Promoting Meaningful Learning by Supporting Interplay within Abstraction Ladder

Sangho Suh
University of Waterloo
Waterloo, Canada
sangho.suh@uwaterloo.ca

***Abstract*—How can we express programming concepts in a more accessible form and manner? To address this question, my research explores ways to design, create, and use *coding strip*, a form of comic strip that offers corresponding code for learners to understand programming concepts in both concrete and abstract context. The motivation that drives this research is my belief that the key to efficient and effective learning lies in enabling dynamic interplay between high-level and low-level abstractions. *Coding strip* is proposed as the first step towards the goal of understanding how to design, create, and use tools that support such interplay.**

***Index Terms*—comics; coding strip; visual language; computing education; abstraction ladder; concreteness fading**

I. Introduction

Recently, computing literacy has received much attention, with countries around the world making programming a mandatory part of students' education. With this surge of interest, many have tried to increase its accessibility. However, this remains a challenge, as many students still find programming difficult. One of the, if not the main, factors contributing to this difficulty is its abstract nature [1].

Programming languages use unfamiliar syntax and arbitrary conventions without a clear explanation of the rationale for these decisions. (While showing how languages are arbitrary by nature can at least mitigate learners' confusion and frustration, this is rarely, if ever, done.) Algorithm procedures are presented as abstractions (e.g., loop) without delineation of the steps, obscuring the process. Concepts, such as data structures that have structural property, are shown as abstract diagrams that students with weaker abstract reasoning find hard to relate, understand, and take an interest in.

Prior work tackled these problems in several ways. Block-based programming languages, such as Scratch, used constructs resembling LEGO blocks to provide affordance. Interactive visualization tools, such as Python Tutor, enabled learners to step through execution steps, making computational procedures explicit [2]. Storytelling was used to contextualize and make programming experience more engaging [3].

Despite such efforts, many students still find programming intimidating. Teachers struggle to engage and motivate students and relate programming concepts to familiar real-life scenarios for meaningful learning [1]. My research explores how we can use the visual language of comics to contribute to these challenges. To this end, I present *coding strip*, a form of comic strip with corresponding code, as shown in Fig. 1.

Fig. 1: Coding strip example on *counted loop* [4]

Some comic books, such as *Hello Ruby* and *Secret Coders*, have been written to introduce computing to children. However, they do not provide corresponding code, limiting

978-1-5386-5541-2/18/$31.00 ©2020 IEEE

the opportunity to transfer the learning to traditional text-based programming. The remainder of this paper describes the research approach, progress, and plans for future work.

A. Research Approach & Progress

*"Some people, it appears, remain more or less permanently stuck at certain levels of the abstraction ladder, some on the lower levels, some on the very high levels... 'The **low-level speaker** frustrates you because he leaves you with no directions as to what to do with the basketful of information he has given you. The **high-level speaker** frustrates you because he simply doesn't tell you what he is talking about'... It is obvious, then, that interesting speech and interesting writing, as well as clear thinking and psychological well-being, require the constant interplay of higher-level and lower-level abstractions."*

— S. I. Hayakawa [5]

Abstraction ladder was used by Hayakawa as a metaphor to describe the process of abstraction, with an example featuring "Bessie the cow" at the bottom of the ladder. As we move up the ladder, "Bessie" abstracts to "cow," which abstracts to "livestock," "farm assets," "assets," and eventually to the concept of "wealth." Using this metaphor, he goes on to explain that for successful communication, one must avoid "dead-level abstracting" (i.e., getting stuck at a certain level, high or low, of abstraction) and operate on all levels of the abstraction ladder [5].

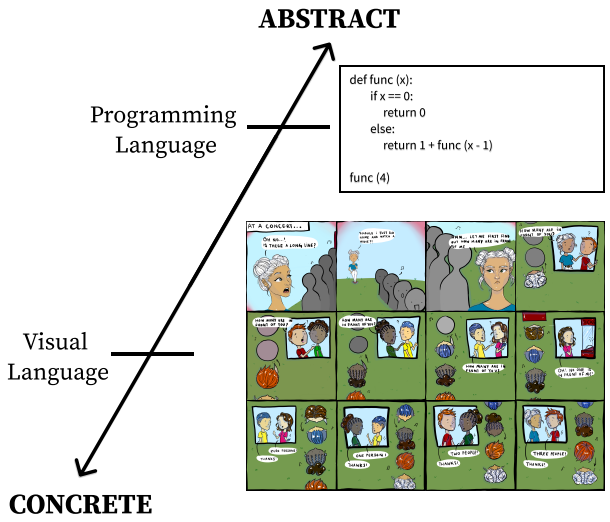

Fig. 2: Coding strip within abstraction ladder [4]

Coding strip, as shown in Fig. 2, was designed with this in mind—support interplay within abstraction ladder. For the first study, I formulated design process and tools (design board and ideation cards) for coding strip and tested their effectiveness in design workshops with high school computer science teachers and university students with prior experience in programming. The workshops were used to also identify how teachers and students would like to use coding strips [4].

Based on findings from this study, coding strips were administered at an introductory programming course. Our results showed students are excited about coding strips, and there are several benefits to using them. For instance, several students mentioned coding strips made lectures more engaging and concepts easier to understand, relate, and remember. A few students even mentioned recalling comics during their midterm to remember how while loop works.

To understand how to guide the transition from high to low or from low to high level of abstraction, I also conducted a comprehensive literature review of concreteness fading [6], a method for teaching abstract concepts by progressing from concrete (e.g., visual-based comics) to abstract (e.g., text-based programming language) representations. The findings from the review were used to design how some coding strips are used in the in-class study. For instance, a 3-stage concreteness fading design was used to introduce while loop in: comic (concrete), English (intermediate), and Javascript (abstract).

II. Future Work and Conclusion

While some work has already been done, more work is needed to ease its adoption. One of them is an authoring tool for coding strip. Currently, it is difficult to scale the creation of coding strips because illustrating comics is a time-consuming task. Fortunately, the design process and tools developed in the first study can serve as a blueprint for this authoring tool. Additionally, the development of a suite of coding strip tools that can support teaching and learning in the classroom and online is necessary, as their availability can facilitate its use. Meanwhile, more coding strips need to be created for a variety of programming concepts so that teachers interested in applying them can use them.

The proposed research explores the potential of the visual language of comics as a tool for expressing and teaching programming concepts. Particularly, I investigate how we can design, create, and use *coding strip*, a form of comic strip that has corresponding code. While the first part (design) of my research is complete, the last two are still underway. As I prepare for this next phase of my research, the consortium provides a great opportunity to receive advice and feedback on my research. I look forward to the consortium to learn, reflect, reassess, and refine my research trajectory.

References

[1] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *Acm sigcse bulletin*, vol. 37, no. 3, pp. 14–18, 2005.

[2] P. J. Guo, "Online python tutor: embeddable web-based program visualization for cs education," in *Proceeding of the 44th ACM technical symposium on Computer science education*, 2013, pp. 579–584.

[3] C. Kelleher, R. Pausch, and S. Kiesler, "Storytelling alice motivates middle school girls to learn computer programming," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, pp. 1455–1464.

[4] S. Suh, M. Lee, G. Xia, and E. Law, "Coding strip: A pedagogical tool for teaching and learning programming concepts through comics," in *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2020.

[5] S. I. Hayakawa and A. R. Hayakawa, *Language in thought and action*. Houghton Mifflin Harcourt, 1990.

[6] S. Suh, M. Lee, and E. Law, "How do we design for concreteness fading? survey, general framework, and design dimensions," in *Proceedings of the 19th ACM Conference on Interaction Design and Children*, 2020.