122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Paper ID #12682

Applying Second Language Acquisition to Facilitate a Blended Learning of Programming Languages

Dr. Lulu Sun, Embry-Riddle Aeronautical Univ., Daytona Beach

Lulu Sun is an associate professor in the Engineering Fundamentals Department at Embry-Riddle Aeronautical University, where she has taught since 2006. She received her B.S. degree in Mechanical Engineering from Harbin Engineering University (China), in 1999, and her Ph.D. degree in Mechanical Engineering from University of California, Riverside, in 2006. Before joining Embry-riddle, she worked in the consulting firm of Arup at Los Angeles office as a fire engineer. Her research interests include engineering education and its pedagogies relating to programming language, and engineering graphics. She is a professional member of the Society of Fire Protection Engineer, and a member of American Society of Engineering Education.

Dr. Christina Frederick, Embry-Riddle Aeronautical University

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. She previously taught at the University of Rochester, Southern Utah University and the University of Central Florida. 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 also served in various roles in University administration between 2004-2012, including Vice President for Academics and Research. Dr. Frederick's current research interests examine how individual differences interact with technology to enhance educational engagement and performance. Dr. Frederick is the author of more than 50 research publications, 4 book chapters and over 60 regional, national and international conference presentations on a wide range of topics in human factors and psychology. She is active in a number of professional associations, and is a Consultant for Psi Chi, the National Honor Society in Psychology.

Applying Second Language Acquisition to Facilitate a Blended Learning of Programming Languages

Abstract

This paper describes a recent NSF funded project under the Research Initiation Grant in Engineering Education (RIGEE) program. It correlates the programming language study to second language acquisition theory. The work begun in Fall 2014, and project materials are under development. This paper outlines the proposed work and the materials developed to support the implementation of the project in Fall 2015.

Introduction

Computer programming is a common mandatory course taught in the first year of engineering and computer science programs. These types of courses typically utilize a common programming language (MATLAB, C, Java) to teach students about syntax, programming techniques, and introduce students to applied problem solving[1-4]. Learning a computer programming language has been known to be difficult for high-school and university students because of the lack of time for practice[5], in addition to the conceptual complexity of the topic and logical reasoning processes required for understanding. Programming courses are critical to the learning needs of students in STEM majors, as they provide students with problem solving skills that are easily transferrable and contextually relevant to math and science courses in the curriculum. A student who is better prepared to understand and solve problems, regardless of the context, will be better prepared to persist throughout higher education.

Learning a programming language can be seen as analogous to learning a foreign or second language since both involve the appropriate use of vocabulary (keywords), grammatical structures (syntax), and punctuation (symbols) that people need to understand in order to communicate with the computer[5-9]. Just as knowledge of vocabulary, grammar, and punctuation does not make someone fluent in a spoken language, being a successful programmer requires more than just rote-knowledge. Current introductory programming courses often struggle to provide enough problem solving because so much time is spent on learning the rote elements of the language[10]. The proposed work is to apply well-developed cognitive frameworks used in Second Language Acquisition (SLA) to facilitate a Blended Learning (aBLe) experience of programming languages (SLA-aBLe), which will accommodate a variety of learning needs and abilities. This will potentially increase student engagement in online components of the course[11], providing better preparation for face-to-face classes. The classes can then focus on specific problem solving needed in other general education courses, instead of just keywords, syntax, and symbols. It will encourage the development of problem solving skills students need to persist in a lifetime of learning.

The research questions that will broadly guide the beginning of this work will include:

- Will SLA-aBLe help engage students to learn in a simplified and easy to understand environment?
- Will SLA-aBLe improve student performance in programming language study?

- How do student's individual differences such as student demographics attributes impact effectiveness of SLA-aBLe?
- How does SLA-aBLe affect student problem solving ability?

