Computers & Education 82 (2015) 162–178

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

# Programming assistance tools to support the learning of IT programming in South African secondary schools

Melisa Koorsse*, Charmain Cilliers [1], André Calitz [2]

*Nelson Mandela Metropolitan University, P.O. Box 77000, Port Elizabeth, 6031, South Africa*

ARTICLE INFO

*Article history:*
Received 25 August 2014
Received in revised form
17 November 2014
Accepted 26 November 2014
Available online 3 December 2014

*Keywords:*
Programming and programming languages
Interactive learning environments
Secondary education
Evaluation of CAL systems

ABSTRACT

Novice programmers struggle to understand introductory programming concepts and this difficulty, associated with learning to program, contributes mainly to the lack of interest in the field of Computer Science at tertiary level. Programming assistance tools have been used to assist novice programmers extensively at education institutions. A programming assistance tool (PAT) is a software program that can be used by novice programmers to learn how to program and/or improve their understanding of programming concepts.

This research proposes that novice programmers, specifically Information Technology (IT) scholars in South African secondary schools, could be supported by PATs. The main objective of this research was to determine whether the use of a PAT impacted IT scholars' understanding of programming concepts and their motivation towards programming in general. Criteria for the selection of PATs were derived from the programming difficulties identified in literature and from surveys to IT teachers and scholars. The selection criteria were grouped into programming knowledge and programming skills categories. Existing PATs were evaluated using the selection criteria and three PATs, namely, RoboMind, Scratch and B#, were selected for evaluation in this research study. A convenience sample of schools participated in the study. The three PATs provided different approaches while being able to support the Delphi programming language used in schools that participated in the study.

The findings of this research indicated that, although scholars perceived the PATs to be useful in the explanation of certain of the programming concepts, there was no conclusive evidence that IT scholars who used a PAT had a significantly better understanding of programming concepts and motivation towards programming than scholars who did not use a PAT. Participant feedback was used to identify the strengths and shortcomings of the three PATs and to provide recommendations for the development of future PATs specifically designed to support IT scholars.

## 1. Introduction

Information and Communication Technology (ICT) related professions have been identified as top professions due to the increase in the number of companies developing web, mobile and cloud computing applications (Strieber, 2011). Despite a great demand for ICT qualifications, countries around the world, including the U.S. (Hug, Guenther, & Wenk, 2013; Wilson, Sudol, Stephenson, & Stehlik, 2010), European countries (Porta, Maillet, & Gil, 2010) and South Africa (Alexander & Twinomurinzi, 2012; Calitz, 2010), have experienced a decline in the number of enrolments in Computer Science (CS) programs at tertiary level education institutions. This decline in enrolments has resulted in a decrease in the number of software developers available to industry. Globally countries have identified the importance of ICT professions for technological and economic growth of the country (Wilson et al., 2010). The importance of promoting ICT related skills and knowledge at school level has also been recognised (Wilson et al., 2010).

* Corresponding author. Tel.: +27 41 504 3925.
*E-mail addresses:* Melisa.Koorsse@nmmu.ac.za (M. Koorsse), Charmain.Cilliers@nmmu.ac.za (C. Cilliers), Andre.Calitz@nmmu.ac.za (A. Calitz).
[1] Tel.: +27 41 504 2639.
[2] Tel.: +27 41 504 2235.

http://dx.doi.org/10.1016/j.compedu.2014.11.020
0360-1315/© 2014 Elsevier Ltd. All rights reserved.

In South Africa, ICT related skills and knowledge including software development are included in the subject matter of the Information Technology (IT) subject in South African secondary schools for Grade 10–12 scholars. IT in South African secondary schools is perceived as being a difficult subject (Havenga & Mentz, 2009). This reputation impacts on scholars' decisions to take IT at school. Many scholars that do attempt the subject change to another, easier subject before Grade 12 (the final year) or remain in the subject class but lack the motivation and interest in the programming content to achieve their full potential. The lack of interest and motivation in the subject and negative impression created amongst scholars seems to affect negatively the number of scholars who decide to pursue further education in a computing discipline.

IT scholars are faced with the same difficulties of learning to program as that experienced by novice programmers in general (Havenga & Mentz, 2009). These difficulties are aggravated by the secondary school learning environment (Koorsse, Calitz & Cilliers, 2010). Short (approximately 45 min), daily lessons make it difficult both to teach and practise complex programming concepts and skills during limited school hours. There is consequently inadequate time available for IT teachers to address the specific programming difficulties of individual IT scholars as well as accurately assess scholar understanding of concepts during class time.

