Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 83 (2013) 605 – 610

Procedia
Social and Behavioral Sciences

2nd World Conference on Educational Technology Researches – WCETR2012

Learning programming, problem solving and gender: A longitudinal study

Halil Yurdugül[a]*, Petek Aşkar[b]

[a] Hacettepe University, Computer Education and Instructional Technologies, Ankara 06800, Turkey
[b]TED University, Faculty of Education, Ankara 06100, Turkey

Abstract

In this study, the differences between gender and general problem solving skills in programming knowledge were investigated. The types of programming knowledge were considered in three groups: conceptual, syntactic, and strategic knowledge. In the data analysis, latent growth model was used with longitudinal data. The results demonstrated the significant differences in favor of male students in prior conceptual and strategic knowledge. Male students were more increased their conceptual knowledge scores and strategic knowledge scores than female students during course of programming. Female students were more successful than male in initial status and in development of syntactic programming knowledge. According to other results, the higher level of the problem solving skill had student, the higher level of all knowledge of programming increased over time.

Selection and/or peer-review under responsibility of Prof. Dr. Hafize Keser Ankara University, Turkey

Keywords: Programming, gender differences, problem solving, latent growth modeling;

Introduction

Computer programming involves complex cognitive skills and in this matter Ramalingam, LaBelle and Wiedenbeck (2004) denoted that programming is a high-level cognitive activity that requires the programmer to develop abstract representations of a process in the form of logic structures. Programming demands a lot of knowledge and skill, ranging from an understanding of the programming language and environment through to generalized problem-solving strategies and knowledge of the domain in which the program will be used (McGill, Volet and Hobbs, 1997). Bayman & Mayer (1988) and McGill & Volet (1997) suggested three interrelated types of programming knowledge to be necessary to understand the underlying complex processes involved in programming: a) conceptual knowledge, b) syntactic knowledge, and c) strategic knowledge. Conceptual knowledge is about understanding of computer programming constructs and principles and involves the development of mental models of the system as well as understanding the meanings of the actions that are executed by the program. Syntactic knowledge refers to knowledge about programming language features and rules for its use. Strategic knowledge is the ability to use syntactic and conceptual knowledge in the most appropriate and effective way to solve

* Corresponding author. Halil Yurdugül Tel.: +90-312-297-7176
E-mail address: yurdugul@hacettepe.edu.tr

1877-0428 © 2013 The Authors. Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
Selection and/or peer-review under responsibility of Prof. Dr. Hafize Keser Ankara University, Turkey
doi:10.1016/j.sbspro.2013.06.115

programming problems (Yuen, 2006).

Each of these programming knowledge types (conceptual, syntactic, and strategic) is branched into two groups as declarative knowledge and procedural knowledge.

Table 1: Types of programming knowledge (Mcgill & Volet, 1997)

	Declarative Knowledge	**Procedural Knowledge**
Conceptual Knowledge	Understanding of and an ability to explain the semantics of the actions that take place as a program executes.	Ability to design solutions to programming problems
Syntactic Knowledge	Knowledge of syntactic facts related to a particular language.	Ability to apply rules of syntax when programming
Strategic/Conditional Knowledge	The ability to design, code, and test a program to solve a novel problem.	

Declarative knowledge, or "know that", is commonly defined as knowledge about something, specifically about certain facts, concepts, or principles. Procedural knowledge, or "know how", is commonly defined as knowledge of how to do something and it refers to the active use of declarative knowledge base when solving problems. The major skill in programming is problem solving based on strategic (conditional) knowledge. Strategic knowledge is the ability to use syntactic and conceptual knowledge in the most appropriate and effective way to solve programming problems (Mcgill & Volet, 1997).

In this study, gender differences with problem solving skill in cognitive development for three programming knowledge was investigated. In related literature, there are studies investigating the relation between gender and achievement of computer programming. In most of these studies (Linn, 1985; Bruck, Jenson, and DeBonte, 2002; Crews & Butterfield, 2003; Webb, 1985), significant gender differences in achievement of programming were not reported. Furthermore, in these studies, exploratory data analysis methods based on latitudinal data were used. Nevertheless, these methods using latitudinal data cannot analyze the development of students. The developmental analysis is possible with 'within subject methods' based on longitudinal data. Another matter in this study was the relation between students' problem solving skill and achievement of programming. In some researches, problem-solving skill effects on the programming learning. Liao and Bright (1991) reported that the effects of computer programming on students' cognitive development. Additionally, Hyde et al.'s (1990) meta-analysis of 100 studies suggested that gender differences in mathematics performance were small but gender differences in mathematical problem solving with lower performance of women existed in high school and in college.

