Design of CoTAS: Automated Computational Thinking Assessment System

Sabiha Yeni
Leiden University
Leiden Inst.of Advanced Computer Science
s.yeni@umail.leidenuniv.nl
Leiden, The Netherland

Felienne Hermans
Leiden University
Leiden Inst.of Advanced Computer Science
f.f.j.hermans@liacs.leidenuniv.nl
Leiden, The Netherland

Abstract

Computational thinking (CT) is widely accepted as a fundamental practice for equipping students to formulate and solve problems in the digital era. Many countries are adopting mandatory curricula for computer science (CS) lessons and CT education at high school. However, assessing how well students perform in CT activities is hard. Teachers face many challenges in the assessment process, because there is a limited number of resources for assessment and a lack of online access to resources. Therefore, the goal of this paper is to support teachers by developing an effective automated Computational Thinking Assessment System (CoTAS) for instructing and evaluating CT skills of high school students in Python course. CoTAS facilitates the assessment of students' CT skills. Supported by CoTAS, teachers will be able to determine students' CT skill levels and shape their learning by continuously observing students' individual levels of development during the learning process. Teachers can access different resources to evaluate CT concepts, practices and perspectives. CoTAS can provide automatic feedback, so teachers can guide students directly when misconceptions arise. Moreover, CoTAS offers multi-disciplinary assessment tools which can be used not only in the programming lessons, but also in other disciplines such as science, mathematics and social sciences in which CT skills are integrated.

In: I. Fronza, C. Pahl (eds.): Proceedings of the 2nd Systems of Assessments for Computational Thinking Learning Workshop (TACKLE 2019), co-located with 14th European Conference on Technology Enhanced Learning (EC-TEL 2019), 17-09-2019, published at http://ceur-ws.org.

1 Introduction

Assessment plays a key role in the development of many educational reform movements. In the Horizon 2017 report, the key trends accelerating adoption of higher education technology are defined [1]. According to this report, there is a growing focus on measuring learning. This trend describes an interest in assessment and a wide variety of methods and tools that educators use to evaluate, measure, and document academic readiness, learning progress, skill acquisition and other educational needs of students. On the other hand, in a study supported by European Commission called "Developing CT in compulsory education", the authors emphasize that the evaluation of CT skills is at an early stage and there is a need for further study, and that current assessment methods are insufficient for evaluating all aspects of CT skills [2]. Research on assessment literacy indicates that particularly teachers new to a content area and teaching practice often face many challenges when it comes to engaging in robust assessment practices in their instructions [3], [4], also many teachers teaching CT lack a strong background in programming [5]. In addition, constructionist approaches are actively used in CS education. The process of evaluating students in the constructionist learning environment has several difficulties, because it is an open-ended and undefined situation. Design-oriented learning environments based on constructionist approaches require frequent evaluation in various forms.

For assessing CT, various evaluation tools are used:

tests, observations, open-ended questions, computer-based coding exams to determine levels of comprehension of programming-related terms and to shape teaching processes [6], [7], [8], [9]; artifact based interviews, portfolios, thinking aloud method, projects, rubrics to assess students' effort and development in the process [10], [11], [12], [13]; performance-based assessment, problem-based assessment, design scenarios to evaluate problem solving, algorithmic thinking and abstraction skills [14], [15], [16]; automated programming assessment tools to provide quick feedback [17], [18], [19], [20] and automated CT assessment tools to determine the CT skill levels of students [21], [22]. However, finding and validating CT measures that assesses CT with the holistic approach remains challenging. Brennan and Resnick [23] have six suggestions for assessing CT: supporting further learning, incorporating artifacts (portfolio analysis), illuminating artifact development processes, checking in at multiple waypoints, valuing multiple ways of knowing and including multiple viewpoints. Therefore, CoTAS aims to evaluate the CT concepts, practices and perspectives of high school students during education in a text-based programming language (Python). For evaluating students' comprehension of CT concepts, the automated quality assessment tool and test tool will be used [21], [24], [25], [26], [27]. CoTAS will offer problem-based assignments and problem-based tests to evaluate CT practices. Finally, CoTAS will present survey and interviews for evaluating students' CT perspectives.

