Applying Cognitive Load Theory to Examine STEM Undergraduate Students' Experiences in An Adaptive Learning Environment: A Mixed-Methods Study

Dolly Bounajim[1], Arif Rachmatullah[1], Madeline Hinckle[1], Bradford Mott[1], James Lester[1], Andy Smith[1], Andrew Emerson[1], Fahmid Morshed Fahid[1], Xiaoyi Tian[2], Joseph B. Wiggins[2], Kristy Elizabeth Boyer[2], Eric Wiebe[1]
North Carolina State University, Raleigh, North Carolina[1]
University of Florida, Gainesville, Florida[2]

This study examined undergraduate STEM students' experiences using an online introductory computer programming learning environment equipped with an automated hint generation system. Following a convergent parallel mixed methods design, this study utilized both quantitative and qualitative data from student experiential data. Analysis by level of prior knowledge demonstrated that elements of the learning environment did not cater to their learning needs and cognitive architecture. Cognitive Load Theory was used to contextualize system elements against both higher and lower prior experience learners, ultimately pointing to a need to design better scaffolds and hints to the needs of novice CS learners.

INTRODUCTION

An understanding of human cognitive architecture through cognitive load theory can guide efforts to promote optimal learning systems. For efficient learning to occur, learning systems should consider the limitations of working memory, especially as it relates to prior knowledge. This paper explores user experiences using an online adaptive system for undergraduate students in introductory computer science courses. When analyzed through the lens of cognitive load theory (Sweller, 2011), examining feedback from learners based on different levels of prior knowledge helps unpack their unique needs.

Cognitive Load Theory and Prior Experience

This paper utilizes a cognitive load theoretical framework to inform aspects of instructional design, as cognitive load impacts learners' experiences when using a learning environment with automated assistance. Cognitive load theory (CLT) can be utilized as an instructional theory based on aspects of human cognitive architecture as it is integrated with instructional design and learning procedures (Sweller, 2011). It distinguishes between two main types of cognitive load: intrinsic and extraneous (Kalyuga & Liu, 2015) which additively contributes to the accumulated cognitive load of the learner (Windell, Wiebe, Converse-Lane, & Beith, 2006). For efficient learning to occur, total cognitive load should abide by the limitations of working memory (WM) capacity (Kalyuga, 2007).

A learner's level of intrinsic cognitive load is driven by both the inherent complexity of the learning material and the learner's prior knowledge. Extraneous cognitive load diverts valuable cognitive resources towards tasks irrelevant to learning, and is also influenced by learner prior knowledge (Kalyuga, 2007). For example, extraneous load can be the result of nonoptimal instructional procedures. While there are many techniques to combat typical sources of extraneous load, what is optimal for a novice learner can present suboptimal conditions for an expert. The effectiveness of various instructional techniques is determined by the levels of prior knowledge of the learner (Kalyuga, 2007). Intrinsic load is controlled by the appropriate match of designed element interactivity and user expertise. Thus, optimal instruction for novice learners lacking sufficient task-specific knowledge requires support through appropriate scaffolding. Scaffolded systems provide guidance through additional designed supports that lower element interactivity to accommodate intrinsic load. Guidance based in cognitive load theory can prevent unproductive search activities associated with problem solving, as it is associated with a large number of interacting elements that generate a heavy extraneous cognitive load (Sweller, 2011). A form of this support is exhibited by the worked example effect, which is demonstrated when novices have increased learning performance by way of studying a problem and its solution rather than partaking in an unguided search to solve the problem by themselves.

Hints and feedback are a form of external scaffold that are designed to compensate for a lack of sufficient prior knowledge. The provision of minimal feedback can have substantial benefits on learning and development (Fyfe & Rittle-Johnson, 2016). Providing novice learners simple feedback as to whether answers are correct or incorrect can lead to significant increase in performance. However, as predicted by CLT, forms of hints and feedback can also hinder the performance of more advanced learners who already have the targeted prior knowledge, In this case, advanced learners are presented with extraneous, distracting material when it would be more effective for them to problem solve on their own (Fyfe & Rittle-Johnson, 2016). This hindrance highlights the *expertise reversal effect*, which occurs when instructional procedures that facilitate novices' learning may hinder experienced users (Kalyuga, 2007). This reversal can be attributed to instruction that does not align with a learners prior knowledge (i.e., cognitive architecture), prompting them to dismantle or bypass their more cognitively efficient automated schemas.

