Supporting Computational Thinking Skills for Adults

Diego Zapata-Rivera
Educational Testing Service
660 Rosedale Road
Princeton, NJ 08541
1 609 734 5141
Dzapata@ets.org

Carol Forsyth
Educational Testing Service
90 New Montgomery Street #1450
San Francisco, CA
1 415 645 8465
Cforsyth@ets.org

Hillary Molloy
Educational Testing Service
90 New Montgomery Street #1450
San Francisco, CA
1 415 645 8466
Hmolloy@ets.org

ABSTRACT
Many adult workers need to keep up with advances in technology to remain relevant in the job market. Adults now need 21st century skills including Computational Thinking. It is challenging for adults to find training opportunities that take into account their limited time, educational, and resource constraints. Our approach provides support to adult learners and their tutors to help them reach their goals. This support can take the form of facilitation messages that suggest possible learning activities, hints on study and time management for learners, and instructional suggestions and alerts for tutors based on current information gleaned from the learner and the interaction. We have collected data with adult learners and their tutors, and are designing an automated facilitator system that can provide tutors and learners with feedback according to their needs.

CCS Concepts
CCS → Applied computing → Education

Keywords
Life-long support; Adult learning; Computational Thinking Support.

1. INTRODUCTION
Learners have different needs for education and training throughout their lives. Many adults find themselves in need of acquiring or improving their technology skills to maintain and thrive in existing jobs or find new career opportunities [1]. Computer science skills are among those skills that older adults need to remain relevant in the current workforce.

Existing online programming tools and mentoring programs do not provide enough support for adult learners looking to improve the skills necessary to practice the basic concepts of computer science necessary for these adults to have access to many jobs. [2]. Mentoring programs usually involve volunteer professionals that act as tutors. These tutors have limited availability and may not have specific training in education. Our work in this area shows that these learners need more support than just training on computer programming concepts; they need support and training in other aspects of Computational Thinking (CT) (e.g., practices and perspectives), study skills, and time management.

Systems that teach computer programming have often focused on syntax-level, basic computer programming concepts, and have mostly been used with first year undergraduate students [3-5]. Although these systems have made great advances, none of them focus on the life-long learning needs of adults from underserved groups.

Working in a general framework for lifelong learning supported by technology [6], our proposed solution builds on advances in areas such as Dialogue-based Systems [7], Virtual Mentors [8], and Recommending Systems [9] to develop a tool that provides both adult learners and their tutors with relevant information and learning opportunities to help them achieve their learning and teaching goals. Although, our initial focus is on CT, we are exploring a general approach to implementing automated facilitators that can be extended to other domains.

2. COMPUTATIONAL THINKING
Computational thinking is considered a subset of computer science. CT refers to solving problems by making use of concepts, methods and processes central to computer science [10]. It involves cognitive processes such as abstraction, decomposition, modeling, pattern recognition, and algorithm design.

According to Brennan and Resnick [11], Computational Thinking (CT) includes three general aspects: computer programming concepts, practices, and perspectives. Computer programming *concepts* include sequences, loops, parallelism, events, conditionals, operators, and data. *Practices* include being incremental and iterative, testing and debugging, reuse and remix, and abstraction and modularity. Finally, *perspectives* include expressing, connecting, and questioning.

Various types of assessments of CT have been developed and deployed with middle school students [12, 13]. However, little support has been created for adult learners to better understand CT [14]. A virtual facilitator can provide this support by offering hints to tutors and adult learners in real-time. For example, the facilitator can offer teaching suggestions to tutors by considering the needs of the learners and hints of programming aspects, common practices and study and time management alerts to adult learners.

3. IDENTIFYING AUDIENCE NEEDS
We collected data from ten tutors and ten adult learners at a Bay Area Outreach program ($N = 20$). The adult learners ranged in age from 18-42 years old, 7 males and 3 females, 3 identified themselves as Asian or Asian American, 4 as Black or African American, 2 as Mexican or Mexican American, and 1 with 2 or more than one ethnicity, and 90% had at least some college education. Demographic information also showed that 20% of the learners felt completely confident with writing computer programs, 50% moderately confident, 30% somewhat confident, and 10% not

at all confident. Three were employed full time, 2 partial time, 1 was a stay at home parent, and 5 were unemployed.

