Using Concreteness Fading to Model & Design Learning Process

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

ABSTRACT

Concreteness fading is a technique for teaching abstract concepts, where a given concept is re-introduced in three stages with decreasing levels of concreteness. Over the years, its effectiveness has been empirically and theoretically supported in mathematics and science education, encouraging the recent adoption of the technique in computing education research. My research aims to advance our understanding of this technique and use it to support learning in computing education. The motivation that drives this research is my belief that the concreteness fading approach can have a significant impact on how we model and design learning process in computing education, and broad implications for how we design learning interfaces and systems.

CCS CONCEPTS

• **Applied computing → Computer-assisted instruction**; *Interactive learning environments*; *Computer-managed instruction*;

KEYWORDS

concreteness fading; levels of abstraction; computing education;

ACM Reference format:
Sangho Suh. 2019. Using Concreteness Fading to Model & Design Learning Process. In *Proceedings of International Computing Education Research Conference, Toronto, ON, Canada, August 12–14, 2019 (ICER '19),* 2 pages.
https://doi.org/10.1145/3291279.3339445

1 INTRODUCTION

The theoretical framework of concreteness fading was first conceptualized by Bruner as an instructional technique to facilitate the learning and transfer of abstract ideas [2]. As shown in Fig. 1, he believed that new concepts and procedures could be better delivered when presented in three progressive forms—enactive form, a physical or concrete model of the concept; iconic form, a graphic or pictorial model; and symbolic form, an abstract model of the concept.

While concreteness fading is generally used to assist the assimilation of abstract concepts, it has the potential to offer insight on how we view, understand, and interact with concept. For instance, the three stages of concreteness fading, concrete-intermediate-abstract, can be used to model three "levels of abstraction," a notion often cited when highlighting the importance of abstraction in technical disciplines, such as computer science and engineering. Donald Knuth argued that a trait that distinguishes computer scientists and engineers is their ability to transition from varying "levels of abstraction" to simultaneously see things "in the large" and "in the small [5]." Two other Turing Award recipients, Juris Hartmanis and Dijkstra, supported this idea, saying that computer science deals with phenomena of an enormous range of scale and that abstraction is "the only mental tool" that can help computer scientists operate and piece these varying levels of abstraction together [3].

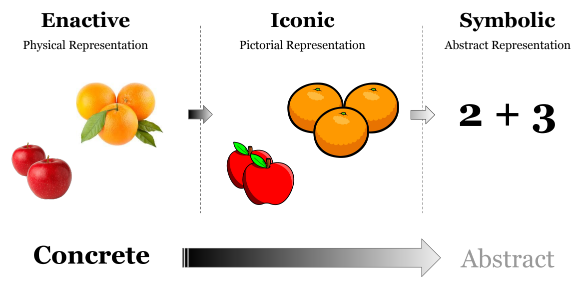


Figure 1: Bruner's framework for concreteness fading

While I am excited by the proposed importance of this notion in computing education, what fascinates me, even more, is its potential to broadly impact the way we design learning interfaces and systems. For example, consider 'explorable explanation' which is a form of interactive medium that engages readers in active reading by allowing them to interact with the simulation of the presented idea [8]. Also, take dynamic textbook, for instance. Its ability to quickly adapt its contents and engage students in learning is starting to make it a popular choice over traditional static textbooks among schools. As learning interfaces and systems become increasingly interactive, it is exciting to envision how enabling these media to present a concept and explanation at different levels of abstraction and support navigation between them may yield new ways to model and design learning process. To this end, I seek to explore the following research questions:

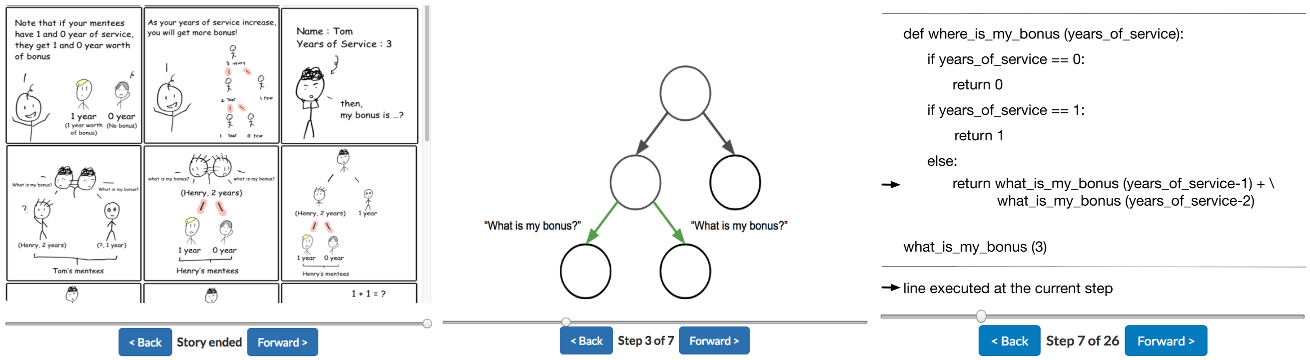
- How should we design interface and interaction methods to support concreteness fading?
- Can we introduce novel interactions for learning using concreteness fading?

