978-84-09-00829-2

Next Generation Conversation-Based Learning Tool for Learning to Code

Enric Mor
Computer Science, Multimedia and Telecommunication Studies
Universitat Oberta de Catalunya
Barcelona, Spain
emor@uoc.edu

Francesc Santanach
eLearn Center
Universitat Oberta de Catalunya
Barcelona, Spain
fsantanach@uoc.edu

Susanna Tesconi
Computer Science, Multimedia and Telecommunication Studies
Universitat Oberta de Catalunya
Barcelona, Spain
stescone@uoc.edu

Carlos Casado
Computer Science, Multimedia and Telecommunication Studies
Universitat Oberta de Catalunya
Barcelona, Spain
ccasadom@uoc.edu

***Abstract*—Learning to code can be considered as one of the most representative examples of learning by doing. Programming is a skill and learning skills requires practice. Online teaching and learning depend on tools that allow students practice and teachers to monitor student progress and provide feedback. In this work, the design and architecture of an educational tool for learning to code are presented. The CodeLab tool is based on skill practice and assessment and is targeted for non-STEM students to develop computational thinking and programming competences. The tool is designed to provide an educational lab experience based on activities to practice through a conversational interface.**

Keywords—Learning to Code, Learning Labs, Learning Tools, Virtual Learning Environments, Conversation-Based Interaction, Dialogue-Based Interfaces, Interaction Design.

I. Introduction

An educational lab is a multidisciplinary space where knowledge is built through to social interaction, self-management, self-training, informal research and learning. The laboratory is the place for experimentation for students, and where peer training processes are activated. It is the quintessential space to develop creativity, test ideas, learn from mistakes and the acquisition of learning by practicing skills and learning by doing [1].

Educational labs are one of the most suitable environments for learning to code since programming can only be learned by programming [2]. Learning to code in a laboratory let students to build knowledge through practice and social interaction since it allows teachers to create scaffolding strategies to support the acquisition and practice of computational thinking [3]. In online, virtual and asynchronous learning settings, the creation of scaffolding strategies to support learning to code constitute a challenge since it is needed to re-enact the complexity of social and cognitive interactions that take place in educational labs.

Technology Enhanced Learning (TEL) has long been developing learning tools and environment to improve teaching and learning. Also, the application of artificial intelligence in education has lent to specific research areas such as Intelligent Tutoring Systems (ITS) and autonomous agents [5]. Recently, conversational interfaces and systems have experienced a large development. Chatbots and conversation based assistants can be found in the main operating systems, smart speakers or online shopping sites. Additionally, chatbots can also be found in chat-based environments as Slack (slack.com), Facebook Messenger (messenger.com), Skype (skype.com), Google Hangouts (hangouts.google.com) or Telegram (telegram.org). Current research on TEL, ITS and agents for education suggest a big impact on teaching and learning experiences. In educational settings, chatbots can provide tutoring, student guidance, question answering, coached-practice, assessment and teacher support [6].

II. CodeLab Learning Tool

A. Learning to Code

CodeLab is learning tool with the goal to provide students a learning code laboratory tool that mimics the interactions that happen in a face-to-face traditional programming laboratory. The tool provides a chat-based communication interface for students and instructors. This interface is also used by a set of chatbots that provide support to the programming language, suggests a series of exercises and activities that the students have to solve or complete and help students to solve questions and monitor their progress. Also, there will be a chatbot that when identifies learners experiencing difficulties, it will proactively suggest practice and will provide help to solve the proposed activities. CodeLab is an ongoing project that takes place at UOC, Universitat Oberta de Catalunya (http://www.uoc.edu), a fully online university based in Barcelona, Spain that offers a wide range of programs and courses. The programming courses at UOC are fully online and are built around: 1) learning resources; 2) interaction with teachers and peers; 3) code activities. Online teaching

and learning to code presents several issues for students and for instructors: 1) installing and IDE (Integrated Development Environment) can be difficult for learners; 2) in a learning context, collaborative coding activities can be tough because they require to interchange code by shared folders or by email or to use version control tools (that can be also tough to learn for novice students); 3) the instructor has to download, classify, review and execute the code of each the activities of each learner; 4) it is not easy for the instructor to track each student learning process, review all the activities or obtain information about students practice. Codelab aims to develop a new approach to deal with these issues, moving to an educational lab approach and rethinking students learning experience and instructors teaching experience by means of a practice-oriented learning environment with a conversation-based interface.