Before the tutoring session, the adult learners (i.e., novices) completed surveys that allowed us to gather demographic information. They also completed individual differences assessments to aid us in better understanding the populations' needs. Then, 10 dyads (i.e., learners and tutors) were audio recorded while completing a tutoring session. Next, we interviewed key tutors and experts in areas such as adult learning (i.e., communications with adults at a Bay Area Outreach center) and tutoring adults on computer science topics for non-profit organizations.

3.1 Method

After completing the surveys and interactive tutoring sessions, we quantitatively analyzed the surveys, and then proceeded to transcribe the audio recordings of the tutoring session taking great care to code discourse moves in a pedagogically meaningful way. Finally, we interviewed experts to ensure that our findings make sense in a practical way.

3.2 Results

Survey results based on this admittedly small sample revealed that the population is indeed unique in that learners scored higher than average on individual difference measures such as Grit (M = 4.51, SD = .66) with a maximum score of 5, Growth Mindset (M = 4.52, SD = .07) and Cognitive Flexibility (M = 4.71, SD =.53), both with maximum scores of 6. The Grit scale measures one's persistence in the face of failure and passion over a long period of time [15]; Growth Mindset [16] measures the ability to view intelligence as malleable rather than fixed; Cognitive Flexibility [17] measures openness to new ways of viewing situations, ability to adapt, and disposition to believe that they can achieve can achieve the desired outcome by being flexible. It's not surprising that self-motivated adults actively seeking to learn computer science skills later in life would score high on these three scales.

Next, we analyzed transcripts and discovered 25 dialogic moves corresponding to pedagogical tactics displayed by tutors and learners. For example, moves include "knowledge check," where the teacher asks the students questions about understanding a topic of discussion; "modeling," where the teacher types code and the student watches; and "procedural tell," where the student actively types code and is the primary participant while the tutor scaffolds the students understanding. After discovering these techniques, we were able to condense the 25 discourse moves into an over- arching framework of tutoring for computer science which is beyond the scope of this paper to discuss.

Interviews with tutors and experts provided additional insight. For example, we discovered that tutors try to make the topic relevant to the learner to help motivate him or her (e.g., by working on problems relevant to the interests of the learners).

We also discovered that the tutors tend to have learners work on their own and help them only as needed. However, quite often the tutors spend too much time explaining basic concepts that learners should learn prior to working with the tutor. Also, tutors described situations that learners usually find difficult to overcome and contribute to increasing their risk of dropping out. For example, balancing work and family commitments, missing deadlines, and accumulating overdue assignments. These types of activities could be handled by a facilitator that provides supporting features like the ones shown in Table 1. This facilitator can provide learners with alerts on assignments due soon and support on basic programing concepts and examples. This is particularly important because experts that serve as tutors often are working software engineers and only have a set amount of time to help others.

4. PROVIDING CT SUPPORT FOR ADULT LEARNERS AND THEIR TUTORS

Rather than implementing a complete intelligent tutoring system, we propose to develop a facilitator that can provide relevant hints and alerts to learners and tutors. Table 1 shows sample supporting features. Tutors often spend a large amount of time explaining basic concepts such as loops, arrays, and so on. Rather than taking up the tutor's time on these concepts, the facilitator can instead point them to a relevant source.

By providing these hint and alerts, we expect that both adult learners and tutors will engage in more productive sessions. Reducing the burden on the tutor by decreasing the amount of time, cognitive energy and attention necessary to tutor a novice adult, it becomes possible for tutors to focus on other aspects that may require close attention (e.g., providing additional help on challenging topics or planning appropriate activities for learners). Also, tutors may be able to help more adults.

Table 1. Supporting Features

Feature	Audience	Description
Suggest relevant resources	Learner	e.g., sample code, discussion forums, sample dialogue exchanges, and similar projects available on the web.
Time & task management alerts	Learner	Time and project management activities (e.g., list of assignments, due dates, and meetings with the tutor).
Instructional hints	Tutor	Hints about how to deal with common errors and relevant best practices.
Alerts about learner performance	Tutor	Information about results on assignments and process log data to keep track of learner progress (e.g., alerts on learner missing assignments, problems with particular pieces of content or other risk factors associated with dropping out).

