compare the non-SLA-aBLe sections and the SLA-aBLe sections. However, this semester, there were only 2 instructors, each having at least 1 SLA section and 1 non-SLA section. As of January 2017, only the grades and the usage of SLA-aBLe videos have been analyzed. We are still in the process of analyzing the survey data. Roughly, 51.25% of the students in the SLA-aBLe sections watched the videos. Similarly to the previous pilot study, there was no significant difference between the final grades; however, the students in the SLA-aBLe courses earned a higher score (M=70.87, SD=20.25) than the students in the non-SLA-aBLe courses (M=67.72, SD=19.99).

Conclusion

Overall, the results of both pilot studies suggest that the use of cognitive frameworks improved engagement and facilitated the learning process for the engineering students. Also, while the

grade data was not significant, the results show that the SLA-aBLe students received higher grades. These results show promise for the use of SLA-aBLe.

References

1   Solomon, J., Programming as a Second Language. Learning & Leading with Technology, 2004.
2   Linn, M.C., & Clancy, M.J., The case for case studies of programming problems, Communications of the ACM, 1992.
3   Siemens, G., Connectivism: A Learning Theory for the Digital Age, 2005
4   Driscoll, M., Psychology of Learning for Instruction, Allyn & Bacon, Needham Heights, MA, 2000.
5   Anderson, D.., Adult Language Learning, 2007.
6   Krashen, S. Second Language Acquisition and Second Language Learning, Pergamon Press Inc.,
7   Garrison, D., & Kanuka, H. Blended Learning: Uncovering its Transformative Potential in Higher Education, 2004.
8   Klein, W., Second Language Acquisition, Cambridge University Press, Cambridge, UK, 1988
9   Bower, G., & Hilgard, E., Theories of Learning, Prentice Hall, Englewood Cliffs, New Jersey, 1981.
10   Xu, Y., & Lee, K.K., Human Behavior Learning and Transfer, Taylor & Francis Group, Boca Raton, FL, 2006.
11   Schwartz, B. & Robbins, J., Psychology of Learning and Behavior, W.W. Norton & Company, New York, NY, 1995.
12   Thompson, R.F., & Madigan, S.A., Memory: The Key to Consciousness, Joseph Henry Press, Washington, DC, 2005.
13   Ambrose, S.A., Bridges, M.W., DiPietro, M., Lovett, M.C., & Norman, M.K., How Learning Works: 7 Research-Based Principles for Smart Teaching, Jossey-Bass, San Francisco, CA, 2010
14   Herrera, H., & White, M., Cognitive Linguistics and the Language Learning Process: a Case from Economics, Estudios Ingleses de la Universidad Complutense, 2000.
15   Milne, I & Rowe, G., Difficulties in Learning and Teaching Programming--Views of Students and Tutors, Education and Information Technologies, Kluwer Academic Publishers, Netherlands, 2002.
16   McDermott, K.B., Agarwal, P.K., D'Antonio, L., Roediger, H.L, & McDaniel, M.A., Both Multiple-Choice and Short-Answer Quizzes Enhance Later Exam Performance in Middle and High School Classes, Journal of Experimental Psychology: Applied, 2014.
17   Marsh, D., Blended Learning Creating Learning Opportunities for Language Learners, Cambridge University Press, 2012.
18   Singh, H., Building Effective Blending Learning Programs, Issue of Educational Technology, 2003.
19   Birner, B., Language Acquisition, Linguistic Society of America, Washington, DC, 1994.
20   Cunningham, R., Sanjuan Espejo, P., Frederick, C., Sun, L., & Ding, L. A Second Language Acquisition Approach to Learning Programming Languages, 2016 ASEE Southeast Section Conference, 2016

Rebecca Rohmeyer

Rebecca Rohmeyer is currently a master's student studying Human Factors and Systems at Embry-Riddle Aeronautical University (ERAU), where she received her B.S. degree in Human Factors in May 2016. Rebecca is interested in psychology, physiology, and the field of user experience. She is a research assistant for the project SLA-aBLe, and in the Aerospace Physiology lab at ERAU. As an undergrad, Rebecca earned three awards for her outstanding

poster at the Human Factors and Applied Psychology (HFAP) Conference at ERAU, outstanding undergraduate researcher, and outstanding undergraduate academic achievement. Rebecca's goal is to work in the user experience field.

Paula Sanjuan Espejo

Paula is an undergraduate junior student in the Aerospace Engineering Program with a concentration in Propulsion at Embry-Riddle Aeronautical University who was born in Spain. She is an Orientation Team Ambassador for the university, a member of the Honors Program, a Women's Ambassador and a research assistant for the SLA-aBLE project in the Engineering Fundamentals Department. She is also Head of Mentors for the International Student Programming Council and is part of the Honor Student Association and Honors Program.

Lulu Sun, Ph.D

Dr. Sun is an Associate Professor of Department of Freshman Engineering at Embry-Riddle Aeronautical University, where she has taught since 2006. She received her B.S. degree in Civil Engineering from Harbin Engineering University (China) in 1999, and her Ph.D. degree in Mechanical Engineering from University of California, Riverside in 2006. Dr. Sun's Ph.D research, which was supported by UCR dean fellowship, USDA, UCR-LANL foundation, focused on pool fire experiments and numerical modeling of fire spread. The project of fire behavior in live fuels was ranked as best among 14 projects funded by the National Fire Plan of US in 2003.

Christina Frederick, Ph.D

Dr. Frederick is currently a Professor and Graduate Program Coordinator in the Human Factors and Systems Department at Embry-Riddle Aeronautical University in Daytona Beach, Florida. Dr. Frederick received her Ph.D. in 1991 from the University of Rochester with a major in Psychological Development. In 2000, Dr. Frederick joined the Human Factors and Systems Department at Embry-Riddle, where her work focused on applied motivation and human factors issues in aviation/aerospace. Dr. Frederick's current research interests examine how individual differences interact with technology to enhance educational engagement and performance.