Proceedings

2020 IEEE 44th Annual Computers, Software, and Applications Conference

COMPSAC 2020

Proceedings

2020 IEEE 44th Annual Computers, Software, and Applications Conference

13–17 July 2020
Virtual Event

Editors
W. K. Chan
Bill Claycomb
Hiroki Takakura
Ji-Jiang Yang
Yuuichi Teranishi
Dave Towey
Sergio Segura
Hossain Shahriar
Sorel Reisman
Sheikh Iqbal Ahamed

CPS Conference Publishing Services

Los Alamitos, California
Washington • Tokyo

IEEE computer society

BMS Part Number CFP20061-ART
ISBN 978-1-7281-7303-0

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: <reprints@computer.org>

Editorial production by Lisa O'Conner

IEEE Computer Society
Conference Publishing Services **(CPS)**
http://www.computer.org/cps

2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)

COMPSAC 2020

Table of Contents

COMPSAC 2020 Symposia

ASYS

CAP

CELT

DSAT

EATA

HCSC

ITiP

MOWU

NCIW

SCH

SEPT

SETA

SISA

Student Research Symposium

Fast Abstracts

COMPSAC 2020 Workshops

ADMNET

AIML

AIOT

BDCAA

CCR

CDS

DADA

DBDM

ESAS

ICT4SmartGrid

IEESD

MediComp

MVDA

NETSAP

OER

QUORS

SAPSE

SCA

SDIM

SECE

SESS

SIS-SS

SoNeC

SSML

STA

STPSA

Automatic Formative Assessment in Computer Science: Guidance to Model-Driven Design

Marina Marchisio
University of Torino
Torino, Italy
marina.marchisio@unito.it

Tiziana Margaria
University of Limerick
Lero - The Irish Software Research Centre
Limerick, Ireland
Tiziana.Margaria@ul.ie

Matteo Sacchet
University of Torino
Lero - The Irish Software Research Centre
Torino, Italy and Limerick, Ireland
matteo.sacchet@unito.it
Matteo.Sacchet@ul.ie

***Abstract*—Adaptive online learning can facilitate students' support by responding immediately to the user's interactions. Good feedback to students helps closing the gap between actual and desired performance. In this paper we analyze how to introduce online adaptive formative learning in Computer Science, a discipline with well documented challenges that are hard to tackle with traditional classroom methods. Specifically, we developed illustrative learning items teaching Model-Driven Design and implemented them in an online system that implements a model for automatic formative assessment developed by University of Torino. The model takes advantage of an automatic assessment system initially designed for STEM disciplines, then adopted for teaching languages and other disciplines too. The key features of the adaptive model supported by the online system are algorithmic questions, availability, contextualization, immediate feedback, interactive feedback, and open answers. These features are portable across subject domains, so the system can be adapted to include new subjects. We chose MDD because it is a topic of Computer Science education connected with Computational Thinking, software design, and formal methods, which are three of the core areas in need of enhanced support.**

Keywords — adaptive assessment, automatic assessment, formative assessment, interactive feedback, Computer Science education, computational thinking, model driven design DIME, model checking

I. INTRODUCTION

Modern technology increases the variety of possible different approaches to education. The growing literature on education and technology confirms this trend, with dedicated sections in the announced Encyclopedia of Education and Information Technologies [1]. The potential of Technology Enhanced Learning (TEL), when wisely adopted, is very large. Assessment helps students during their learning process because they can trace their progress and gain motivation or seek help when scoring poorly. Assessment also helps teachers because they can find out whether their teaching is effective and what could be improved. The use of digital technologies, compared to other kinds of technologies, can enhance education even further through interactivity. Teachers need to be trained in all the methods, techniques ad tools, and there is evidence that using effective strategies [2] and the same technology [3] for instructor training can lead to promising results. In the same way, assessment is enhanced by digital technologies, mainly in relation to the capabilities to automatically compute grades and offer real time feedback. Digital technologies give immediate feedback to students and support teachers in grading, promoting the processes of formative assessment. The Information and Communication Technology (ICT) sector is the main responsible for introducing digital technologies, and education in Computer Science (CS) has a special role in this setting.

