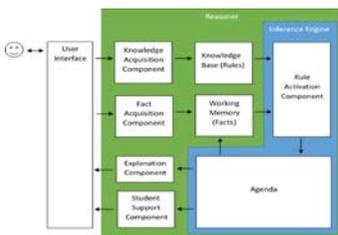
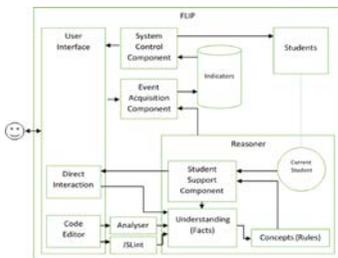


Research Student
 Sokratis Karkalas

Supervisors
 Sergio Gutiérrez Santos
 Keith Mannock



FLIP Learning (Flexible, Intelligent and Personalised Learning)

Research Aims

FLIP is an Exploratory Learning Environment (ELE) used for teaching introductory programming to University students. Teaching and learning programming at an introductory level is particularly challenging for tutors and students. Teaching requires a major effort in terms of design, preparation and implementation from academic staff. Learning takes place in computer laboratories, where students under the supervision of tutors, attempt to solve programming exercises. It is self-evident that there is an analogy between the effectiveness of the processes that take place during the practical exercises in the lab and the actual learning outcome that can eventually be achieved. If students can utilise as much as possible of the resources available and the system is flexible enough so that individualised support is provided in a timely fashion then it is more likely for them to achieve an optimum result.

Existing tutoring systems provide assistance to well-defined Problem-Based Learning (PBL) scenarios in a relatively controllable way. Help in FLIP is not provided in an intrusive way and can be task-independent. Support is always available but is only given on demand. Students are given Inquiry-Based Learning (IBL) scenarios which by definition are open-ended ill-defined problems and try to discover knowledge in an exploratory manner.

Research Approach

FLIP utilises a rule-based expert system that interacts with students when there is a misconception to be resolved. It takes the role of the teacher and repetitively exchanges information with the student in order to assess the current situation, identify the problems and provide individualised support whenever possible.

This system accepts two inputs: rules (misconceptions) and facts (current student understanding). Rules are inserted by experts and form the knowledge base of the system. Rules correspond to known student misconceptions identified in computer laboratories. The conditional part of these rules corresponds to one or more characteristics identified in the code. Facts are inserted dynamically into the system during the development process. The student's code is statically analysed and divided into distinguishable characteristics that depict the student's current understanding and thus play the role of the facts.

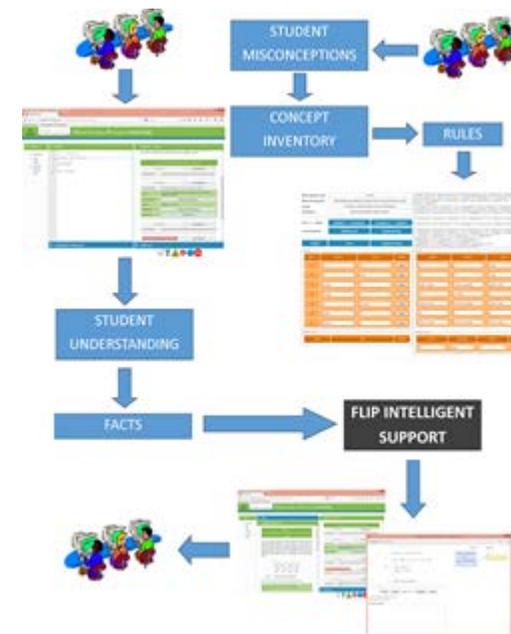


Figure 1. Elicitation of students' misconceptions, construction of concept inventory in the form of a rules in a knowledge base, static analysis of code and use of rules to provide task-independent personalized support

Selected Publications

1. Karkalas, S., & Gutierrez-Santos, S. (2014). Intelligent Student Support in the FLIP Learning System based on Student Initial Misconceptions and Student Modelling. In *International Conference on Knowledge Engineering and Ontology Development*.
2. Karkalas, S., & Gutierrez-Santos, S. (2014). Enhanced JavaScript Learning using Code Quality Tools and a Rule-based System in the FLIP Exploratory Learning Environment. In *IEEE International Conference on Advanced Learning Technologies*.
3. Karkalas, S., & Gutierrez-Santos, S. (2014). Enhance Teaching and Learning of Computer Programming in Exploratory Learning Environments using Intelligent Support. In *IEEE International Conference on Advanced Learning Technologies*.
4. Karkalas, S., & Gutierrez-Santos, S. (2014). Eclipse Student (In) Activity Detection Tool. In *Open Learning and Teaching in Educational Communities* (pp. 572-573). Springer International Publishing.