This study aims to find out the gender differences in development in programming learning domains within conceptual, syntactic, and strategic knowledge types. In addition, we investigated the relation between students' development in programming learning and problem solving skills for gender.

Methods:

We used the latent growth analysis to find out the development of students' programming learning. The data gathered from students who attended to computer programming course in a department of Computer Education and Instructional Technologies. The sample group consists of 86 students, 48 male and 38 female. The achievement measurement tool, designed to measure programming knowledge and skills of students in conceptual, syntactic and strategic learning domains, was administered three times periodically. At the beginning of the semester, we also measured students' general problem solving skills with Problem Solving Inventory, Form-A (PSI-A), developed by Heppner and Petersen (1982). We used the multivariate latent growth model to determine the students' developmental process in each programming learning domains (in Table 1) with gender and general problem solving skill as time-invariant covariates (Duncan *et al.*, 2006).

Latent Growth Modeling:

Latent growth modeling has been used for analyzing the longitudinal data gathered from more occasional data. Latent growth models analyze longitudinal changes in means, variances, and covariances of variables. Latent growth curve methodology also provides a means of modeling a developmental function as a factor repeated observations over time in educational researches (Aşkar & Yurdugül, 2009). Many statistical methods are available for longitudinal data e.g. dependent *t* tests, repeated measures ANOVA, multivariate ANOVA, autoregressive models and random effect models. However, in LGM, cognitive development and achievement growth is considered as a psychological construct. LGM has also been found better than univariate ANCOVA (Muthén & Curran, 1997; Curran & Muthén, 1999), ANOVA and dependent *t* tests, and was also highly more powerful than MANOVA (Fan & Fan, 2005; Fan, 2003). However, Fan (2003) suggested that ANOVA may be more powerful in detecting differences in intercepts (Wanström, 2009).

Each student's achievement measures were gathered over three times and were presented as graphics named cognitive development trajectory. We used latent growth modeling (LGM) to estimate students' cognitive development trajectories of change in programming learning outputs using a structural equation modeling program (LISREL 8, Jöreskog & Sörbom, 1993). These trajectories describe intra-individual change over time by estimating the initial levels (named as the intercept) in t_0 and rates of change (named as the slope). To investigate the trajectories of individual change in a variable, two latent constructs corresponding to initial level (intercept) and slope are defined in a structural equation model. Measurements of the variables (X_1, X_2, and X_3) at different time points (t_0, t_1, and t_2) are used as multiple indicators of the two latent constructs in this model.

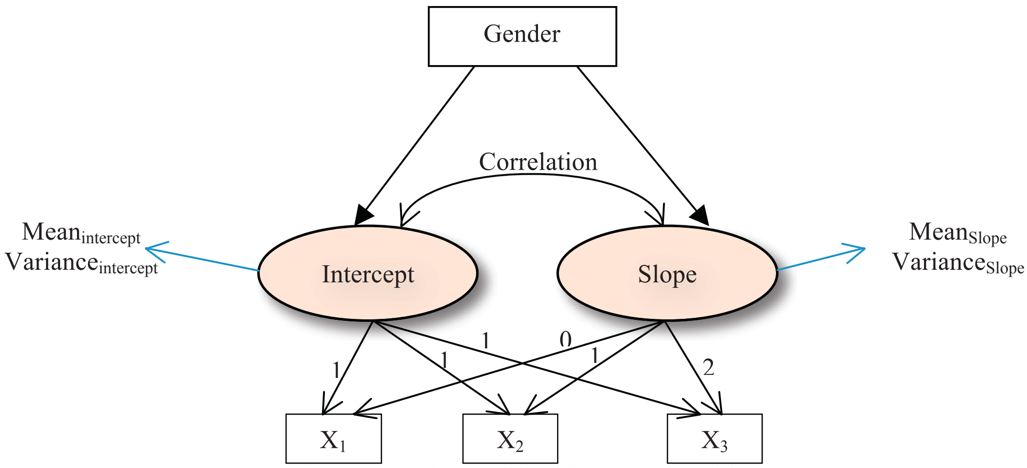

Figure 1: Univariate conditional latent growth model

In the (unconditional) LGM, there are five parameters allow us to investigate both developmental change and stability in programming learning over time. The parameters are a) mean of intercept (initial status), b) variance of intercept, c) mean of slope, d) variance of slope, and covariance/correlation of intercept and slope. If the mean of the intercept is statistically significant, it shows us that students have prior knowledge at beginning of semester. The variance of the intercept indicates the heterogeneity of student's prior knowledge if it is statistically significant. If mean of the slope is statistically significant, the slope mean indicates a significant average increase in the students' knowledge over time. The variance of slope indicates the heterogeneity of student s' development in programming learning, if it is statistically significant. The covariance/correlation between the two latent factors indicates the initial level and rate of change over time are related.