Figure 1 shows a screenshot of the tool that will be used to collect data about the perception and effectiveness of current hints and alert messages. The screenshot shows two types of facilitation messages: one to the learner providing feedback on the input selected by the learner (in this case, the *for* statement, highlighted in green), and a message for the tutor suggesting how to elaborate on best practices when debugging code (i.e., making and testing one change at a time).

The tool is implemented on top of the ETS Platform for Collaborative Assessment and Learning (EPCAL) [18]. The EPCAL platform features a modularized design with full capability to manage team formation, task progress, and receive external feedback. This platform can be used to provide private and public facilitation messages to the participants based on their interactions [19].

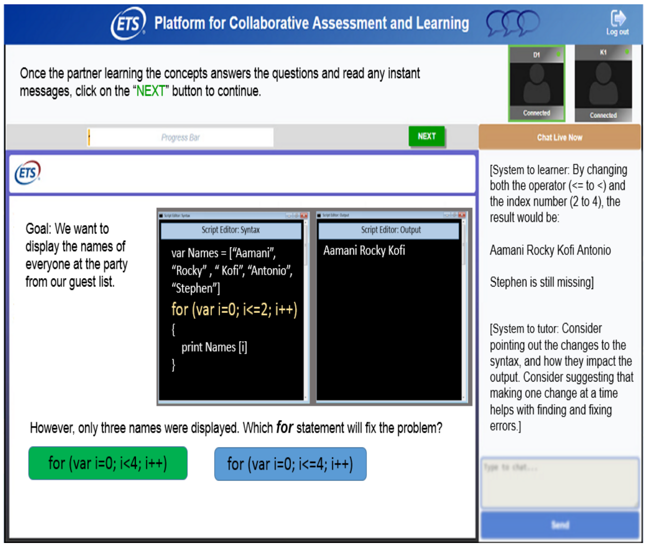

Figure 1. A screenshot of a tool that will be used to collect data and evaluate initial facilitation messages. A private message to the learner shows that by changing both the operator and the index only four out of the five names are displayed. Similarly, a private message to the tutor suggests talking about making one change at a time.

In the above interaction, both the learner and the tutor are sent private messages by the facilitator. The learner has already answered the question and now the facilitator (referred to as "system") is helping the learner better understand the underpinnings of the problem. This provides supplementary information to the learner and eases the burden on the tutor. Furthermore, the tutor is provided information to aid them in teaching the learner, as we recognize that experts in computer science may not necessary have expertise in pedagogy. Thus, the facilitator should ease the burden on the tutor by aiding the learner and the tutor.

5. A DATA-DRIVEN APPROACH

The creation of the facilitator involves collecting large quantities of data from learners and tutors and collecting and analyzing existing data from online forums and through crowdsourcing mechanisms. Machine learning algorithms will be used to find potential domain topics that learners find challenging, resources that have been used provide help (e.g., sample code, sample dialogue exchanges), and areas/situations that require the supporting alerts. This iterative data-driven approach will result in gradually refining the different supporting mechanisms of the facilitator.

6. SCALABILITY CONSIDERATIONS

By implementing a facilitator rather than a whole intelligent tutoring system, we expect to produce a system that can provide effective support in situations that adult learners and tutors find challenging. The resulting supporting features and the iterative, data-driven approach have the potential to be repurposed in other adult learning domain areas. The proposed approach can be used to continually refine the system (e.g., improving current adding supporting mechanisms) as more data are collected.

7. SUMMARY AND FUTURE WORK

The proposed system builds on advances in artificial intelligence to provide the needed support to adult learners and tutors. We expect the types of alerts and hints provided by the system will be effective in improving adult learning of CT concepts and practices while reducing the dropout rate characteristic of these types of programs. Also, we expect the facilitator will help tutors become more effective at providing learning support.

Future work will involve collecting data to evaluate different types of alerts and hints and implementing the components of the facilitator that will keep track of learner and tutor interactions to decide what alerts and hints to provide in which situations.

We expect to use this type of approach to support adult learners with other learning and training needs that they may encounter throughout their lives. The results of this project will help us refine our approach to focus on the alerts and hints that are more effective at supporting adults' and tutors' educational needs.

8. ACKNOWLEDGMENTS

Our thanks to Irvin R. Katz, Jung Aa Moon, Burcu Arslan, and Donald E. Powers for their feedback on a previous version of this paper, as well as Jennifer Lentini and Stephanie Peters for their work on this project.