Many approaches to online education in Computer Science cover the general aspects of content creation and sharing, for example through Open Education Resources [4]. Formative assessment in CS most commonly addresses the problem of evaluating programming skills and code-based artifacts, for example through automatic execution and testing of code, or the provision of hybrid environments that include code in texts (like e.g. the now popular Jupyter notebooks [5]) or viceversa, with a literate programming style. The conceptual aspects of CS education are still largely neglected. The basis chosen for this research is the model of Automatic Formative Assessment (AFA) and interactive feedback for enhancing learning and self-regulation in STEM disciplines developed at the University of Torino [6]. We chose it for its essential features: *Permanent availability, Automatic evaluation, Algorithm-based questions and answers, Open answers, Interactive feedback. Immediate feedback, Contextualization. Portability to different subject areas*.

Given the ease of portability, the natural contextualization consists in its applications to the conceptual topics in CS. We chose three sub-topics of Model-Driven Design (MDD) [7] because with its relation to Computational Thinking, Rapid Application Development, Low-code and No-code Application Development MDD is at the forefront of modern software engineering yet currently not widely taught in curricula. MDD spans theoretical/mathematical knowledge, problem solving and design, formal reasoning about the models, and subsequent tool-based system design and production. In MDD, developers work on various kinds of models rather than directly coding.

In this paper we provide the first instance of the online formative assessment platform for conceptual aspects of CS education that are clearly problematic for many students. It is also the first instance in English, integrated with the Moodle of the University of Limerick. After a brief review of the literature

978-1-7281-7303-0/20/$31.00 ©2020 IEEE
DOI 10.1109/COMPSAC48688.2020.00035

on online formative assessment and its actual state in CS education in Section II, Section III presents the research question and the methodology. Section IV illustrates three sample questions according to the model for automatic formative assessment, while Section V discusses some results.

II. State of the Art

A. Automatic Formative Assessment

Formative assessment defines a wide range of assessment strategies provided by educators while students are learning in order to improve students' results and adapt the learning environment to the needs. Formative assessment uses feedback rather than marks for both student and teacher. The feedback focuses on the details of content and performance. The interpretation of the term *formative assessment* has developed according to different ideas and new available tools, for example the use of digital technologies. In 2009, Paul Black and Dylan Wiliam [8, 9] and the Assessment Reform Group [10] defined:

"*Practice in a classroom is formative to the extent that evidence about student achievement is elicited, interpreted, and used by teachers, learners, or their peers, to make decisions about the next steps in instruction that are likely to be better, or better founded, than the decisions they would have taken in the absence of the evidence that was elicited.*"

The importance of feedback is one of the most distinctive influences on learning and achievement related to formative assessment and thus object of numerous studies. [11] conceptualizes feedback as "information provided by an agent (e.g., teacher, peer, book, parent, self, experience) on aspects of one's performance or understanding". Teachers and the learning system should indicate all these items: the learning goals, the progress toward the goal and the corrections one can look at for greater learning possibilities. Feedback can be effective only when students read it, otherwise there is no way of measuring the quality. In this case, feedback about the self as a person plays an important role, encouraging students to read more about the other pieces of information.

AFA uses adaptive strategies to reach its purposes. Even if there is no standard definition, we will refer to Adaptive Learning as *personalized support strategies in the learning process*. A little bit more in detail, Adaptive Learning is the delivery of custom learning experiences in order to face the unique different needs of an individual learner through feedback, pathways, and resources. It is then assumed that students are aware of how they learn and how they drive their learning. Students are co-designers of the curriculum and of the learning environment. Students' capabilities, interests, needs and skills determine the pace of learning [12]. One of the main strategies with which Adaptive Learning works is in guidance in case of students' failures. In a question when students answer a question incorrectly, the system can gradually guide the student in a discovery procedure that leads to a correct solution, breaking up complex concepts one step at a time [13].