Adaptive learning systems that do not correctly target the prior knowledge of learners—matching their human cognitive architecture—can impose negative affective responses. Learners who struggle with nonoptimal systems may be subjected to higher feelings of stress and frustration (Sharek & Wiebe, 2014). Examples of this complex relationship have been noted in introductory computer programming courses (Stachel et al., 2013). Computer programming integrates a variety of novel concepts germane to the field which pose additional cognitive workloads for novices, both abstract concepts of program structure along with learning the interface of the programming environment. Thus, it is evident that attributes of programming can create challenging learning conditions for novices, exposing them to multiple factors that potentially overload WM capacity and increase stress and frustration (Stachel et al., 2013). Similarly, an adaptive system that does not take into account a learner's existing computer programming knowledge may also create frustration if the system elicits extraneous load through the expertise reversal effect.

We predict that student feedback of the system will reflect elements of incongruence between the system scaffolds and their cognitive needs. This will be based, in part, on their cognitive architecture related to their level of prior knowledge.

RESEARCH QUESTION

This analysis was guided by the following research question:

> *To what extent does prior knowledge influence undergraduate STEM students' experiences in a computer science learning environment that provides automated assistance?*

METHODS

Participants

The sample for this study was 215 undergraduate STEM students in a university located in the Southeastern region of the United States. Their age ranged from 17 to 35 (M = 19.92, SD = 11.87). The students were broadly diverse in terms of their year in college, gender, ethnicity, and prior computer programming experience. Of the participants who preferred to answer, 68% identified as male and 31% as female. Most participants were in their freshman year (77%), 11% were sophomores, 9% juniors, and 2% seniors. Of the total participants, 57% were White, 20% Asian, 9% Black, 6% Hispanic, 1% Native American, 2% gave Other as their response, and the remaining 5% preferred not to answer. When asked about having prior computer programming experience, 41% answered Yes and 59% answered No.

Materials

PRIME is a system designed for undergraduate students in introductory computer science courses. The online system utilizes Blockly (Fraser, 2013), a block-based programming language, and is divided into three units. The units introduce foundational computer science concepts such as variables, loops and functions via programming activities. The system employs a form of instructional scaffolding which provides students with increasingly fewer instructions and assistance as they progress through the activities (i.e., faded scaffolding). In addition, PRIME provides students with automated hints as they work through the programming challenges, but are only triggered when they hit specific errors or roadblocks within their programming. For example, blocks not connected to the "start" block will trigger a hint button that prompts a hint saying "make sure your blocks are connected to the start block before running your code."

Context

The research study was structured as an extra-credit opportunity for students in an introductory engineering course. PRIME learning material did not overlap with content of the course in which participants were recruited. Students interacted with PRIME in their free time outside of class, and were offered extra credit for attempting and/or completing activities.

Design

To pursue the stated research question, this study utilized a convergent parallel mixed methods design, in which both quantitative and qualitative data were analyzed separately and the results compared (Creswell & Creswell, 2017). This research design was chosen because the quantitative and qualitative data collected for this study both describe students' experiences with the scaffolded system in different ways, making the separate analysis followed by direct comparison an optimal approach. The convergent parallel mixed methods design requires both forms of data to consist of the same variables or constructs in order to produce valid results. Both the quantitative and qualitative data used in this study explored the concept of user experience within the PRIME system, suggesting that the data could be converged using the side-by-side comparison approach. Using this approach, qualitative themes are discussed in reference to the statistical findings to either confirm or disconfirm results (Creswell & Creswell, 2017).

Measures and Analyses

Three different units of introductory computer programming presented in the PRIME system were used in this study. Before starting, students took a CS assessment measuring their understanding of core computer science concepts (i.e., variables, loops, conditionals, and algorithms). This assessment was a subset of items from an assessment developed by Rachmatullah et al. (2020) and consisted of 26 multiple-choice questions (Cronbach's *alpha* = .863). At the end of each unit students were asked to respond to five seven-point Likert-type experiential questions around the adequacy of instruction, length, difficulty, needing assistance, and feeling of frustration about the previous unit. These questions were an adapted version of the UES survey validated by Wiebe et al. (2014). Additionally, students were also asked to answer an open-ended question about their experience with the unit they just worked on.