9. REFERENCES

[1] Alden, J. and Taylor-Kale, L. 2018. The work ahead: Machines, skills, and U.S. leadership in the twenty-first century, *Independent Task Force Report.* Council on Foreign Relations.

[2] Brooks, K., and Lindgren, C. (2015). Responding to the coding crisis: From code year to computational literacy. In L.C. Lewis (Ed.), *Strategic Discourse: The Politics of (new) Literacy Crises*. Logan, UT: IJEMT, Vol.9, No. 1, 2015, pp. 56-67 ISSN 1882–2290 66 Computers and Composition Digital Press/Utah State University Press. Retrieved from http://ccdigitalpress.org/strategic.

[3] Naser, S. S. A. 2008. JEE-Tutor: An Intelligent Tutoring System for Java Expressions Evaluation. *Information Technology Journal*, 7(3), 528-532.

[4] Lane, H.C., and K. VanLehn, K. 2005. Teaching the tacit knowledge of programming to novices with natural language tutoring. *Computer Science Education*, 15(3), 183-201.

[5] Zapata-Rivera, J.D. and Greer, J. 2001. SMODEL Server: Student Modelling in Distributed Multi-Agent Tutoring Systems. *International Conference on Artificial Intelligence in Education AIED 2001*. 446-455.

[6] Kay, J. 2008. Lifelong Learner Modeling for Lifelong Personalized Pervasive Learning. In *IEEE Transactions on Learning Technologies*, vol. 1, no. 4, 215-228ec.

[7] Graesser, A. C., D'Mello, S. K., Hu. X., Cai, Z., Olney, A., and Morgan, B. 2012. AutoTutor. In P. McCarthy & C. Boonthum-Denecke (Eds.), *Applied natural language processing: Identification, investigation, and resolution.* 169-187.

[8] Dimitrova, V. and Brna, P. 2016. From Interactive Open Learner Modelling to Intelligent Mentoring: STyLE-OLM and Beyond. *International Journal of Artificial Intelligence in Education*, 26 (1), 332-349.

[9] Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., & Koper, R. 2011. Recommender systems in technology enhanced learning. In L. Rokach et al. (Eds*.), Recommender systems handbook: A complete guide for research scientists & practitioners*. (pp. 387-415). Berlin, Germany: Springer.

[10] Wing, J. M. 2006. Computational Thinking. *Communications of the ACM*. 49 (3): 33-35.

[11] Brennan, K, and Resnick, M. 2012. New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada. April.

[12] Grover, S. 2017. Assessing Algorithmic and Computational Thinking in K-12: Lessons from a Middle School Classroom. In *Emerging Research, Practice, and Policy on Computational Thinking*. 269-288.

[13] Snow, E., Rutstein, D., Basu, S., Bienkowski, M., & Everson, H. T. 2019. Leveraging Evidence-Centered Design to Develop Assessments of Computational Thinking Practices. *International Journal of Testing*, 19(2), 103-127.

[14] Smith, M. 2016. Computer Science for All. The White House, President Barack Obama. Retrieved from https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

[15] Duckworth, A.L., Peterson, C., Matthews, M.D., and Dennis, K. R. 2007. Grit: Perseverance and Passion for Long-Term Goals. *Journal of Personality and Social Psychology*, 92, 6, 1087-1101.

[16] Dweck, C.S. 2008. *Mindset: The New Psychology of Success: How we can learn to fulfill our potential*. Ballantine Books, New York, NY.

[17] Martin, M.M., & Rubin, R. B. 1995. A New Measure of Cognitive Flexibility. *Psychological Reports*, 76, 623-626.

[18] Hao, J., Liu, L., von Davier, A.A., Lederer, N., Zapata-Rivera, D, Jkal, P., and Bakkenson, M. 2017. EP-CAL: ETS Platform for Collaborative Assessment and Learning, *ETS Research Report No. RR-17-49*. Princeton, NJ: Educational Testing Service. https://doi.org/10.1002/ets2.12181

[19] Hebert, D., & Zapata-Rivera, D. 2018. Intelligent Support for Collaboration: A Literature Review and Proposed Framework. *Paper presented at the annual meeting of the American Educational Research Association*. New York City, NY.