The approach for the CodeLab learning tool is on learning to code in an online laboratory where students learn by doing: they explore how to solve problems, complete activities can discuss solutions and problems with peers or with teachers. The research also considers the design of the most suitable interaction style and interface for the tool. Technology enhanced learning, human computer-interaction and ethnography are the research fields involved.

Learning to program is not easy [7, 8] and it requires a lot of practice. Taking this into account, we foresee a learning environment that is able to trigger the understanding of programming concepts (knowledge) as well as the acquisition of programming skills. Learning skills require continuous practice and also reflecting about the practice. The learning process within the tool is simple: first, students are provided with some basic theoretical concepts. Once the basic concepts are reviewed, they are invited to practice as much as possible as a way to experience and explore concepts, problems and solutions. Once the theoretical concepts are practiced, students can move on to more advanced ones. This process is not easy to develop since it requires the teacher to review each student practice and decide if they have acquired the skills and knowledge needed before progressing with new ones. There are other educational elements to take into account when designing the student learning process and experience with the tool like, for example, the student learning pace. In distance learning, in order to succeed, students have to self-organize and take full responsibility over their learning. That means that each one determines the time dedicated to study and when and how to do it. This can add complexity to teachers when designing learning itineraries and providing support and feedback to students, especially in practice-based learning scenarios. Taking this into account, there is a need for learning tools where students can develop practice, receive feedback and teachers can easily provide support and monitor their progress.

B. CodeLab

The main features of the CodeLab learning tool are: 1) code can be written and its execution can be visualized; 2) activities for practice and learn are presented in a consistent and logic order; 3) students can solve, visualize execute and save activities in their profiles; 4) student code practice and progress solving activities can be monitored; 5) teachers can follow up on the student's activity and performance and, if necessary, step in and provide recommendations or feedback; 6) a dialogue-based interface is provided for participants (learners and teachers) to interact, answer questions and share comments; 7) chatbots are available in the conversational interface ready to offer answers to questions about activities or helping in specific aspects of algorithmic learning or programming language.

There are currently several programming tools and environments to write code and view its execution in real time. Some of them are used in educational settings to learn and train people to code. However, most of these tools are designed as coding tools but not as learning tools since do not provide learning itineraries, support, assessment or feedback. CodeLab is being designed with the goal to provide each learner a set of activities to complete in the tool. The activities are designed to encourage experimentation and explore theoretical concepts, when needed, in order to develop programming skills on learners. Activities are presented in sequences that are related to previously defined learning itineraries. Practice-based learning itineraries are designed by teachers, taking into account the skills that learners need to develop on each stage. Learning itineraries can have variants depending on the skills to develop and the progression that a student takes and have diverse types of activities. Template-based activities allow CodeLab to provide different activities to each learner where the same skills and concepts are put in practice.

Assessment is one of the goals of CodeLab as a learning tool. We want to be able to provide formative assessment based on learner continuous practice. To do so, CodeLab will be able to track the activities solved and the practice carried out by each learner, capturing the time spent and the results obtained. This will allow the tool to make recommendations based on the learning itineraries and will allow teachers to provide significant feedback based on learner progress. Monitoring learner progress makes possible summative and formative assessments. For summative assessments, assignments can be submitted within the tool and become a one more item of student profile. On the other hand, there is a need to provide formative assessment, specially suitable in subject areas as programming, where learning is mostly based on practice. A set of activities carried out through the CodeLab in a given time-period can allow teachers to provide feedback and, if necessary, a mark.

Feedback is an essential element of assessment and learning. The analysis of the information collected about learners (activities solved, code executed, interaction with the tool, etc) will allow Codelab to offer feedback related with the learning itineraries, reinforcing areas where students need to improve and providing recommendations about activities to be solved and theory concepts to be reviewed. Therefore, the feedback takes an intelligent tutoring system approach but will take advantage of the conversational interface of the learning tool.

One of the big educational and design challenges of the CodeLab learning tool is the design of the chat-based

interface that will allow conversations to take place in context, that is, where learners practice, explore and experiment with code. The goal is not only to allow but to promote the discussions that take place in a traditional face-to-face learning laboratory. Discussions in context are very important when learning skills. For example, when a learner has a question about a piece of code, she ask the teacher or discuss with a colleague at the same time that she points a specific part of the code. In addition to this, the CodeLab project explores ways of providing learning and assessment support by means of educational chatbots.