2 General features of CoTAS

The goal of CoTAS is to improve the effectiveness of CT education and support teachers during the evaluation of CT skills of high school students. CoTAS provides both summative and formative evaluation tools for evaluating students' CT concepts, practices and perspectives [23], [28].

2.1 CoTAS Tools for CT Concepts

In order to follow students' comprehension of CT concepts (such as data structures, operators, conditionals, sequences, loops and functions), the automated quality assessment tool and the test tool of CoTAS will be used (Table 1). (1.1)The automated quality assessment tool will measure the proficiency level of students' codes without the need of human supervision. CT metrics are identified to measure the proficiency-complexity level of code [21], [22], [24], [25], [26]. This tool of CoTAS will alleviate teachers' struggle with manual assessment of students' CT skills, as well as providing real-time feedback on how students develop CT competency over time. (1.2) Students' knowledge level about CT concepts will be evaluated at the end of each unit with the test tool consisting of multiple choice questions in order to follow the development during the training, to detect misconceptions and to identify issues that are not understood. The contents of the test will be programming related and Bloom's lower order thinking (knowledge, comprehension and application level) questions.

The evaluation results for CT concepts: In "My Progress" page of CoTAS, four different types of assessment scores are shown in Figure 1 (1 to 4). The first part (Fig1.1) shows the proficiency level of students' projects. The percentage of projects' proficiency level will be presented in three levels (basic, developing and proficient). The second part (Fig1.2) shows the usage frequency for CT concepts according to proficiency levels.

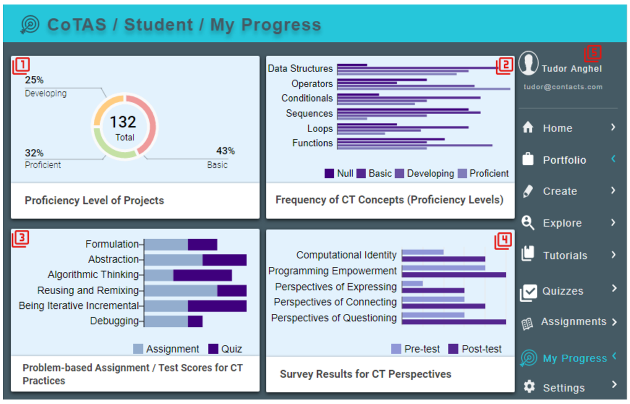

Figure 1: My Progress Page of CoTAS

2.2 CoTAS Tools for CT Practices

In order to evaluate students' proficiency level of CT practices (such as formulation, abstraction, algorithmic thinking, reusing and remixing, being iterative and incremental, debugging) problem-based assignment tool and problem-based test tool of CoTAS will be used (Table 1). (2.1) The problem-based assignment tool will include authentic assessment questions [13] and problem solving scenarios to evaluate students' CT practices. As the automatic feedback, the information comprising of the number of attempts performed while solving a problem, the duration of code writing and the similarity of the final output to the desired output will be provided to analyze the problem solving process of students. Teachers can also score assignments manually according to predetermined rubrics. The rubrics will offer descriptors of performance levels for each CT practice. (2.2) Problem-based test tool will consist Bloom's higher order thinking (evaluation, synthesis and analysis level) multiple choice questions in order to detect development in the

CT practices. This tool can be used not only in the programming lessons, but also in other disciplines such as science, mathematics and social sciences in which CT skills are integrated.

The evaluation results for CT practices: The third part of "My Progress" page (Fig1.3) presents problem-based assignment and test scores for CT practices.

2.3 CoTAS Tools for CT Perspectives

In order to evaluate students' CT perspectives (such as computational identity, programming empowerment, perspectives of expressing, connecting and questioning), CoTAS will offer the survey and interview questions (Table 1). (3.1) Students will be required to rate their agreement or disagreement with the statements in the survey. Surveys will be conducted at different time points to capture the development of students' CT perspectives. (3.2) Interviews will be used to obtain more details on students' CT perspectives; however the interview results will be manually evaluated according to predetermined rubrics so it will require time and effort.

