Computer Science Student-Centered Instructional Continuum

Jane Waite
Queen Mary University of London, UK

Christine Liebe
Colorado School of Mines, USA

ABSTRACT

See conference copy for DOI and to reference.

The Computer-Science Student-Centered Instructional Continuum (CS-SCIC) is a new framework to support PreK-12 instructors in their lesson design. Educators are faced with choices when building lessons; there is a tension between direct instruction, constructivism and constructionism and difficulty in providing differentiated instruction. Theoretically aligned to Vygotsky's zone of proximal development, CS-SCIC places research-based instructional strategies on a simple learning continuum. Teachers use the continuum to discuss, review and design learning events. Used internationally, initial qualitative feedback from teachers who attended pilot CS-SIC workshops was emphatically positive. Future work includes more feedback from academia and formal research, including pre and post-professional development workshop surveys.

CCS CONCEPTS

• **Social and professional topics → K-12 education**.

KEYWORDS

computer science education, differentiation, scaffolding, K-12

ACM Reference Format:

Jane Waite and Christine Liebe. 2021. Computer Science Student-Centered Instructional Continuum. In *Proceedings of Association for Computing Machinery. (SIGCSE '21).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION/PROBLEM

There is a lack of research on how to teach computer science in schools but there is increasing suggestion from academic studies that a blended approach should be used [6]. Current trends in CS PreK-12 instruction tend to focus on either structured copy code software offering immediate feedback (i.e. Code.org, programming language apps, and IDE's), or more unstructured tinkering, open projects and inquiry-based learning. Tinkering (e.g. learning by doing), or pure constructionist experimentation, is prized by some [3], and evidenced as "natural" for precocious or savant students who enjoy this approach outside of class [1]. More structured activities, such as code reading tasks or Parson's Problems, provide greater scaffolding and are being increasingly portrayed as important in computer science education [6].

SIGCSE '21, March 2021, Toronto, Canada

ACM ISBN https://doi.org/?…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 CS-SCIC: A CONTINUUM OF INSTRUCTIONAL STRATEGIES

Computer-Science Student-Centered Instructional Continuum (CS-SCIC) provides a simple overview of instructional strategies along a continuum of scaffolding, from most tightly controlled to most loose; in doing this, it highlights opportunities on how to differentiate instruction. CS-SCIC has six instructional approach categories of: 1. Copy code: students are given step by step instructions to follow e.g.copy an example program; 2. Targeted tasks: students are given a short task e.g. fix buggy code, Parsons Problems; 3. Shared coding: the teacher thinks aloud as they design and write code, sometimes called demonstrating or live coding e.g. teacher models how to write a program; 4. Project based: students are provided with a project goal and create a solution e.g.create a quiz in Scratch; 5. Inquiry based: students consider a scenario or question and create a solution e.g. explore a set of code commands and discover ways to use them; 6. Tinkering: completely unstructured student-led exploration e.g. explore a software. CS-SCIC's more scaffolded activities are likely to bridge a learner's zone of proximal [5] through the task itself, whereas at the other end of the continuum, in less scaffolded tasks, any gap must be bridged independently by learners accessing support from more knowledgeable others.

3 PREVIOUS WORK, PILOT RESULTS AND FUTURE WORK

The continuum was created for professional development (PD) as translational work from a literature review and since has reached around 250 K-12 teachers in the United Kingdom, Ireland and Germany [7, 8]. Educators have also used the continuum as a framework to review teaching tools [2]. More recently, a 2 hour pilot study of the impact of CS-SCIC on PD has commenced in the United States. The majority (75%) of pilot teachers (n=8) reported they "absolutely" improved their CS teacher knowledge and skills. In next steps, we will obtain feedback from academic colleagues on the continuum, pursue research to assess the impact of CS-SCIC, and are interested to explore "localisation" across countries and educational phases.

REFERENCES

[1] Morgan G Ames. 2019. *The charisma machine: The life, death, and legacy of One Laptop per Child.* Mit Press.
[2] Cheung Helena and Eleni Vasileiadou. 2018. Tangible Programming In Eyfs Ks1. Online. https://helloworld.raspberrypi.org/issues/5/pdf
[3] Seymour Papert. 1980. Mindstorms: children, computers, and powerful ideas Basic Books. *Inc. New York, NY* (1980).
[4] Leo Porter, Mark Guzdial, Charlie McDowell, and Beth Simon. 2013. Success in introductory programming: What works? *Commun. ACM* 56, 8 (2013), 34–36.
[5] Lev S Vygotsky. 1978. Mind in society (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.).
[6] Jane Waite. 2017. Pedagogy in teaching computer science in schools: A literature review. *London: Royal Society* (2017).
[7] Jane Waite. 2018. A continuum of scaffolding: from copying code to tinkering. Online. https://blogs.kcl.ac.uk/cser/2018/01/05/a-continuum-of-scaffolding/
[8] Jane Waite. 2018. Shared Coding, Tinkering And Other Techniques For Teaching Programming. Online. https://helloworld.raspberrypi.org/issues/4/pdf

