Implicit Theories of Programming Aptitude as a Barrier to Learning to Code: Are They Distinct from Intelligence?

Michael James Scott
Information Systems, Computing & Mathematics
Brunel University
Uxbridge, Middlesex, UB8 3PH
United Kingdom
michael.scott@brunel.ac.uk

Gheorghita Ghinea
Information Systems, Computing & Mathematics
Brunel University
Uxbridge, Middlesex, UB8 3PH
United Kingdom
george.ghinea@brunel.ac.uk

ABSTRACT

Contemporary psychology has shown that self-theories can have a profound influence on affect and behavior. Entity-theorists, believing their traits are fixed, adopt maladaptive learning strategies in the face of difficulty. In contrast, incremental-theorists, believing their qualities can change, often adopt mastery-orientated strategies. However, can this concept be domain-specific? This poster presentation challenges the notion of a single dominant mindset. People can nurture a variety of beliefs about different traits, so in the minds of learners, programming aptitude may not be the same as intelligence. The results from a confirmatory factor analysis of 94 responses to an undergraduate programming experience survey indicate that beliefs towards aptitude are empirically distinct from those towards intelligence, suggesting that alternate self-traits should be considered when extending self-theories into specific domains.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education.

General Terms

Theory, Human Factors.

Keywords

Self-Theories, Implicit Theories, Entity, Incremental, Dweck, Aptitude, Intelligence, Programming, Learning.

1. INTRODUCTION

Programming is a craft which often demands ongoing reflexive practice to master. Unfortunately, students often encounter barriers that can impede such practice [3]. One barrier could be the different ways the act of learning to program can be experienced. Negative experiences are often associated with difficulties, but some learners appear able to persevere. In line with research being conducted on self-theories [1], it has been proposed that implicit beliefs about innate ability could be a significant contributor to this phenomenon. According to the theory, people conceive their traits as either being malleable qualities that can grow with them (the incremental theory), or as natural traits that do not change (the entity theory). However, within the context of learning to program, is the concept of *intelligence* the same as *programming aptitude*? More importantly, do the students themselves believe they are the same?

Table 1: Fit Indices and Criteria for the Models

Fit Index	1-Factor Model	2-Factor Model	Adequete Fit Criteria [2]
SRMR	.146	.077	$< .08$
CFI	.507	.959	$> .90$
RMSEA	.092	.027	$< .08$
Bollen-Stein p	.062	.369	$> .05$

2. FINDINGS

A survey adapting Dweck's [1] psychometric scale was distributed to undergraduate students on two programming modules at the authors' institution using SurveyMonkey. There were 94 respondents (a response rate of ~25%). The data was analyzed using the unweighted least squares method of confirmatory factor analysis in AMOS 21.0.0, with all cases included in the analysis.

The results show that the single factor model, based on the notion of a single dominant self-theory, was significantly different to the data ($\chi^2 = 64.330, df = 36, p = .003$). In contrast, the two-factor model, where intelligence and programming aptitude are distinct self-traits, had adequate fit ($\chi^2 = 36.382, df = 34, p = .358$). This is further illustrated by the fit indices presented in Table 1 above. This finding suggests that conceptions towards different self-traits should be considered separately when extending self-theories to specific domains, such as programming education.

3. REFERENCES

[1] C. S. Dweck. *Self-Theories: Their Role in Motivation, Personality, and Development.* Psychology Press, PA, USA, 2000.

[2] J. Hair, B. Black, B. Babin, and R. Anderson. *Multivariate Data Analysis, Seventh Edition.* Psychology Press, NJ, USA, 2010.

[3] M. J. Scott and G. Ghinea. Educating programmers: A reflection on barriers to deliberate practice. In *Proceedings of the 2nd HEA Conference on Learning and Teaching in STEM Disciplines, (Birmingham, UK, Apr 17-19, 2013)*, page 028P.

ITiCSE'13, July 1–3, 2013, Canterbury, England, UK.
ACM 978-1-4503-2078-8/13/07.