

Learning Introductory Programming with Smart Learning Environment

by Boban Vesin (University of South-Eastern Norway), Katerina Mangaroska and Michail Giannakos (NTNU)

Programming Tutoring System (ProTuS) is an adaptive learning system developed to support introductory programming. ProTuS utilises a cross-platform architecture that aggregates and harmonises learner analytics coming from different systems and quantifies learners' performance through a set of indicators. ProTuS has been successfully used within universities to support teaching and learning.

Programming tutoring system

ProTuS [L1] was developed in the Learner-Computer Interaction lab [L2] at the Norwegian University of Science and Technology (NTNU), in cooperation with the University of South-Eastern Norway (USN). The system represents an effort to develop an adaptive and interactive environment that provides personalised learning to students (Figure 1) [1]. ProTuS covers a complex interplay of learning resources, tasks, instructions, social dynamics, interactions, and learning analytics aimed at helping students to learn introductory programming [2]. The system has been utilised at several universities for over a decade [3].

The current version of ProTuS includes interactive learning resources from various content providers, developed at several universities from Norway, USA and Canada. In addition, the system offers adaptive features based on content recommendation, automatic assessment and grading.

Interactive learning content

The interactive learning content (i.e., resources) included in the system comprises four types of activity that support individual work. Students can practice and learn programming through the following learning resources [2]:

Explanations. ProTuS contains reading content (i.e., tutorials) on 15 topics that

are aligned with the curriculum presented in the introduction to Java course. The objective behind the reading content is to allow students to leverage on their background knowledge on procedural programming (normally taught via Python) and progress with object-oriented programming (OOP). To do so, the content provides a comparison of the syntax and the basic concepts in Python (procedural) and Java (OOP).

Interactive examples. For each of the proposed topics, the students have the chance to explore different examples, called Program Construction EXamples (PCEX). Each PCEX content starts with

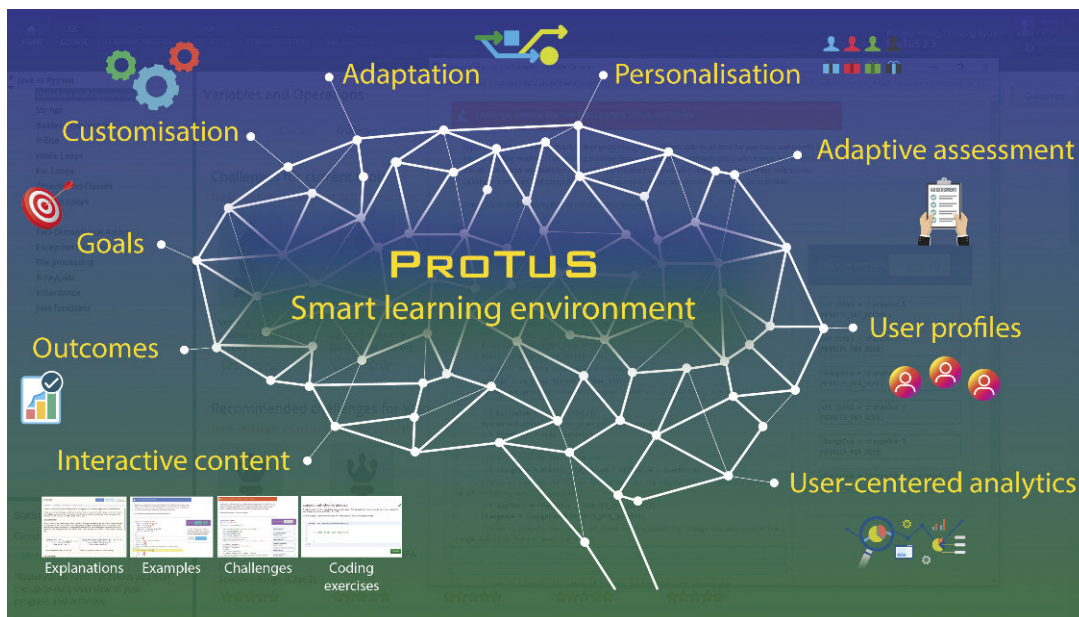


Figure 1:
Personalised learning
in ProTuS.

a worked-out example with explanation of how to write a code for a particular problem. The explanations (that are hidden until a student clicks on the lines of interest) are available for almost all lines in the example and they focus on why students need to write a code in a certain way or why certain programming constructs are used in the code.

Interactive challenges. To allow a seamless connection between theory and practice, as well as give students the possibility to try out the given examples, we present a challenge after each example. This allows students to solve one or more challenges related to the example that they previously viewed.

Coding exercises. This is a type of smart content that requires students to write code or complete a given code to achieve a certain goal. Each coding exercise has a problem description and a baseline code. When students submit their code, it is tested against a set of predefined unit tests and the student receives automated feedback on whether the tests were passed or not.

Interactive examples and challenges (Mastery Grids, PSEX) were developed at the University of Pittsburgh, and coding exercises (PCRS) at the University of Toronto [2].

Scaling up research efforts to support introductory programming

As a cross-platform architecture, ProTuS aggregates and harmonises learner-generated data from several sys-

tems into a set of meaningful indicators. Thus, it acts as a learning eco-system that leverages on various learning analytics to enhance personalised and adaptive learning. To address barriers hindering the usefulness and efficiency of an adaptive learning system, we examined ProTuS usability and evaluated its cross-platform learning analytics capacities to support personalised and adaptive learning [1]. The system has been tested in different universities throughout Norway and other European countries, with the results indicating that it shows promise for supporting introductory programming.

In the future, the authors plan to further develop the learning analytics component of ProTuS and provide more learner-centred visualisations and cross-analytics (e.g., data coming from assignments, project-work, course grading, and third-party software) [2]. Furthermore, the overall goal is to investigate how analytics derived from various sources can help us to construct efficient teaching strategies and support online and blended teaching and learning of introductory programming.

This project is a product of a research collaboration between NTNU and USN, receiving funding from Norwegian Agency for International Cooperation and Quality Enhancement in Higher Education (DIG-P44/2019), the Norwegian Research Council (255129/H20) and NTNU's Excellent Education program (NTNU Toppundervisning).

Links:

[L1] <https://protus.idi.ntnu.no/>

[L2] <https://lci.idi.ntnu.no/>

References:

- [1] B. Vesin, K. Mangaroska, M. Giannakos: "Learning in smart environments: user-centered design and analytics of an adaptive learning system," Smart Learn. Environ.
- [2] K. Mangaroska, B. Vesin, and M. Giannakos, "Cross-platform analytics: A step towards personalization and adaptation in education," LAK
- [3] A. Klačnja-Milićević, B. Vesin, and M. Ivanović, "Social tagging strategy for enhancing e-learning experience," C&E

Please contact:

Boban Vesin
University of South-Eastern Norway,
Norway
boban.vesin@usn.no