2 RELATED WORK

One of the key works that shape my dissertation is a recent paper by Fyfe and Nathan [4]. They provide a theoretical ground and

ICER '19, August 12–14, 2019, Toronto, ON, Canada

ACM ISBN 978-1-4503-6185-9/19/08.
https://doi.org/10.1145/3291279.3339445



(a) Concrete Stage (b) Intermediate Stage (c) Abstract Stage

Figure 2: Learning progression used to study how concreteness fading can be used to teach recursion. Participants moved from comic to visualization and then to code.

motivation for my research. The authors note that despite years of research in concreteness fading, it remains an underspecified theory of instruction with some critical gaps in the literature. To address these gaps, the authors contribute (1) clarification of the terms concrete, abstract, and fading; and (2) six testable hypotheses that can be used to establish design principles for concreteness fading. My research uses the theoretical underpinnings presented by Fyfe and Nathan, and will explore most, if not all, of the suggested hypotheses to improve our understanding of how to implement concreteness fading.

Other works in computing education close to my research include those of Trory et al. [7] and Arawjo et al. [1]. Trory et al. explored how to design augmented reality-based learning environment for teaching computing concept (internet routing), using the concreteness fading approach. Inspired by the success of Dragonbox [6], a mobile game app that uses concreteness fading to teach algebra, Arawjo et al. [1] studied the potential of teaching programming with gamified semantics where programming constructs are initially presented as physical, familiar manipulative, such as reflecting glass, and gradually fade into abstract constructs resembling traditional text-based programming languages.

3 RESEARCH APPROACH AND METHODS

To understand how to design for concreteness fading and how it supports learning, I conducted a comprehensive systematic literature review, using five research databases associated with computing (ACM Digital Library and IEEE Xplore), education (ERIC), and multidisciplinary (SCOPUS, Google Scholar) areas. The analysis of 221 papers unveiled valuable insights, such as how it was implemented, what type of tasks and concepts were tested, and what groups benefited from this technique. By analyzing a diverse set of implementations, I formulated a general framework for concreteness fading and derived design dimensions. The summary of the other insights concerning the identified effect of concreteness fading on different tasks and groups are used for formulating hypotheses.

I have also conducted an empirical study with CodingStrip[1], a proof-of-concept programming learning environment that employs concreteness fading, as shown in Fig. 2. After participants learned recursion using the progression in CodingStrip, they were given a transfer test. In it, they were asked to solve recursion problems on 'abstract code,' which has the same structure as the 'concrete code' they learned but had arbitrary naming schemes and different input values. Pre- & post- questionnaires, interview, observation, and think-aloud methods were also used to study the effect of our intervention on their interest in learning programming, and perceived difficulty of learning programming. Additional computing concepts are in the process of being added to CodingStrip.

4 CONCLUSION

My research is still at an early stage, and thus, I would greatly benefit from attending the doctoral consortium. Some of the challenges I am looking forward to discussing at the consortium are identifying (1) the next step with CodingStrip and (2) the best way to design experiments for some future projects. The consortium will be a great opportunity to get some feedback on the planned experiments and advice on how to best prepare and conduct them.

REFERENCES

[1] Ian Arawjo, Cheng-Yao Wang, Andrew C Myers, Erik Andersen, and François Guimbretière. 2017. Teaching programming with gamified semantics. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM, 4911–4923.
[2] Jerome Seymour Bruner et al. 1966. Toward a theory of instruction. Vol. 59. Harvard University Press.
[3] Edsger W Dijkstra. 1972. The humble programmer. Commun. ACM 15, 10 (1972), 859–866.
[4] Emily R Fyfe and Mitchell J Nathan. 2018. Making "concreteness fading" more concrete as a theory of instruction for promoting transfer. Educational Review (2018), 1–20.
[5] Juris Hartmanis. 1994. Turing award lecture: On computational complexity and the nature of computer science. *Commun. ACM* 37, 10 (1994), 37–44.
[6] Nyet Moi Siew, Jolly Geofrey, and Bih Ni Lee. 2016. Students' algebraic thinking and attitudes towards algebra: the effects of game-based learning using Dragonbox 12+ App. The Electronic Journal of Mathematics & Technology 10, 2 (2016).
[7] Anthony Trory, Kate Howland, and Judith Good. 2018. Designing for concreteness fading in primary computing. In Proceedings of the 17th ACM Conference on Interaction Design and Children. ACM, 278–288.
[8] Bret Victor. 2011. Explorable Explanations. (2011). http://worrydream.com/ExplorableExplanations/

[1] https://codingstrip.github.io/