Chapter 9

Intelligent Tutoring System for Learning Programming

Tonći Dadić

University of Split, Croatia

ABSTRACT

Tutoring systems for programming help students understand features of target programming languages, and develop their general problem solving skills. Our system guides novices in learning syntax and semantics of programming language, problem decomposition, program design and testing. It is language independent, automatically adjusting course material to the language - subject of learning. Semantic analysis of student’s programs is based on comparison of abstract semantics trees of student’s and model-program. Model-program is coded by innovative programming language describing goals and variants of possible solutions. The notional machine defined by programming language, its verbal description of instruction actions and visualization of program execution, helps students to understand semantics of teaching language. Advancement through the course material controlled by computer teacher supports connection of new concepts to the present student’s knowledge. The system applies distinct teaching strategies to stoppers and movers.

INTRODUCTION

Intelligent tutoring systems for programming are an interesting area of artificial intelligence application. Since early seventies, tens of systems have been designed and hundreds of papers published, related to procedural (Adam 1980, Barr 1976, Johnson 1984, Ueno 1994), functional (Anderson 1985), logical and object languages (Butz 2006, Chee 1993), including problem analysis, program specification, planning (Eisenberg 2003) and coding. In addition, researchers focused to adaptive navigation (Brusilovsky 1992), visualization of program execution and intelligent debugging (Eisenberg 2003).

Regardless significant research effort in field, there are very few intelligent tutoring systems or integrated learning environments used regularly.
for teaching programming in a real classroom, and most of them can not be used outside the university where they were created (Brusilovsky 1995). Author explains such situation by complexity of intelligent systems, where research and development of each component requires a couple of man-years. Design of an environment supporting efficient teaching of programming is a challenging domain with many open issues to explore (Guzdial 2004).

In this chapter we present our system for teaching to program by procedural languages. We based design of our system to fundamentals referred in section Learning programming and tutoring systems. Many attributes of our system are typical, while we believe that some characteristics make it unique in the field. Section History of intelligent tutoring systems for programming depicts history of the field through characteristic of some specific systems, leaving reader to compare our work. We explain our system in necessary details. In The system evaluation section we present findings of our students in using the system in real classroom.

BACKGROUND

We believe that students have to acquire computer programming knowledge, skills and capabilities already in primary education. Educational contents that would enable this process, starts with problem solving formalism (in individual and group work), and proceed with programming paradigm using some appropriate programming languages as LOGO, Basic or Pascal. Our consideration is directed toward goals presented in Croatian Educational National Standard (CENS) for elementary school computer science. This standard was carried out in Republic of Croatia during 2004-2005. (Budin et al. 2006). Our long experience of working with students enrolled in basic computer programming courses (Introduction to computing and Programming 1), undoubtedly imply many difficulties that have occurred due to late beginning of learning computer programming. Sometimes teachers wonder about programming language to use as a starting point. Our opinion is that choice of programming language is not the most important point, but more relevant is an early start of computer programming during elementary education.

We believe that programming course may be introduced to primary education if an efficient framework supporting students in learning programming is available. Motivated by this problem, we have started developing of prototype model of computerized tutor for learning and teaching programming. In this chapter we introduce our system that helps student to acquire basic programming knowledge. We believe that application of this system in real classroom may greatly improve process of learning and teaching of computer programming. It should realize assumptions for students and teachers to solve mentioned problems in an individualized computer-based approach combined with traditional education. Development of the computerized tutor for learning and teaching of programming is based on age-long research and development of the Tutor–Expert System, a hypermedia authoring shell model for building intelligent tutoring systems (Stankov 1997).

LEARNING PROGRAMMING AND TUTORING SYSTEMS

Researchers and experts of the field agree that learning and teaching programming is a complex activity for students and teachers. Understanding and visualization of abstract processes make problem to students when they learn programming as well as other fields with similar characteristics.

Programming Knowledge to Acquire

Linn and Dalbey (1989) have identified the ideal chain of cognitive achievement by learning to