Students were grouped into three levels of prior knowledge based on their pre-intervention CS assessment scores: low, medium, and high. Kurtosis and Skewness were used to determine the normality of the data (George & Mallery, 2010), and ANOVA tests were run to identify whether there were differences in the five experiential questions between prior knowledge groups. Partial eta squared (η_p^2) was used as the measures of effect sizes, with 0.01, 0.06, and 0.14 representing small, medium, and large effect sizes (Cohen, 1988).

Students' responses to the open-ended question were analyzed qualitatively using constant comparative methods and through open coding (Strauss & Corbin, 1998). Axial coding was then conducted after codes from the open coding process were identified. The process of axial coding resulted in a codebook that was used by the second coder to code 30% (randomly selected) of the data. The agreement percentage between the first and second coders ranged from 72% to 98% for each category, and 89%, on average, indicating satisfactory agreement (Kurasaki, 2000). Any disagreements were then discussed between coders and resolved. A selective coding was then conducted to generate emerging themes. We compared the results from quantitative analysis to these emerging themes; whether they confirm or disconfirm one another.

RESULTS

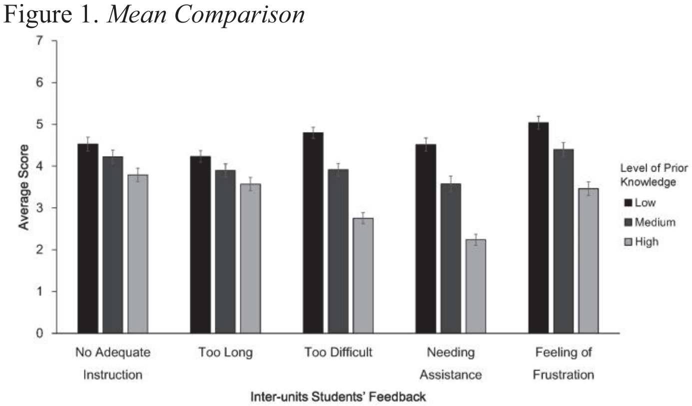
Quantitative Results

Using the five experiential questions, ANOVA tests were run to explore whether there were any impacts of students' prior knowledge on their experience with the PRIME system. The ANOVA results are presented and visualized in Table 1 and Figure 1, respectively. It can be seen in Table 1, that there was significant (p < .05) impact of prior knowledge on the students' experiences with the PRIME system indicated by the five measured variables. We found that students with low prior knowledge consistently had the highest (most negative) average scores across all five variables, followed by medium and then high prior knowledge students. Results from pairwise comparisons indicated that the low and medium groups did not significantly (p > .05) differ in the first two variables: no adequate instruction and unit is too long. The three groups were found to be significantly different from one another in the other three variables, which may explain the larger effect size (η_p^2 > 0.10) than in the former two items (η_p^2 < 0.03).

Table 1. *ANOVA results*

Variable	Prior Knowledge	*M*	*SD*	*F[2, 411]*	*p-value*	η_p^2
Not Adequate Instruction	Low	4.53	1.96	5.28	0.005	0.025
	Medium	4.19	1.91			
	High	3.78	1.92			
Too Long	Low	4.23	1.64	4.49	0.012	0.021
	Medium	3.9	1.87			
	High	3.58	1.91			
Too Difficult	Low	4.8	1.52	57.2	< .001	0.218
	Medium	3.91	1.83			
	High	2.72	1.54			
Need Assistance	Low	4.51	1.88	52.18	< .001	0.202
	Medium	3.56	2.15			
	High	2.22	1.6			
Frustration	Low	5.04	1.83	24.6	< .001	0.107
	Medium	4.39	1.99			
	High	3.43	1.94			

Figure 1. *Mean Comparison*



Qualitative Results

The qualitative analysis provides further insight into the quantitative findings. Quantitative results showed that, when interacting with the same system, users have experiential differences based on their level of prior knowledge. The Low Prior Knowledge group showed higher frustration as well as higher levels of other negative experiences. They similarly expressed a desire for more assistance, these findings were supported by themes emerging from the qualitative results.