B. Education in Computer Science

CS education is not yet widespread in the education system of many countries and organizations. This situation contrasts the awareness that CS and IT are disciplines that affect everyone in the society. While basic computer use literacy has been promoted for years in the professional field for example through the ECDL [14], too many schools at any level including the third-level (universities) still struggle in many dimensions to introduce CS, in particular what segment of CS and how.

Whether: Many first-year CS students are not sufficiently aware of the specific challenges related to the discipline. For example, surveys of 1st year CS students in a BSc Hons CS program show that only 50% of the students choosing a CS path have previous programming experience. Analyzing data about programs in CS education, [15] inferred that gender is not an ability discriminant, yet there is a gender difference in CS and STEM education. The general Secondary to Tertiary Transition (STT) problem is probably the main cause for the high attrition in year one. Success in the first year of students' career is essential to a positive experience of learners, a central objective both at the institutional and at the national level.

What: concerns the curricular contents. The classic core subjects in CS education are Discrete Mathematics, data structures and algorithms, programming languages, computer organization or computer architecture. In the early days of CS, there was a greater emphasis on lower-level programming and courses dealing with computer hardware, and fewer courses on data structures and software engineering.

How: In [16], the six authors summarize observations and ongoing discussions about online education, OER and resource sharing. With the digital transformation rolling at high speed into the society, it is disheartening that there is still so much disinformation and opposition on the side of education managers and decision makers.

The Computer Science Curricula 2013 of the ACM/IEEE Computer Society [17] are still widely used as a reference, although much has happened since that they do not include. It is not the only Computing curriculum available on the world stage, and various meetings and workshops are devoted to reforming and modernizing aspects of curricula, see e.g, the recent FMFun workshop [18] about formal methods and rigorous modelling practices. However, CS2013 and the Bologna framework approach curriculum not based on "classes" but on the basis of "hours." An hour is meant to be the amount of material covered in an hour of "lecture," although the CS2013 document takes pains to not endorse frontal lecture as the preferred method of pedagogy. Many people think of lively labs and material sharing, with the OER movement [19] producing large scale initiatives like the Merlot platform [20]. So far, there is generally no self-assessment available for prospective and actual students.

Researchers tried to use different tools in order to improve students' learning. In [21], the authors create a new way to introduce high school students to CS whit chatbots that teach basic concepts and provided AFA to the students while using finite state automata and pattern matching. When errors occur, feedback is immediately generated. Other models provide AFA over the whole learning process [22], designing the list of tasks for the students, setting criteria and receiving summative reports about the class progress along the course timeline.

III. RESEARCH QUESTIONS AND METHODOLOGY

Given the many wide experimentations about the use of AFA for STEM disciplines, we wondered if a similar model can be developed and applied to CS.

RQ1. Does the adaptive learning path created to assess and recover students' gaps in CS fulfill the required features of the model developed by University of Torino and by the related literature?

RQ2. What more do learners in CS expect from the formative assessment in order to ease their learning path at the university?

RQ3. Which features of formative assessment are more important for students in CS?

RQ4. How do these questions impact the learner's experience?

According to the original model by the University of Torino, the adaptive questions follow a precise structure. For each question, metadata provide information about Performance, Requirements and Objective. The Performance descriptor defines the standards of accuracy and completeness and activates a reflection on how to use the online materials. This descriptor should be visible, describe a learning outcome and show the conditions under which the student performs the task correctly. The Requirements descriptor states the essential prerequisites to achieve the learning goals of the question. They must be met before attempting to answer in the response area. The Objectives descriptor explains with action-verbs the expected outcomes after the performance.

Students start their assessment through a main question which tests the mastery of a complete active learning process requiring one (or more) answers. Once the main question is answered, there is an adaptive split. If the student responds correctly or sufficiently (e.g. reaches a minimum score) the learning goal is fulfilled, and the question is complete. Otherwise, the student is directed to a remedial path. Starting from the basic knowledge bricks (the prerequisites), the student is guided (and thereby obliged) to reason on a subtopic that is close to his actual knowledge and answer a relative question. At the end of each attempt, both in case of correct and incorrect answer, students receive the correct answer in order to continue the process, and feedback that includes more detailed explanations. The sequence of adaptive split and proposal of a new question (one step further than the previous one) can be applied multiple times and with different approaches, until at the end of the process the entire path answers the main question.