The evaluation results for CT perspectives: The fourth part of "My Progress" page (Fig1.4) shows the survey scores for CT perspectives of students at different time points. Finally, Fig1.5 shows all actions a student can perform in CoTAS.

3 Benefits of CoTAS

CoTAS will provide different facilities for teachers, students and researchers during the assessment of CT skills of high school students. With the help of CoTAS, teachers will be able to access CT evaluation resources. The time spent for evaluation will be reduced through automatic feedback and provided resources. Teachers will be able to follow the progress of students during learning and they can manage the evaluation content easily. Teachers will see students' mistakes and misconceptions those are frequently made.

Students will be able to follow their own progress with the help of CoTAS. Students will receive instant and guiding feedback related during learning and evaluation. CoTAS will provide opportunity for accessing to resources anytime and anywhere for students. Through different assessment tools, students will be able to realize their own inadequacies and receive guidance to support their individual development.

Researchers will be able to identify the factors those are effective in improving students' CT skills by making predictive assessments. They will be able to examine whether there is a relationship between the data obtained from CoTAS (such as access frequency to learning resources or number of shared projects etc.) and students' level of CT skills. Researchers will have the opportunity to examine the effectiveness of different assessment tools in predicting students' final achievements.

4 Conclusion

Many countries have taken steps to bring the CT concepts into their respective curriculum. Although there is a high level of consensus regarding the inclusion of the concepts related to CT into the curriculum, it is known that there is resource shortage and insecurity about how this high-level thinking skill should be evaluated. In this context, CoTAS will contribute to the (inter)national CS education field. Moreover, CoTAS will provide different advantages for improving the CT skills of students. Coding is generally perceived as difficult by students, and it is one of the most difficult learning outcomes to evaluate for teachers. CoTAS will provide guided and instant feedback to students for improving the CT learning processes. CoTAS proposes to carry out formative and summative evaluation tools together in a holistic approach. Thus, it can be used not only in the programming lessons, but also in other disciplines such as science, mathematics and social sciences in which CT skills are integrated. The continuation of learning with the help of CoTAS (using automatic feedback) outside of regular lessons will provide the maximum benefit for students to reach their learning goals about CT skills.

Table 1: CoTAS Tools Used for CT Components Evaluation

CT Components	**CoTAS Tools**	**Feedback Types**	**Availability for Other Disciplines**
Concepts (Data Structures, Operators, Conditionals, Sequences, Loops, Functions)	(1.1) The automated quality assessment tool	Automatic	Python
	(1.2) The test tool (knowledge, comprehension and application level questions)	Automatic	Python
Practices (Formulation, Abstraction, Algorithmic Thinking, Reusing and Remixing, Being Iterative and Incremental, Debugging)	(2.1) The problem-based assignment tool	Manual & Automatic	Python
	(2.2) The problem-based test tool (evaluation, synthesis and analysis level questions)	Automatic	All disciplines developing CT
Perspectives (Computational Identity, Programming Empowerment, Perspectives of Expressing, Connecting and Questioning)	(3.1) The survey	Automatic	All disciplines developing CT
	(3.2) Interviews	Manual	All disciplines developing CT

References

[1] Becker, S. A., Cummins, M., Davis, A., Freeman, A., Hall, C. G., Ananthanarayanan, V. (2017). NMC horizon report: 2017 higher education edition *The New Media Consortium*, 1-60.

[2] Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., Punie, Y. (2016). Developing computational thinking in compulsory education. *European Commission, JRC Science for Policy Report.*

[3] DeLuca, C., Klinger, D. A. (2010). Assessment literacy development: Identifying gaps in teacher candidates' learning. *Assessment in Education: Principles, Policy and Practice*, 17(4), 419-438.

[4] Popham, W. J. (2009). Assessment literacy for teachers: Faddish or fundamental?. *Theory into practice*, 48(1), 4-11.