C. *Chatbots and conversational UI*

The interaction and user experience aspects of educational chatbots are another design and research challenges of CodeLab. Ethnographic studies [9] of current both face-to-face and digital educational labs are taken into account. Also, the current research on chatbots from Artificial Intelligence (AI) and Human-Computer Interaction (HCI) are considered [6]. A set of educational features that bots can provide have been identified, from providing help with the programming language syntax to providing feedback and recommendations based on activities completed and theory accessed by learners. For example, a programming language bot can be asked about functions (reusable blocks of code), its arguments or the syntax. The chatbot can provide answers based on the programming language reference and help systems. Other example of chatbot for codelab is the recommender bot that provides feedback based on the activities carried out by students. If a learner complete a set of activities of a learning itinerary, the bot can suggest to move to the next level and start a new itinerary. If a learner is stuck with some key activities in an itinerary, the chatbot can suggest to read some theoretical concepts and complete similar but easier activities. Chatbots will use the dialogue-based interface of the CodeLab tool, sharing the conversational space of the learning lab with learners and teachers.

III. ARCHITECTURE

CodeLab learning tool needs to be integrated in a virtual learning environment. Therefore, the conversational interface of CodeLab needs to provide an integrated interaction experience with the different elements of this virtual learning environment. The challenge here is to design and deploy the chatbot conversation-based interaction into the main application contexts of the virtual learning environment. The three main contexts are: digital learning resources context; virtual classroom context; and conversational interface tool context.

In the digital learning resources context, the chatbot is deployed inside the learning materials, acting as a learner assistant, providing, depending on the topic or objective of each material section, the most appropriate exercises, activities and assessments. In the virtual classroom context, the chatbot is deployed inside the virtual classroom, it acts as an assistant, providing, through the chat interface, access to the resources and assessments available in the classroom. In the conversational interface tool context, the chatbot is deployed using the Slack tool. Slack is a conversational tool and in that case, unifies all learning and teaching communications in one place. In CodeLab, Slack brings all code learners together in the same place, there are channels for each programming language and for promoting the community engagement. Here, the slackbot acts as an automatic lab assistant, solving the students' code questions, providing the appropriate lab resources for solving activities, assessments and transferring to the real lab instructor those questions that it doesn't understand or needs to be answered by a real human teacher.

The chatbot is also an applications/systems aggregator. From the logical point of view, we can distinguish the following modules: a) workspace for code practice; b) assessment management system; c) grading & feedback; d) dashboards & analytics; and e) the chatbot itself.

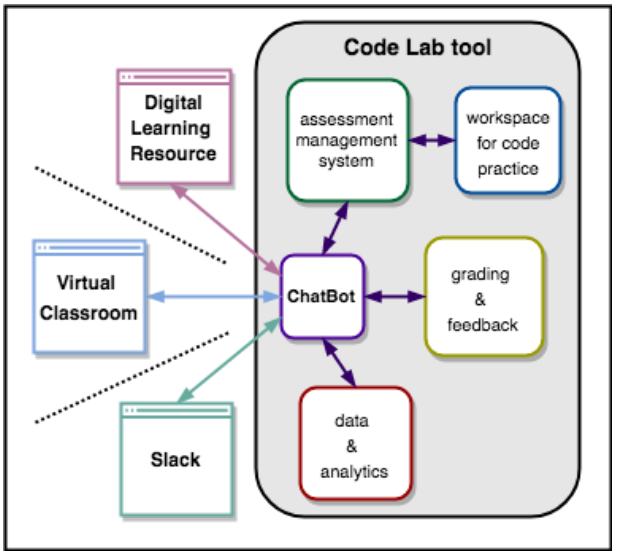

Figure 1: *CodeLab.*

The workspace for code practice (a) is essentially a cloud-base IDE for learning programming. In its first iteration, it provides P5.js as a programming language to learn. Since it is based on JavaScript, it does not require a server to compile and execute the code but can be directly interpreted and executed in any web browser.

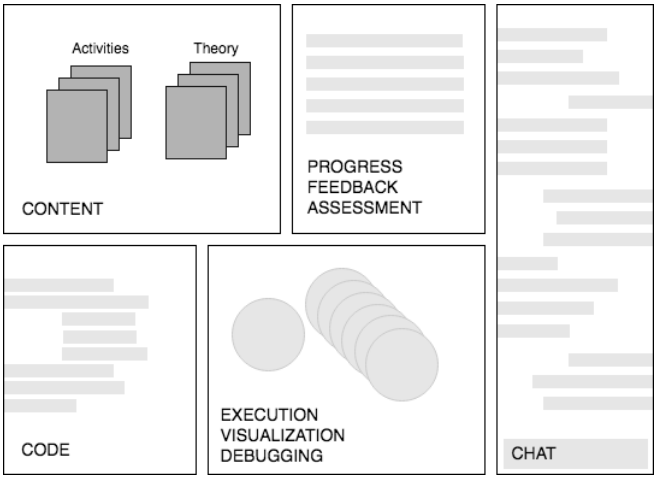

Figure 2. *Workspace for code practice*.