The experimentation so far involved the participation of 6 test individuals (doctoral students and postdoctoral fellows) with different backgrounds (in CS and not in CS), different countries and continents of origin (Germany. Ireland, Argentina, Saudi Arabia, Pakistan, Afghanistan) and of study (also France, South Africa, Estonia). Some tester has teaching experience, others are going to start soon teaching and mentoring. For a better variety of feedback, not all testers are experienced in CS and for most of them it was the first contact with the topic. We used a **Thinking Aloud** technique, test participants were asked to use the system and, at the same time, verbalize their thoughts while moving through the interface and experiencing the appearance, guidance, and feedback. At the end of the test, a **set of questions** was asked about difficulties experienced, feedback, interactivity, and impact of the remedial path on learning through a 5-point Likert scale.

IV. EXAMPLES OF ADAPTIVE QUESTIONS

A. *Model-based construction of an algorithm*

Chain Reaction is a web application that implements a 2 players board game. Students would see it as a first design laboratory (2 days to 1 week, depending on the age and time available) where they program a game strategy using predefined building blocks for operations and decisions. The 2 game players place in turn tokens (atoms) on the board. Every board cell has a (token) capacity equal to the number of neighboring cells (2 to 4). Cells can contain only atoms of one player, indicated by the color (red/blue). When reaching its capacity, the cell "explodes" and spreads its atoms to the surrounding cells, whose atoms take the color of the incoming atom. These cells may reach capacity too and explode, hence the chain reaction effect. When all the atoms have one color, that player wins. The students are asked to build a strategy that the computers follow while facing a human player. This happens by using an MDD environment where the elementary actions and decisions of the "computer" are predefined. Effectively, the students learn this way to design increasingly complex algorithms, in a Problem Based Learning approach to computational thinking that teaches them the basic patterns of workflow composition and programming languages in a programming-free environment, as in [23] and [24].

The example strategy shown during laboratories consists in assigning an initial score 0 to every cell, design an algorithm (criteria and decision mechanisms for changing this score) which computes how good it is to put a token there, and choose for next move the cell with the highest value. The algorithm is designed as a workflow, built as a graph where nodes represent labelled actions and labelled directed edges express the control flow. To build such graph, students must comprehend how it works and what it represents.

The adaptive question provides a snippet of the graph and tests the student's comprehension of both, graph representation and the algorithm. It asks the precise value to be assigned to a specific cell under certain conditions. Students who know the terminology (atom, path, critical cell…) and can read the diagram are able to answer by following the execution path for the specific circumstances (values, cell location and property, other cells' location and properties) of the question instance. The diagram is also provided in the question. When answering incorrectly, students are guided stepwise through the diagram from the "start" until "success". The main response types are numeric, since students must understand the meaning of "values of a cell" (which is equivalent to value of a variable, for a given position in a 2-dimensional array) and need to compute the value of the cell at different points of the diagram (corresponding to different steps of the algorithm). Response areas where students must understand whether the considered cell fulfills certain properties are Boolean, and clearly do not afford multiple attempts. Table I explains the descriptors.

TABLE I. DESCRIPTORS OF THE CHAIN REACTION QUESTION

Descriptor	Chain Reaction question
Performance	Given a process model (graph) of a strategy in the CR game, insert the value (number) of a local process variable under a precise set of initial conditions after evaluating control and data flow of the process model.
Requirements	Know how to read and correctly interpret the graphical elements of a process model (directed graph). Know how to compute a variable's value changes Know basic concepts and terminology of the Chain Reaction game.
Objective	Understand and interpret a process model that contains control and data flow.