Previous Work

The Introduction to Computing for Engineers (EGR115) course is one of the most difficult introductory level courses offered at a private institution in the Southeast. The course's main issue is the lack of practice time. Combined with the algorithm-centric nature of programming, it results in inadequate comprehension of the course material. The course has been revised multiple times in response to comments from students and faculty. One of the most significant changes was switching from programming in C to programming in MATLAB in the fall of 2009, since MATLAB has become the major language used in various engineering disciplines for problem solving[12-14]. Following this, the course changed its meeting time from three times a week to four times a week. It now uses a 2+2 format: two days of lecture per week, with each lecture day followed by laboratory time to facilitate material understanding by hands-on practice. Approximately 120 students attend a one-hour lecture in an auditorium. The following day, students attend a small lab session, usually 26 students, to allow more contact with each student while s/he practices. There has been concern voiced regarding large lectures with respect to attendance rates, effectiveness of large lecture instruction, and connectivity between the instructor and students[15].

To provide a more flexible learning environment and improve student learning outcomes[16-18], a blended learning approach was adopted in 2010 by approximately half of the EGR115 sections and is still in use today. The general format of the blended and traditional course remains the same: 2 hours of lectures per week and 2 hours of lab time per week. However, in the blended course, each 1-hour lecture in the auditorium is replaced by online self-study activities which also last one hour. Thus, instructor and students only meet face-to-face twice a week, during the lab time to solve student's problems and help them with hands on practice. The self-study online activities consist of watching recorded audio-visual PowerPoint lectures, joining online discussion, and completing exercise/quizzes before each face-to-face lab time. With 24/7 unlimited course content access online, students have more flexibility to learn at any time as often as they want.

Unfortunately, from previous course assessment it was shown that students were still afraid of learning a programming language because of its conceptual complexity and logic reasoning process[19]. There was still a lack of engagement in the online instruction and therefore a lack of preparation for face-to-face exercises that needed to be addressed. Therefore, the over-arching problem is to reduce the intimidation and anxiety associated with learning programming languages, and provide a more effective online learning environment to engineering students.

Proposed Work

Learning a programming language is analogous to students acquiring a second language. A programming language has vocabulary, syntax, grammar and communicative outcomes that must be sufficiently developed for the learner to function successfully in the environment that utilizes the language. Different cognitive skills are focused on at each stage of SLA with the implementation of associated instructional strategies. This proposed study utilizes an SLA approach to instruction in a programming language in a blended learning environment. In order

to involve the students in this process, simple innovative modifications to the course pedagogy will be made, so that students can see the correspondence between SLA and computer programming language acquisition. Table 1 shows a comparison of current blended learning and SLA-aBLe development.

Table 1. A comparison of current blended learning and SLA-aBLe development

	Preproduction (minimal comprehension)	**Early Production (limited comprehension)**	**Speech Emergence (increased comprehension)**	**Intermediate Fluency (very good comprehension)**	**Advanced Fluency**
Current Blended Learning	Few pictures and visuals. Some topics are not well explained. No enough self testing questions in the screencasts.	There are multiple choice questions but no simple programs. Facebook is used but there is no group discussion.	Students begin reading and writing in their programming language by solving different engineering problems.	Give students more challenging problems to synthetize what they have learned.	Open-ended engineering project to challenge their understanding and expand their knowledge.
Teaching Strategies in SLA-aBLe	Use pictures and visuals; speak slowly and use simple and shorter words to draw connection between SLA and programming languages; Reinforce learning by giving more self testing questions without adding in pressure.	Reinforce learning by asking students to produce simple programs in addition to the multiple choice questions; use Facebook to encourage group discussion.	Emphasize tiered questions and ask students to do a "think, pair, share" to process the new concepts.	Emphasize compare and contrast different concepts. Allow students to explain their problem solving process.	Project presentation opportunity will be offered to students to enhance their understanding .

At each of five stages of SLA, different proficiencies are focused on and different cognitive skills related to language learning are developed. The online study should help students accomplish the preproduction by using visuals such as pictures, objects, or animations to aid in comprehension. Early production skills can be accomplished by multiple-choice questions and short answers after online study. Facebook can be used to encourage group discussion and provide instructional assistance online. The lab practice on the second day with assigned homework will focus on the speech emergence and intermediate fluency. At the speech emergent stage, learning focuses on simple written assignments with instructional emphasis placed on correction for meaning. Students will be involved in the activities that require them to speak and do. Such activities will include peer instruction, group discussion, and "think, pair, share" exercises to advance student skill development following classroom directions. At intermediate fluency stage, students can demonstrate excellent comprehension by completing more complicated homework problems individually. Finally, at the advanced fluency stage, students develop and refine their knowledge of more sophisticated aspects of grammar and syntax when they start the open-ended final project. Project presentation opportunities will be offered to the students to enhance their understanding of the comprehensive materials learned in the whole semester.