The Assessment management system (b), is a system for managing the authoring and the deployment of the code exercises into the workspace for code. Through this module, is possible to connect with the authoring tools and to set up

specific learning activities for both, to be performed by students in specific time and to catalog for subsequent reuse.

The grading & feedback (c), is a tool for grading and providing feedback to students. This is a complex issue, due to the complexity of the grading code. Rubrics in this context could be very diverse such as the code style, the test cases, the legibility, the accuracy, the performance, and others.

The data & analytics module (d), includes interfaces and services related to the data gathered by the chatbot and by the other tools and modules in the ecosystem. It allows the chatbot provide recommendations to learners and student progress dashboards to instructors.

The chatbot (e) is a conversational-based engine that interacts with students and instructors. For instructors, the chatbot provides access to modules for the authoring, delivering, grading and monitoring processes. For students, it provides assistance in context, recommending the most appropriate exercises to them.

The chatbot, as a students assistant, provides 24x7 assistance to students; listens to the student-instructor interaction with the goal to solve those questions that it can understand and answer correctly; introduces itself to the student as a bot and requests confirmation of comprehension; in all other contexts, it forwards the message to the instructor and informs the student.

In the other hand, as a instructors assistant, the chatbot offers automatic answers to common, well known and identified questions; it never takes the instructor's identity; identifies itself as a bot at all times; sets an interface that enables to enhance answers and learn new ones; and allows instructors to focus on the teaching process and improve their job.

IV. Discussion

The goal of the CodeLab project is to provide a different approach to online learning of programming. We want to to solve some of the common learning problems and provide an improved learning experience through learning by doing in practice-based laboratories and conversation-based interactions. We pursue two main goals: defining the architecture and design for the learning tool and designing the bot interaction that provides support to the tool. In online educational settings, the way students communicate with other students and with instructors is a key element of it success. Chatbots in education can be challenging but also provide interesting opportunities for to guide and support students and learning. Developing chatbots in virtual learning environments entail the redesign of several modules of the LMS in order to introduce conversational interaction.

We are in the process of setting up a first version of the CodeLab learning tool, based on the design and architecture presented in this paper. The tool will be used and tested in a P5.js course with arts and design students. Students from non-STEM degrees usually need more support and guidance when learning to code. We expect the CodeLab tool can provide it. Additionally, from this first experience, we expect to get rich data to iterate the design and implementation of the learning tool and the chatbot, being able to offer better conversation and guidance.

References

[1] Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. International Journal of Educational Technology in Higher Education, 14(1), 42. doi: 10.1186/s41239-017-0080-z

[2] Hassinen, M., & Mäyrä, H. (2006, February). Learning programming by programming: a case study. In Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006 (pp. 117-119). ACM doi: 10.1145/1315803.1315824

[3] Van Merrienboer, J. J., & Paas, F. G. (1990). Automation and schema acquisition in learning elementary computer programming: Implications for the design of practice. Computers in Human Behavior, 6(3), 273-289. doi: 10.1016/0747-5632(90)90023-A

[4] Veletsianos, G., Heller, R., Overmyer, S., & Procter, M. (2010). Conversational agents in virtual worlds: Bridging disciplines. British Journal of Educational Technology, 41(1), 123-140. doi: 10.1111/j.1467-8535.2009.01027.x

[5] Roll, I., & Wylie, R. (2016). Evolution and revolution in artificial intelligence in education. International Journal of Artificial Intelligence in Education, 26(2), 582-599. doi: 10.1007/s40593-016-0110-3

[6] Song, D., Oh, E. Y., & Rice, M. (2017). Interacting with a conversational agent system for educational purposes in online courses. In Human System Interactions (HSI), 2017 10th International Conference on (pp. 78-82). IEEE. doi: 10.1109/HSI.2017.8005002

[7] Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. In International Conference on Engineering Education–ICEE (Vol. 2007).

[8] Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. In Acm Sigcse Bulletin (Vol. 37, No. 3, pp. 14-18). ACM. doi: 10.1145/1067445.1067453

[9] Fink, R. D., & Weyer, J. (2014). Interaction of human actors and non-human agents. A sociological simulation model of hybrid systems. Science, Technology & Innovation Studies, 10(1), 47-64.