B. *Evaluate a simple system-level property on a system model.*

System models can be very complex, and the systems they describe need to satisfy some properties. Properties are typically known to domain experts, and designers often use those properties as functional or non-functional requirements. But how to ensure automatically on a model that it satisfies a set of given properties? A powerful approach is by (algorithmic) Model Checking [25], whose inventors were awarded the prestigious Turing Award in 2007. The first step for students to understand it is to use a simple model and a simple property formulated in natural language. We chose as system a popular model of a microwave oven with seven states, and four elementary (Boolean) properties which are true/false for each state of the oven: "it has *started*", "the door is *open*", "*heat* is on", "there is an *error*". The model is technically a Kripke Transition System (KTS) [25] where states are connected by transitions labelled with actions (like "*open door*", "*start oven*"). After looking at the diagram, students are asked to decide (i.e., evaluate by informal model checking) the property "If there is an error, there is a way to resolve it". For the given model, can the system transition via an action from states where the error property holds to another state where the *"error"* property is false. Beyond an intuition of what is model checking, this question requires knowledge and understanding of KTS diagrams in all their structural and operational elements: states, actions, arrows and their interpretation as state transitions, elementary state properties, and the interpretation of local and non-local properties over a state diagram. Corresponding descriptors are given in Table II.

TABLE II. DESCRIPTORS OF THE MODEL CHECKING QUESTION

Descriptor	Model checking question
Performance	Given a model with states and actions describing the usage of a simple microwave oven, decide if an informal non-local property is true or false for the model.
Requirements	Know states, transitions, actions and atomic propositions of a Kripke Transition System.
Objective	Recognize when a property is true or false in a state Interpret the transition system to evaluate a non-local property.

C. *Understand and interpret a simple CTL formula*

Properties are not written in natural language for model checking, but in temporal logics. Accordingly, students need to be able to properly write and understand temporal logic formulas expressing system properties. The question about Computation Tree Logic (CTL) [26] shows a KTS with a single atomic property ("black") that is true in black states and fails in the white ones. The question asks where a given non-local CTL property holds. Students who fail to answer correctly are guided along the syntactic decomposition of the formula and led to understand its semantics in a bottom-up fashion. To this aim, the students must be familiar with the CTL syntax and the meaning of its operators. The descriptors are explained in Table III.

TABLE III. DESCRIPTORS OF THE COMPUTATION TREE LOGIC QUESTION

Descriptor	Model checking question
Performance	Given a KTS, list the names of the states (in this case, states are numbered) where a given non-local CTL property holds.
Requirements	Know the syntax of the CTL operators. Know the semantics of the CTL operators.
Objective	Understand the abstract model Comprehend how to evaluate bottom-up non-local CTL properties for a KTS.

V. DISCUSSION

The experimentation provided helpful feedback, since it was tested by actual and future instructors with different backgrounds. Table I summarizes the answers to the questionnaire the testers completed after the experimentation.

TABLE IV. EVALUATION ITEMS

Item (1 = a little, 5 = very much)	Average	Median
To which extent did you make use of the multiple attempts provided by the question?	2.8	3
How important do you consider having immediate feedback?	5	5
How deeply did you read what was shown by the immediate feedback?	4.8	5
How much did the interactive feedback allow you to reach the solution?	4.5	4.5
How useful was the remedial path in order to find where your thinking was wrong?	4.5	4.5
How useful do you find the immediate show of the correct answer during the remedial path?	4.4	4
To which extent did the remedial path allow you to understand how to change your answer?	4.75	5
To which extent does the remedial path replace the role of a teacher in explaining a correct step-by-step approach?	3.2	3
How much stimulating did you find the chance to respond with different kinds of response areas?	4.4	4

When setting up the questions with multiple attempts, typographical mistakes or errors due to distraction are easily recoverable. Multiple attempts are not allowed for questions with Boolean answers (true/false, yes/no). Testers suggested to make the initial number of attempts more visible since it is an important information for the students. When asked to which extent they made use of the multiple attempts provided, the average is 2.8 and median 3, indicating that some testers in some parts of the questions used several attempts (Table IV).