The conditional LGM contains covariate variables such as gender, socio-economic status, or level of problem solving skill.

Results

The univariate unconditional LGMs provided a very good fit to the data with all goodness of fit measures at an acceptable level. The initial status of students' conceptual knowledge in programming was 1.31 and it increases at a rate of .0.74 over time. Similarly, the initial status of students' syntactic knowledge in programming was 2.15 and it increases at a rate of 1.92.

Table 2: Parameters of unconditional latent growth models

	Types of Knowledge		
LGM Parameters	**Conceptual**	**Syntactic**	**Strategic**
Mean of Intercept	1.31**	2.15**	0.24**
Variance of Intercept	1.04**	1.42**	0.97**
Mean of Slope	1.74**	1.02**	0.84**
Variance of Slope	0.67**	0.49**	0.96**
Covariance (Intercept, slope)	-0.24**	0.19**	0.31**

At the beginning of the semester, students' average level of conceptual knowledge was 1.31 (P≤0.05), and of syntactic knowledge was 2.15 (P≤0.05), and of strategic knowledge was 0.24 (P≤0.05) in the programming course. Students brought more syntactic knowledge than other kinds of programming knowledge from high school to higher education. Additionally, students had prior knowledge at different level of conceptual (1.04; P≤0.05), syntactic (1.42; P≤0.05), and strategic knowledge (0.97; P≤0.05), as seen from value of variance. It should be borne in mind that variance is individual differences parameter.

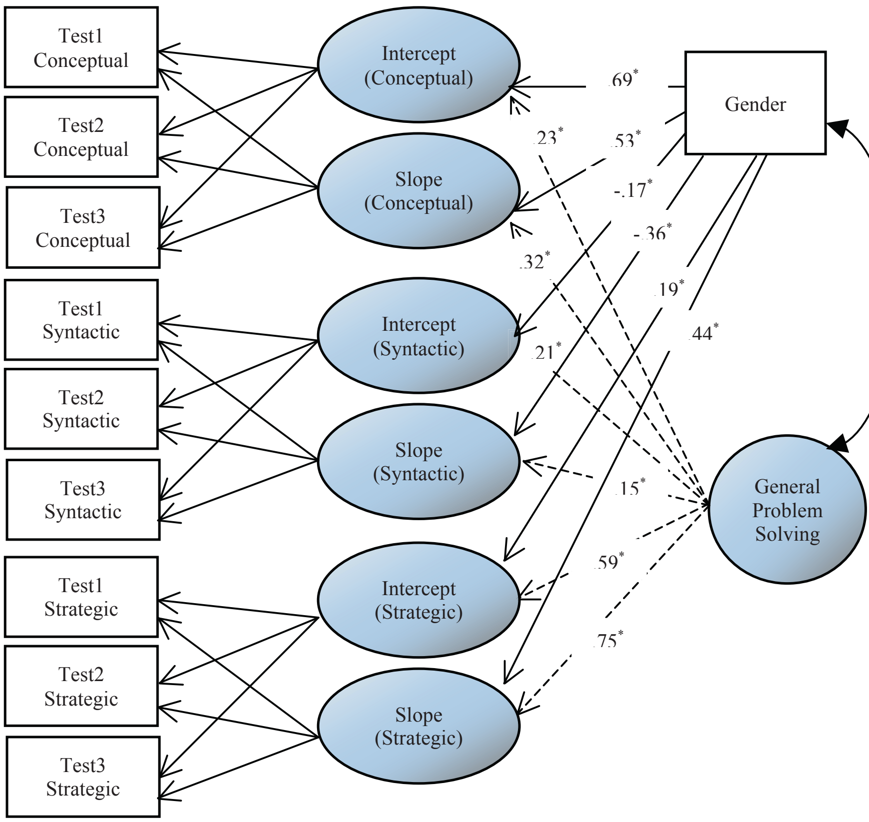

Figure 2: Multivariate conditional latent growth model in programming learning

Each variance in Table 2 indicates individual heterogeneity around the average level of programming knowledge. During the course of programming, on average, students increased their conceptual knowledge score by 1.74, and syntactic knowledge score by 1.02, and strategic knowledge score by 0.84 units at each time point. There was significant variation across students in this rate of achievement growth (cognitive development) according to variance of slopes in each knowledge types. In Table 1, the covariance (-.24) between intercept and slope latent factors of conceptual knowledge is significant at the level $p \leq 0.05$, indicating the initial level of conceptual knowledge and development rate of conceptual knowledge over time are related. The negative correlation means that students who have higher initial conceptual knowledge have a lower rate of increase (and/or students who have lower initial conceptual knowledge have a higher rate of increase) and this situation may be caused by misconceptions which student brought from high school.