Theme 1: Affective experiences

Prior knowledge plays an important role in students' affective experiences imparted by the system. Students with low prior knowledge felt activities were not well designed for beginners, and reported higher levels of negative emotion, including frustration. Novices, when expressing negative emotions with the design of the activities, made statements that highlighted the relationship of cognitive load and frustration:

> "There is nowhere near enough instruction to complete these activities. If you don't know how to do it and then ask for a hit, the hint simply repeats itself every time. This is frustrating." (Student A, low group)

> "I felt frustrated completing this assignment because I felt like if I had only a little more explanation I would be able to move forward." (Student B, low group)

While mid-level students expressed many of the same concerns, they were fairly neutral in their emotional valence. They made statements concerning the adequacy of instruction and suggested modifications in ways that avoided the use of emotion in their wording. There were statements related to their openness to more challenge for both mid and high-level students:

> "I actually really enjoy the problem solving aspect of this activity." (Student C, high group)

Students with high prior knowledge expressed frustration due to issues with system latency. In addition, they also expressed frustration when they created an acceptable solution that did not replicate the solution required by the system:

> "My answer to 1.7 was correct but was not being counted as such. Frustrating" (Student D, high group)

Theme 2: Hint Utilization

Students' level of prior knowledge dictated their ability to effectively utilize hints. While students with low prior knowledge felt the most in need of hints, they reported that when they did receive hints, the hints did not provide them with the insight they needed to progress. For these learners, hints seem to not be supportive because their prior knowledge didn't allow them to leverage the hint content. Thus, they did not understand the hints:

> "If there were hints that applied to every individual step of the activities, it would be much easier." (Student E, low group)

> "The hints were useless and I was frustrated." (Student F, low group)

Hint feedback from mid-level students revolved more on the desire for increasing hint frequency rather than not understanding the existing hint. For both low and mid-level students, desire for more hints may be a result of the system not triggering hints, as they are only available when specific code block configurations are met. Thus, if students are not actively manipulating blocks and trying different configurations, hints will not be triggered:

> "I wish there were more hints for what tools to use for which activities." (Student G, mid group)

Students with high-level prior knowledge did not report the same level of statements regarding hints as the other two groups, possibly due to their higher performance and thus not triggering hints.

Theme 3: Feedback on scaffolding

As with the response to the hints, student feedback toward scaffolding differed based on prior knowledge level. Students with low levels of prior knowledge often demonstrated help seeking behavior via requests for more scaffolding—both through hints, but also through other interface elements. In particular, they wanted support through breaking down information into more steps, and providing more instructional guidance throughout those steps. Beyond adding to the existing formal instructions, low-level students requested guidance in the form of both feedback as to why answers are incorrect (as opposed to the more indirect hints), as well as in the form of visualizations. These needs were also articulated through requests for more external guidance from feedback and worked examples:

> "Hardly any visuals to help you or confirm what you are doing right or wrong. This is super frustrating" (Student H, low group)

Students with mid-level prior knowledge also sought additional support, but were not as specific in their requests as low prior knowledge students. For high prior knowledge students, criticism towards scaffolding was reported through requests for fewer scaffolded boundaries that would allow them to utilize their existing coding techniques:

> "I can't write the code normally. Terrible website." (Student I, high group)

High prior knowledge students also desired the ability to demonstrate a creative expression of their knowledge and abilities and by having the system accept various conditions that produce the correct answers:

> "I understood how to do the code, but the conditions for completion could be less rigid." (Student J, high group)

DISCUSSION

Feedback provided by the three groups classified based on prior knowledge differed both quantitatively and qualitatively, thematically aligning with their unique cognitive needs. Based on their feedback, novices, with a low level of prior knowledge, exhibited the highest level of mismatch between system supports and their cognitive architecture. High-level and mid-level prior knowledge students also had their own challenges.

As demonstrated by their affective experiences, novices exhibited signs of heavy cognitive load and frustration. Novices' frustration centered on the utility of both the hint structure and scaffolding. While novices were a primary target for the hints, the takeaway is that they were not written or situated in a way novices could effectively use them to lower cognitive load and forward learning. Novices who were unable to utilize hints implies inadequate understanding of the initial material to effectively leverage their content, driving their various feedback for further scaffolding. There was a desire for a higher degree of structuring to provide an "entry ramp" into the content. While mid-level students also sought additional support, they did not express the same level of frustration as low-level students— possibly indicating that they did not surpass the same threshold of cognitive resource deficit involved in creating frustration. Additionally, they did not share the same range of feedback, potentially pointing to their ability to utilize prior experience to know what scaffolded supports work best for them.