Since students are forced to see the feedbacks, this feature leads to an approach of *learning by doing* and *learning by mistakes*, where the students cannot proceed past mistakes without resolving them: the students must stop for a while and focus on why their answer is wrong. One tester liked the idea of guidance a lot, saying that is the same approach he would have taken with his students, noting that this system is "cool" because of its interactivity, its clarity and its completeness. Moreover, since the questions required the analysis of system models, which are essentially graphs depicted in a picture, many feedbacks contained a new picture with new information for the student, showing a relevant part or highlighting a specific element or feature of the model. A tester emphasized that the procedure of gradual (re)discovery inside the question encourages students to build their own learning and understanding of concepts, even when these students give a wrong answer. All the testers agreed on the importance of an immediate feedback (average 5 and median 5), and most of them carefully read the information provided, in order to learn something new even while testing the questions.

Breaking the questions into different and simpler sub-questions leads the students in their own search of the gap, i.e. where the problem in understanding is located. In general, splitting the question into little sub-questions allowed even the testers unfamiliar with the subject matter to reach the full comprehension of the question and of the correct solution, and to understand why their initial answer was incorrect (when applicable). Both cases had an average 4.5 and a median 4.5.

The step-by-step approach is like a personal digital tutor that helps each student throughout the whole process. Teachers cannot afford to conduct such personalized interventions in the classroom with each student, since the relative numbers and short time usually do not allow it. However, the digital tutor embedded in the assessment cannot replace the role of the teacher, which is essential and cannot be excluded from any learning path. The testers agreed on this point, with average 3.2 and median 3, since a good learning path is composed with a balance between the presence of a digital and a "live" tutor. Anyway, the remedial path was very useful to the testers in order to find where they made a mistake (average 4.5 and median 4.5) and how to change the previous answer (average 4.75 and median 5).

Algorithmic questions are an advanced feature of the platform that allows teachers to transform individual questions into a pool of questions covering different cases. This is achieved by enlarging the range of possible question by adding randomization of parameters. This feature can make the repository of individually assessable test questions much larger with little effort. The questions proposed during the experimentation were not "algorithmic" in this sense because they were not yet implemented with random parameters. In the Chain Reaction question this generalization is quite simple, since the learner deals with a value that must be determined according to several decision points corresponding to different paths in the model. By making those parameters algorithmic (for example critical or not, endangered or not, number of initial tokens, etc.), one could easily randomize the choice of the path. Dealing with algorithmic questions requires careful design, because the text elements need to fit the specific parameter instance, which is supported by the platform.

Different types of response areas imply different formats - graphical, numerical, textual - involving all kinds of experience for the student. Thus, different competences can be activated while the learner navigates through the different steps of the remedial procedure. This feature was found stimulating by testers (average 4.4 and median 4).

From an instructional point of view, from this collection of remarks and question scores we infer that most of the relevant features of adaptive questions are appreciated. Students expect surely more than these features, they need to check their knowledge and, when wrong, understand the underlying reason. Immediacy is a key component, and the time to complete the remedial path can be slowed by the incremental stepwise structure of the adaptive question. The remedial path usually searches the exact point in which the student made a mistake by starting from the very beginning and covering all the steps.

VI. CONCLUSION

This research provided useful insights on our prototyping Automatic Formative Assessment (AFA) in Computer Science (CS). The original model by University of Torino looks easy to adapt to new subject area settings. According to the feedback provided by the small-scale experimentation, recalling the research questions of Section III, we remark that:

RQ1. The adaptive learning path containing the remedial procedure created to assess and recover students' gaps in CS does indeed fulfill the features of the University of Torino model and the related literature. The features were a guideline in designing the three learning paths. We adopted all but the algorithmic one, but we see applications also for this one in other areas of CS.

RQ2. Learners are facilitated in their comprehension path. Since one layer of knowledge is built on the previous ones, it is important that, whenever an answer is wrong there is a tool that can help unwind the thread of connected knowledge and fill the gap.

RQ3. The adaptive question's features are all very important, but the top feature according to the results is the immediacy of feedback, which is provided through a direct interaction with the learner. Thus, posing the online question requires good communication skills too, in order to reach its full explanatory and experimental potential.