IT scholars further lack appropriate support in the form of resources to supplement the prescribed IT subject textbook. Programming assistance tools (PATs) are one such supporting resource (Pears et al., 2007). PATs are designed specifically for novice programmers in order to enhance comprehension of algorithms and computer programs, assist with code debugging and/or assess programming knowledge and skills (Kelleher & Pausch, 2005). PATs support self-study (Al-Imamy, Alizadeh, & Nour, 2006) and promote interaction (Baldwin & Kuljis, 2000). Research studies have evaluated the use of a PAT as part of the teaching environment such as Scratch (Malan & Leitner, 2007) and a web-based tool (Al-Imamy et al. 2006). However, this study aims to evaluate the impact of PATs provided as part of self-regulated learning. It is also not clear from literature if there are specific PATs that could be used to promote the understanding programming concepts and skills that are required specifically for IT scholars. None of the PATs identified in literature have been evaluated in terms of the programming knowledge and skills supported or the PATs can be used to support the learning of IT programming in South African (SA) secondary schools.

This research study was important in identifying the programming knowledge and skills required by novice programmers in general (Section 2), and specifically IT scholars in SA schools (Section 2.2). This resulted in a list of selection criteria that indicated the programming skills and knowledge that should be supported by a PAT (Section 2.2). Literature identifies many PATs that have been developed for use by novice programming subject (Section 3) with different programming environments. However, using the selection criteria, this research study was able to select PATs that met the requirements of IT programmers (Section 3). The research study then evaluated whether or not PATs could have an impact on IT scholar understanding of programming concepts and thus whether or not PATs could support the teaching and learning of IT programming in SA schools (Section 6). Three PATs were selected for evaluation using the selection criteria identified. The three PATs were RoboMind (Section 3.2.1), Scratch (Section 3.2.2) and B# (Section 3.2.3). The following research questions were addressed by the study to determine the impact of these PATs:

RQ1: *What impact does the use of the PATs have on IT scholar knowledge of programming concepts?*
RQ2: *What impact do the PATs have on IT scholar perceived difficulty of programming?*
RQ3: *What impact do the PATs have on IT scholar perceived usefulness with regards to the understanding of specific programming concepts?*

The analysis of the results related to the research questions provided feedback on the impact, if any, of the different PATs on the understanding of programming concepts from a scholar's perspective and in terms of formal assessment (Section 5). Certain of the programming concepts such as loops were better supported by PATs and supported scholar understanding of these concepts (Section 5). However, overall there was no evidence to support that scholars using the PATs had a significantly better understanding of programming concepts than scholars who did not. Qualitative feedback indicated the importance of the PAT interface, in particular that the interface should not be too simple. The results and findings of the research study provide further criteria and design guidelines for PATs to be developed to support the learning of IT programming in SA schools, and programming in general (Section 6).

## 2. PAT selection criteria

A dependence relationship exists between the different types of knowledge required by a novice programmer (syntax, computers, programming principles and programming concepts) and code comprehension and generation (Fig. 1). Novice programmers need to understand that code generation involves three sequential steps:

1. *Studying a given problem statement or set of requirements and deciding on the best programming strategy to use*. This skill requires an understanding of the programming principles that can be applied as well as knowledge of how a computer executes code in order to formulate a solution.
2. *Producing an algorithm to solve the problem.* The algorithm will often be formulated using pseudo code. Knowledge of programming principles and concepts is required in order to select appropriate data types and structures or apply concepts such as looping or selection, for example. Knowledge of computer execution of code would assist programmers to efficiently and effectively order the sequence of algorithm instructions.
3. *Translating the algorithm into the code of the programming language being used* requires knowledge of the programming language syntax. The program code is tested and changed until the program meets the original set of requirements and thus solves the problem.

The development of programming strategies in terms of applying programming knowledge correctly when formulating a solution to a programming problem is important. Different knowledge and skills that are important in the programming process (Fig. 1). A particular PAT may not address all these areas, instead supporting only a few areas. Surveys were administered to IT scholars and teachers to identify the deficiencies in the skill and knowledge areas required by a programmer (Section 2.2). By identifying skill and knowledge areas that need attention for novice programming (Section 2.1) and particularly in IT schools (Section 2.2), selection criteria for PATs were formulated