[5] De Groot, J. (2018). Teaching Computational Thinking - What do our educators need? Delft University of Technology. *MasterThesis.*

[6] Yadav, A., Burkhart, D., Moix, D., Snow, E., Bandaru, P., Clayborn, L. (2015). Sowing the seeds: A landscape study on assessment in secondary computer science education. *Comp. Sci. Teachers Assn., NY.*

[7] Grover, S., Cooper, S., Pea, R. (2014). Assessing computational learning in K-12. *In Proceedings of the 2014 conference on Innovation and technology in computer science education*, 57-62, ACM.

[8] Atmatzidou, S., Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670.

[9] Mishra, S., Iyer, S. (2015). An exploration of problem posing-based activities as an assessment tool and as an instructional strategy. *Research and practice in technology enhanced learning*, 10(1), 5.

[10] Kong, S. C. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education*, 3(4), 377-394.

[11] Kotini, I., Tzelepi, S. (2015). A gamification-based framework for developing learning activities of computational thinking. *In Gamification in Education and Business*, 219-252, Springer.

[12] Bers, M. U. (2010). The TangibleK Robotics program: Applied computational thinking for young children. *Early Childhood Research and Practice*, 12(2).

[13] Fronza, I., Ioini, N. E., Corral, L. (2017). Teaching computational thinking using agile software engineering methods: A framework for middle schools. *ACM Transactions on Computing Education (TOCE)*, 17(4), 19.

[14] Werner, L., Denner, J., Campe, S., Kawamoto, D. C. (2012). The fairy performance assessment: measuring computational thinking in middle school. *In Proceedings of the 43rd ACM technical symposium on Computer Science Education* , 215-220, ACM.

[15] Webb, D. C. (2010). Troubleshooting assessment: an authentic problem solving activity for it education. *Procedia-Social and Behavioral Sciences* , 9, 903-907.

[16] Djambong, T., Freiman, V. (2016). Task-Based Assessment of Students' Computational Thinking Skills Developed through Visual Programming or Tangible Coding Environments. *International Association for Development of the Information Society.*

[17] Ala-Mutka, K. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education*, 15(2), 83–102.

[18] Edwards, S. H., Perez-Quinones, M. A. (2008). Web-CAT: automatically grading programming assignments. *In ACM SIGCSE Bulletin*, 40(3), 328-328. ACM.

[19] Joy, M., Griffiths, N., Boyatt, R. (2005). The boss online submission and assessment system. *Journal on Educational Resources in Computing (JERIC)*, 5(3), 2.

[20] Korhonen, A., Malmi, L., Silvasti, P. (2003). TRAKLA2: a framework for automatically assessed visual algorithm simulation exercises. *In Proceedings of Kolin Kolistelut/Koli Calling–Third Annual Baltic Conference on Computer Science Education*, 48-56.

[21] Moreno-León, J., Robles, G., Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *Revista de Educación a Distancia*, (46), 1-23.

[22] Aivaloglou, E., Hermans, F., Moreno-León, J., Robles, G. (2017). A dataset of scratch programs: scraped, shaped and scored. *In Proceedings of the 14th international conference on mining software repositories*, 511-514, IEEE Press.

[23] Brennan, K., 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, 1, 25.

[24] Seiter, L., Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *In Proceedings of the ninth annual international ACM conference on International computing education research*, 59-66, ACM.

[25] Wolz, U., Hallberg, C., Taylor, B. (2011). Scrape: A tool for visualizing the code of Scratch programs. *In Poster presented at the 42nd ACM Technical Symposium on Computer Science Education*, Dallas, TX.

[26] Basawapatna, A. R., Repenning, A., Koh, K. H. (2015). Closing the cyberlearning loop: Enabling teachers to formatively assess student programming projects. *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 12-17, ACM.

[27] Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., Franklin, D. (2013). Hairball: Lint-inspired static analysis of scratch projects. *In Proceeding of the 44th ACM technical symposium on Computer science education*, 215-220, ACM.

[28] Kong, S. C. (2019) *Components and Methods of Evaluating Computational Thinking for Fostering Creative Problem-Solvers in Senior Primary School Education.* In: Kong SC., Abelson H. (eds) Computational Thinking Education. Springer, Singapore.