RQ4. The action on learners' experience plays a key role. In the next phase more experimentation will be carried on with cohorts of enrolled students.

Future work concerns the development and delivery of adaptive assessments in an open online learning environment. First year university students need all the adaptive support that online courses can provide, an example for Mathematics is provided in [27]. Considering the most up to date trends in education (openness, mobile learning, adaptive learning, etc.), in a newly acquired project at our institution aimed at providing individual orientation to prospective 1st year students the platform contents will help prospective students to consciously make their choice about an academic career in CS. This way, prospective students will for the first time test their skills and competences in advance of their application, while evaluating different possible career paths. They will be able to try the tests multiple times. The tests are automatically evaluated and the students receive immediately automatic feedback. In a second use of the platform, enrolled 1st year students will be able to attend online paths for core concepts and skills in order to remedy knowledge gaps more deeply and get more support to learn the terminology, as precision of expression is one of the identified hurdles. This new opportunity will also directly affect students that have difficulties in attending classes, like working students or students with limited mobility. Data collection analysis and learning analytics techniques [28] applied to the fine granular data obtained from the online platform is likely to provide new insights in how the students learn on their own time, thus helping us to improve teaching and learning in CS.

ACKNOWLEDGEMENTS

This research was funded in part under grant Science Foundation Ireland grant 13/RC/2094 to Lero - the Irish Software Research Centre (www.lero.ie), HEA/NF SATLE 2019 Funding from the Irish National Forum for the Enhancement of Teaching and Learning in Higher Education, and under the grant "Bando per l'internazionalizzazione della ricerca - Anno 2018" from the bank foundation Compagnia di San Paolo.

REFERENCES

[1] A. Tatnall, Encyclopedia of Education and Information Technologies, Springer International Publishing, 2020.

[2] M. Marchisio, M. Sacchet, D. Salusso, "Instructional design to "train the trainers": The start@unito project at the university of turin ", Multi Conference on Computer Science and Information Systems, MCCSIS 2019 - Proceedings of the International Conference on e-Learning 2019, pp. 195-202, 2019. doi:10.33965/el2019_201909F025.

[3] A. Brancaccio, M. Esposito, M. Marchisio, C. Pardini, M. Sacchet, "Open professional development of math teachers through an online course", Multi Conference on Computer Science and Information Systems, MCCSIS 2019 - Proceedings of the International Conference on e-Learning 2019, pp. 131-138, 2019. doi:10.33965/el2019_201909F017.

[4] H. Chan, C. Demartini, T. Margaria, E. Tovar, D. Towey, S. Reisman "OER: Six Perspectives on Global Misconceptions and Challenges", in Proc. IEEE Int. Conf. on Teaching, Assessment, and Learning for Engineering (TALE 2019), pp. 889-895, 2019.

[5] In Project Jupyter, retrieved from https://jupyter.org/. Last accessed February 17th, 2020.

[6] A. Barana, A. Conte, M. Fioravera, M. Marchisio, S. Rabellino, "A Model of Formative Automatic Assessment and Interactive Feedback for STEM", Proceedings of 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), IEEE, Tokyo, Japan, pp. 1016–1025, 2018.

[7] A. L. Lamprecht, T. Margaria, "Modeling of Scientific Workflows". In: Encyclopedia of Education and Information Technologies. Springer, Cham, 2019.

[8] P. J. Black, D. Wiliam, "Developing the theory of formative assessment," Educational Assessment, Evaluation and Accountability, n. 21, pp. 5–31, 2009.

[9] P. J. Black, D. Wiliam, Inside the black box: raising standards through classroom assessment. London: King's College London School of Education, 1998.

[10] In Assessment Reform Group web-site, retrieved from https://www.nuffieldfoundation.org/project/the-assessment-reform-group/. Last accessed February 17th, 2020.

[11] J. Hattie, H. Timperley, "The Power of Feedback," Review of Educational Research, vol. 77, n. 81, 2007.