Qualitative findings were more revealing for high prior knowledge students regarding their concerns about system hints and scaffolds then their quantitative measures. High-level students reported a degree of frustration with system latency and rigid conditions that did not accommodate their creative solutions. These reports could indicate an extraneous load brought on by expertise reversal, as their feedback targeted elements that were intended to help those not yet familiar with coding. Alternately, it could be less cognitive load and more an issue centered on personal agency and preferred pace of work. Regardless, the scaffolding and support designed for low prior knowledge students seemed to be somewhat counterproductive for the affective state (i.e., motivation) of high prior knowledge students.

CONCLUSION

Cognitive load theory was used to guide the evaluation of a computer science learning system and its automatic hint generation system. Coupled with the use of a convergent parallel mixed methods design, analysis of quantitative and qualitative data guided by cognitive load theory lent itself to a richer interpretation of learner experience based on their level of prior knowledge. Quantitative results show that low medium and high groups had an approximately consistent downward trend, with this trend being captured by the medium level group echoing the similar concerns as the low level to a lesser degree. However, while the high prior knowledge group did follow the trend of lower negative responses in the Likert-type items, qualitative findings show that they provide feedback on elements unique from those provided by the low and medium groups. By examining limitations of systems provided to learners with varying prior knowledge, this research could be beneficial in the development of adaptive learning systems. In particular, this work pointed to understanding the differences of learner needs based on their prior knowledge and how this information could be used to guide the hint generation and scaffolded support of such an adaptive system.

REFERENCES

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Erlbaum.

Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Thousand Oaks, CA: Sage.

Fraser, N. 2013. Blockly. Google.

Fyfe, E. R., & Rittle-Johnson, B. (2016). Feedback both helps and hinders learning: The causal role of prior knowledge. *Journal of Educational Psychology*, 108(1), 82-97.

George, D. & Mallery, P. (2010). *SPSS for Windows step by step: A simple guide and reference, 17.0 update* (10 ed.). Boston, MA: Pearson

Kalyuga, S. (2007). Enhancing instructional efficiency of interactive E learning environments: A cognitive load perspective. *Educational Psychology Review*, 19(3), 387-399.

Kirschner, P. A. (2002). Cognitive load theory: Implications of cognitive load theory on the design of learning. *Learning and Instruction*, 12(1),1-10.

Kurasaki, K. S. (2000). Intercoder reliability for validating conclusions drawn from open-ended interview data. *Field Methods*, 12(3), 179-194.

Rachmatullah, A., Akram, B., Boulden, D., Mott, B., Boyer, K., Lester, J., & Wiebe, E. (2020). Development and Validation of the Middle Grades Computer Science Concept Inventory (MG-CSCI) Assessment. *Eurasia Journal of Mathematics, Science and Technology Education, 16*(5), em1841.

Rumelhart, D. E. (2017). Schemata: The building blocks. *Theoretical issues in reading comprehension: Perspectives from cognitive psychology, linguistics, artificial intelligence and education*, 11(1), 33-58.

Stachel, J., Marghitu, D., Brahim, T. B., Sims, R., Reynolds, L., & Czelusniak, V. (2013). Managing cognitive load in introductory programming courses: A cognitive aware scaffolding tool. *Journal of Integrated Design & Process Science,* 17(1), 37-54.

Strauss, B., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.). Thousand Oaks, CA: SAGE.

Sweller, J. (2011). Chapter two - cognitive load theory. *Psychology of Learning and Motivation*, 55, 37-76.

Wiebe, E. N., Lamb, A., Hardy, M., & Sharek, D. (2014). Measuring engagement in video game-based environments: Investigation of the User Engagement Scale. Computers in Human Behavior, 32, 123-132.

Windell, D., Wiebe, E., Converse-Lane, S., & Beith, B. (2006). A Comparison of Two Mental Workload Instruments in Multimedia Instruction. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(17), 1764–1766.