Figure 1: Computer-Science Student-Centered Instructional Continuum (CS-SCIC) used with PreK-12 US pilot teachers

Figure 2: Instructional activity cards used with PreK-12 US pilot teachers

4 CONTEXT FOR REVIEWERS

Vygotsky's Zone of Proximal Development (ZPD) is the "extra" range of understanding which students can access through support of a more knowledgeable other (MKO) [5]. The MKO could be a person, such as the instructor, a peer (such as when pair programming) or a group to collaborate with. Alternatively, the MKO can be a tool, such as a book, software, or the task itself. In CS-SCIC, more scaffolded tasks provide the MKO through the task itself; the task has built in extra knowledge that the student needs. Whereas, in less scaffolded tasks students will need independent skills to use a wider range of MKO, such as finding things out on the internet. CS-SCIC helps teachers identify those learning activities which will require this extra mastery and flag the need to ensure students have the necessary skills and MKO available for them to access.

In selecting what CS-SCIC categories (see Figure 1) to include in lessons, teachers balance many factors, some teachers and their students may need a great deal of support and may choose to focus on more scaffolded tasks. Teachers and students ascribing to constructionism [3], desirous of a less-structured approach to learning, may choose to focus on the less scaffolded. Those educators looking to prepare students for CS professions may include activities with peer MKO, in collaborative learning experiences [4].

To introduce the continuum to teachers, a simple collaborative card sorting activity is used, this can be face to face or online. Teachers are presented with a set of cards, each with a statement describing a teaching scenario, such as shown in Figure 2. In pairs, teachers order the cards from most scaffolded, where the task is more tightly controlled, to the least scaffolded. In doing this, the

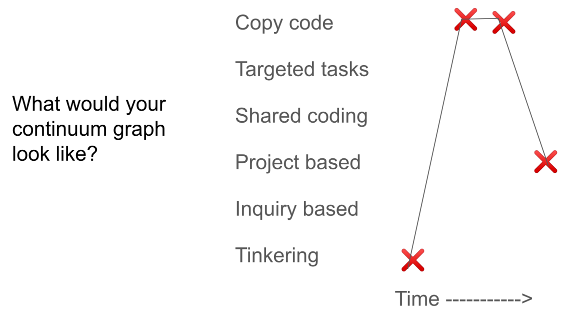

Figure 3: Example continuum graph of a sequence of teaching activities

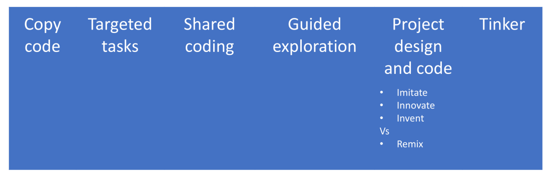

Figure 4: Original continuum of scaffolding used with PreK-5 teachers in Europe

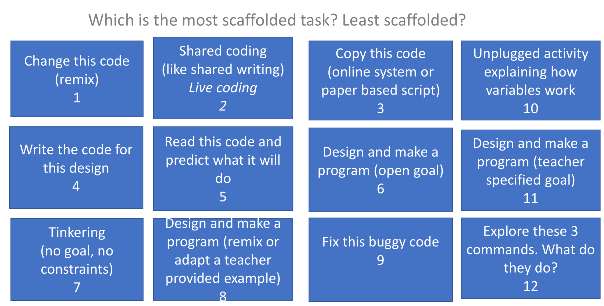

Figure 5: Instructional activity cards used with PreK-5 teachers in Europe

teachers give more generic names to the activities, they discuss and argue. Finally, the continuum is presented, see Figure 1 and teachers are asked, where on the continuum their learning activities reside. In some CPD, teachers have also been asked to plot over a sequence of lessons, the classifications of approaches used, see Figure 3. Some teachers noted, all they do is copy code, followed by a project, in doing this they started to think about other approaches they might use and how they provided the MKO in project activities.

Over time, the continuum itself has changed, initially it was presented as shown in Figure 4 in the European PD but in the US as Figure 1 to meet local focus on inquiry based learning. Similarly, as shown in Figure 5 compared to Figure 2, the sorting cards have been changed according to the context they have been used in.

It is important to note that the continuum does not prescribe one activity or category better than another. Further work is needed to compare each activity and sequences of categories.