[12] D. Nandigam, S. Sremath Tirumala, N. Baghaei. "Personalized Learning: Current Status and Potential". IC3e 2014 - 2014 IEEE Conference on e-Learning, e-Management and e-Services, 2014. doi:10.1109/IC3e.2014.7081251.

[13] A. Barana, M. Marchisio, M. Sacchet, "Advantages of Using Automatic Formative Assessment for Learning Mathematics". In: Draaijer S., Joosten-ten Brinke D., Ras E. (eds) Technology Enhanced Assessment. TEA 2018. Communications in Computer and Information Science, vol 1014. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-25264-

[14] In ECDL Foundation - ICDL - International Computer Driving Licence, retrieved from http://ecdl.org/. Last accessed February 17th, 2020.

[15] C. McInerney, A. L. Lamprecht, T. Margaria, "Computing Camps for Girls – A First-Time Experience at the University of Limerick". In Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing, Springer V., pp. 494–505, 2017.

[16] J. Voas, R. Kuhn, C. Paulsen, K. Schaffer, "Computer Science Education in 2018," in IT Professional, vol. 20, no. 1, pp. 9-14, January/February 2018. doi: 10.1109/MITP.2018.011021350

[17] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society, "Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science", Association for Computing Machinery, New York, NY, USA, 2013. https://doi.org/10.1145/2534860

[18] In FMFun Website, retrieved from https://fmfun.github.io/. Last Accessed February 18th, 2020.

[19] In Forum on the Impact of Open Courseware for Higher Education in Developing Countries, UNESCO, 2002: final report - UNESCO Digital Library, from https://unesdoc.unesco.org/ark:/48223/pf0000128515. Last accessed February 18th, 2020.

[20] In MERLOT, retrieved from https://www.merlot.org/merlot/. Last Accessed February 18th, 2020.

[21] L. Benotti, M. C. Martnez, F. Schapachnik, "A Tool for Introducing Computer Science with Automatic Formative Assessment," in IEEE Transactions on Learning Technologies, vol. 11, no. 2, pp. 179-192, 1 April-June 2018. doi: 10.1109/TLT.2017.2682084

[22] R. Rashkovits, L. Ilana, "FACT: A Formative Assessment Criteria Tool for the Assessment of Students' Programming Tasks.", Proceedings of the World Congress on Engineering WCE 2013, London, U.K., pp. 384-389, July 3-5, 2013.

[23] F Gossen, D Kühn, T Margaria, AL Lamprecht, "Computational thinking: learning by doing with the Cinco adventure game tool", 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, 2018, pp. 990-999, doi: 10.1109/COMPSAC.2018.00175.

[24] T. Margaria, "From computational thinking to constructive design with simple models." In ISoLA 2018, Lecture Notes in Computer Science, vol 11244. pp. 261-278. Springer, Cham, 2018.

[25] M. Müller-Olm, D. Schmidt, B. Steffen, "Model-Checking". In: Static Analysis. SAS 1999. Lecture Notes in Computer Science, vol 1694. Springer, Berlin, Heidelberg, 1999.

[26] E. A. Emerson, "Temporal and modal logic". In Handbook of Theoretical Computer Science, vol. B. MIT Press, pp. 955–1072, 1990.

[27] M. Marchisio, S. Remogna, F. Roman and M. Sacchet, "Teaching Mathematics in Scientific Bachelor Degrees Using a Blended Approach", IEEE 44th Annual Computer Software and Applications Conference (COMPSAC), 2020, in press.

[28] M. Marchisio, S. Rabellino, F. Roman, M. Sacchet, D. Salusso, "Boosting up Data Collection and Analysis to Learning Analytics in Open Online Contexts: an assessment Methodology". Journal of E-Learning and Knowledge Society, 15(3), pp. 49-59, 2019. https://doi.org/10.20368/1971-8829/1135048

[29] A. Barana, A., Conte, C. Fissore, M., Marchisio, S., Rabellino, "Learning Analytics to improve Formative Assessment strategies". Journal of E-Learning and Knowledge Society, 15(3), 75-88, 2019. https://doi.org/10.20368/1971-8829/1135057

Author Index