There are four topics (data type, input and output, conditional statement, and loop) which will be designed and implemented following the SLA approach in this project. PowerPoint will be designed first to include pictures, animation, self-analysis questions, and MATLAB code demonstration. Figure 1 shows the initial design of the PowerPoint slides following SLA-aBLe development.

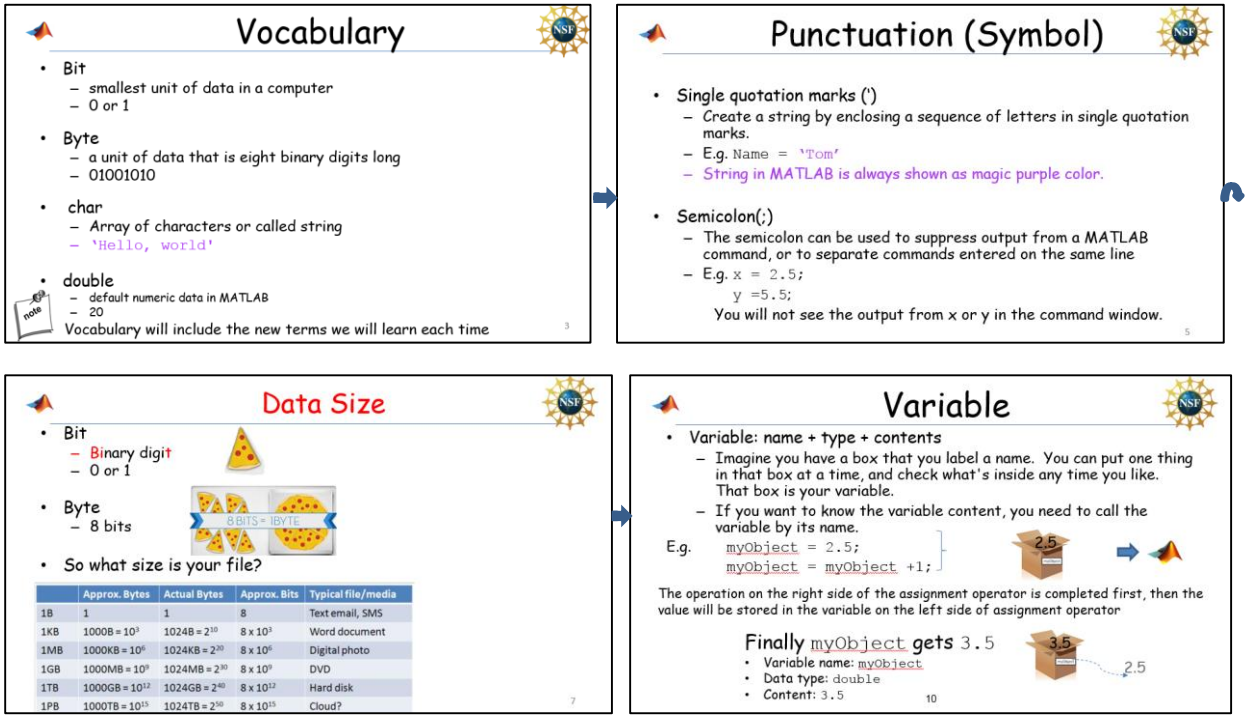

Figure 1. PowerPoint slides design following SLA-aBLe development

Conclusion and Expected Significance

This study will test the hypothesis that the use of cognitive frameworks in second language acquisition for the development of a blended learning experience of programming languages can improve engagement and the learning experience of engineering students. Using this approach will place greater emphasis on problem solving techniques that can be utilized in all courses. We expect that the qualitative and quantitative data collected during the study will help improve the implementation of the SLA-aBLe in programming courses and the use of the cognitive frameworks in alternative settings. The course modules will be disseminated to students and instructors who are either learning or teaching an introductory programming course to facilitate student learning outcomes. Likewise, the lessons learned from applying SLA-aBLe frameworks in this context will be disseminated. This will provide a richer understanding of how SLA-aBLe affects student's learning of the programming language.