The result of conditional LGM includes differences between gender and general problem solving skill was given in Figure 2. Gender was coded as 1 for male and 0 for female. That is why, the positive and significant differences are in favor of male students. The results demonstrated the significant differences in favor of male students in prior conceptual and strategic knowledge. Male students increased their conceptual knowledge scores (0.53; $P \leq 0.05$) and strategic knowledge scores (0.44; $P \leq 0.05$) more than female students during the programming course. Female students were more successful than males in their initial status and in development of syntactic programming knowledge.

Another covariate variable in this study was students' general problem solving skill. According to the results given in Figure 2, the higher level of the problem solving skill had student, the higher level of all knowledge of programming increased over time. Additionally, general problem solving was related to gender issue (0.63; $P \leq 0.05$). Moreover, male students had more general problem-solving skill than females. The development of strategic knowledge (ability to design, code, and test a program to solve a novel problem) and conceptual knowledge over time was in favor of males. One of the reasons why male students developed more in conceptual and strategic knowledge might be attributed to their problem solving skills.

References

Aşkar, P., & Yurdugül, H. (2009). The using of latent growth models for educational researches. *Elementary Education Online, 8(2),* 534-555,.

Bruckman, A., Jenson, C., & DeBonte, A. (2002). Gender and Programming Achievement in a CSCL Environment. In *Proceedings of the 2002 Conference on Computer Supported Collaborative Learning*, 119-227).

Crew, T., & Butterfield, F. (2003). Gender differences in beginning programming: an empirical study on improving performance parity. *Campus-Wide Information Systems, 20(5)*, 186-192.

Curran, P. J., & Muthén, B. O. (1999). The application of latent curve analysis to testing developmental theories in intervention research. *American Journal of Community Psychology, 27*, 567-595.

Duncan, T. E., Duncan, S. C., Strycker, L. A., Li, F., & Alpert, A. (2006). *An introduction to latent variable growth curve modeling: Concepts, issues, and applications*. 2nd Edition. Mahwah, NJ:Erlbaum.

Fan, X. & Fan, X. (2005). Power of latent growth modeling for detecting linear growth: number of measurements and comparison with other analytic approaches. *Journal of Experimental Education, 73*, 121-139.

Fan, X. (2003). Power of latent growth modeling for detecting group differences in linear growth trajectory parameters. *Structural Equation Modeling: an Interdisciplinary Journal, 10,* 380-400.

Heppner, P. P., & Peterson, H. C. (*1982*). The development and implications of a personal problem-solving inventory. *Journal of Counseling Psychology, 29*, 66-73.

Hyde, J. S., Fennema, E. and Lamon, S. J. (1990) Gender differences in mathematics performance: A meta-analysis, *Psychological Bulletin*, 107(2), 139-155.

Liao, Y. C., & Bright, G. W. (1991). Effects of computer programming on cognitive outcomes: A metaanalysis. *Journal of Educational Computing Research, 7,* 251-268.

McGill, T.J. & Volet, S.E. (1997). A conceptual framework for analysing students' knowledge of the various components of programming. *Journal of Research on Computing in Education* , 29(3) 276-297.

McGill, T., Volet, S., and Hobbs, V. (1997). Studying programming externally: Who succeeds? *Distance Education*, 18 (2), 236-256.

Muthén, B. O., & Curran, P. J. (1997). General longitudinal modeling of individual differences in experimental designs: a latent variable framework for analysis and power estimation. *Psychological Methods, 2*, 371-402.

Ramalingam, V., LaBelle, D., Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. The 9th Annual Conference on Innovation and Technology in Computer Science Education, Leeds, UK. 171-175

Wanström, L.(2009).Sample sizes for two-group second-order latent growth curve models. *Multivariate Behavioral Research, 44(5)*, 588-619.

Webb, N. M. (1985). The role of gender in computer programming learning processes. *Journal of Educational Computing Research, 1,* 441-458.

Yuen, A. H. K. (2006). Learning to program through interactive simulation. *Educational Media International,* 43,3, 251–268.