Acknowledgements

The authors are grateful for the support provided by the National Science Foundation, Division of Engineering Education and Centers, grant number EEC 1441825. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Bibliography

1. Bualuan, R. (2006). Teaching Computer Programming Skills to First-year Engineering Students Using Fun Animation in MATLAB," Paper presented at the 2006 American Society for Engineering Education Annual Conference & Exposition, Chicago, IL.
2. Devnes, P.E. (1999). MATLAB and Freshman Engineering. Paper presented at the 1999 American Society for Engineering Education Annual Conference & Exposition, Charlotte, NC.
3. Morrell, D. (2007). Design of an Introductory MATLAB Course for Freshman Engineering Students. Paper presented at the 2007 American Society of Engineering Education Annual Conference & Exposition, Honolulu, HI.
4. Naraghi, M.H.N. & Litkouhi, B. (2001). An Effective Approach for Teaching Computer Programming to Freshman Engineering Students, Paper presented at the 2001 American Society for Engineering Education Annual Conference & Exposition, New York.
5. Solomon, J. (2004). Programming as a Second Language. Learning & Leading with Technology, 32(4), 34-39.
6. Tran, L. (2014) Computer Programming Could Soon Be Considered a Foreign Language in One State. Retrieved March 7, 2014, from http://www.policymic.com/articles/81067/computer-programming-could-soon-be-considered-a-foreign-language-in-one-state
7. Tyre, P. (2013) Is Coding the New Second Language? Retrieved March 7, 2014, from http://www.smithsonianmag.com/innovation/is-coding-the-new-second-language-81708064/
8. Van Roy, P., (2003). The Role of Language Paradigms in Teaching Programming. Paper presented at the 34th SIGCSE Technical Symposium on Computer Science Education, New York, NY.
9. Wynn, M., (2015). Ky. Ponders Teaching Computer Code as Foreign Language. Retrieved January 29, 2015, from http://www.usatoday.com/story/tech/2015/01/29/ky-computer-code-as-foreign-language/22529629/
10. Victor, B. (2012). Learnable Programming. Retrieved March, 7, 2014, from http://worrydream.com/LearnableProgramming
11. Marsh, D. (2012). Blended Learning Creating Learning Opportunities for Language Learners. Cambridge University Press.
12. Azemi, A., Pauley, L.L. (2006). Teaching the Introductory Computer-Programming Course for Engineering Using MATLAB and Some Exposure to C. Paper presented at the 2006 American Society for Engineering Education Annual Conference & Exposition, Chicago, IL.
13. Bjedov, G. & Andersen, P. (1996). Should Freshman Engineering Students be Taught a Programming Language. Paper presented at the Proceedings of the 26th Annual Frontiers in Education Conference, Salt Lake City, UT.
14. Herniter, M.E. & Scott, D.S. (2001). Teaching Programming Skills with MATLAB. Paper presented at the 2001 American Society of Engineering Education Annual Conference & Exposition, New York.
15. Abbitt, J. & Carroll, B. (2013). Using Technology to Enhance Undergraduate Learning in Large Engineering Classes. Paper presented at the 2013 American Society for Engineering Education Annual Conference & Exposition, Atlanta, GA.
16. Brown, C. & Meyers, D. (2007). Experimental Hybrid Courses that Combine Online Content Delivery with Face-to-face Collaborative Problem Solving. Paper presented at the 2007 American Society of Engineering Education Annual Conference & Exposition, Honolulu, HI.
17. Brown, C., Lu, Y., Meyer, D., & Johnson, M. (2008). Hybrid Content Delivery: Online lectures and Interactive Lab Assignments. Paper presented at the 2008 American Society for Engineering Education Annual Conference & Exposition, Pittsburgh, PA.
18. Yale, M., Bennett, D., Brown, C., Zhu, G., & Lu, Y. (2009). Hybrid Content Delivery and Learning Styles in a Computer Programming Course. Paper presented at the 39th ASEE/IEEE Frontiers in Education Conference, San Antonio, TX.
19. Sun, L., Kindy, M., & Liron, C. (2012). Hybrid Course Design: Leading a New Direction in Learning Programming Languages Paper presented at the 2013 American Society of Engineering Education Annual Conference & Exposition, San Antonio, TX.