Learning Styles and Performance in the Introductory Programming Sequence

Lynda Thomas, Mark Ratcliffe, John Woodbury and Emma Jarman
Department of Computer Science
University of Wales, Aberystwyth, UK
{ltt, mbr, jyw}@aber.ac.uk

Abstract

This paper reports on the implication of different preferred learning styles on students' performance in the introductory programming sequence and on work in progress on how to accommodate these different styles.

Students were given a learning styles preference test and then their preferred learning styles were compared to their performance on the exam and the practical programming part of the introductory programming module. There were significant differences in performance between groups of students.

This result could lead one to two possible conclusions. One might be that some students' learning styles are more suited to learning programming than others.

An alternative explanation is that our current methods of teaching advantage students with certain learning preference styles. We are at present in the process of testing this second assumption by providing students with a wider range of learning materials. We will then see if student performance is improved by using our current results as a baseline for comparison

Keywords

Learning styles, Software Engineering Education, Introductory programming, Diagnostic testing, Student profiling.

1 Motivation

There seems to be little doubt that students frequently have problems with introductory programming courses. Anecdotal evidence leads us to recognise that many students of Software Engineering and Computer Science claim to 'hate programming' and not be able to do it. Objective assessment confirms that student performance in this essential activity is nothing like as good as educators would like to believe [3].

SIGCSE'02, February 27- March 3, 2002, Covington, Kentucky, USA.

Many students seem to be put off by introductory programming courses. They either fail and give up on the subject or they continue with their degrees but vow that their future careers will not include programming. A similar problem with student achievement in introductory courses is, in fact, experienced in all the sciences. Sheila Tobias [10] has completed a study of introductory science courses in which she concludes that many students are driven away from science by the failure of introductory courses to motivate students, by the passivity and competitiveness that is forced upon them and by the focus on algorithmic problem solving rather than conceptual understanding. We may not always do so, but there is no reason to suppose that we cannot remedy some of these problems in introductory programming modules and make our courses more interesting and appealing to the wider range of students who now enter higher education.

In our search to do this at Aberystwyth, we are in the process of developing an educational environment that allows students to maintain control of their own learning by providing a suite of resources that bundles traditional pedagogical approaches such as lectures, tutorials and seminars, with innovations exploiting new technologies based on the Internet [11]. Our consideration of how to improve the first year sequence and other courses has led us to consider the issue of individual learning styles [9].

In this paper we examine correlations between preferred learning style and performance on both the exam and practical aspects of an introductory programming course. We then consider the resulting implications for the design of our learning environment.

2 Learning Styles

Let us first examine some theories of how people best learn. Educational psychologists have long been of the opinion that different people learn in different ways. This usually manifests itself as a preference for one style, rather than absolute categorisation of individuals.

There have been many characterisations of learning style preference. One theory, first popularised in the 1960s, claims that each brain hemisphere is specialised for a different mode of thinking: the left for linguistic, analytical and sequential tasks and the right for artistic, gestalt and creative tasks [6] (this may be what DeBono describes as 'lateral thinking'.)

Another way of looking at learning examines it from the perspective of Visual, Auditory and Kinaesthetic/Tactile preferences. Variations of this analysis have, for example, been applied in the context of learning English as a second language [5]. Some writers believe that each of these preferences is then used at a different level of cognition: perception, organisation and retention [3].

Richard Felder has studied different learning styles in the context of Chemical Engineering. Felder's writings [2] provide both a valuable overview and an extension of the work in this area, and also prescribe an approach to education that incorporates the recognition of different learning style preferences. Initially, it is observed that students exhibit a preference for different ways of internalising and processing information: by seeing and hearing, via reflection and action, logical and intuitive reasoning, analysing and visualising, in a steady stream or in fits and starts.

Felder reviews four models of Learning Styles:

The Myers-Briggs Type Indicator (MBTI) categorises students according to their position on scales based on Jung's theory of psychological types. The types being extraverts/introverts, sensors/intuitors, thinkers/feelers, judgers/perceivers.

Kolb's Learning style Model classifies people along the scales of preference for 'concrete experience' or 'abstract conceptualisation' (how information is taken in), and 'active experimentation' or 'reflective observation' (how they internalize information)..

The Herrmann Brain Dominance Instrument (HBDI) uses theory about the specialised functioning of areas or quadrants of the brain to classify students' learning preferences by left and right brain dominant and cerebral or limbic.

Felder's own synthesis of these learning styles is:

The Felder-Silverman Learning Style Model [7] which classifies students as

- active learners (learn by trying things out, working with others) <--> reflective learners (learn by thinking things through, working on their own)
- sensing learners (concrete, practical, oriented toward facts and procedures) <--> intuitive learners (conceptual, innovative, oriented toward theories and meanings
- visual learners (prefer pictures, diagrams, flow-charts) <--> verbal learners (prefer written or spoken explanations)
- sequential learners (learn in incremental, orderly steps) <--> global learners (holistic, learn in large leaps)
- inductive learners (prefer explanations that move from the specific to the general) <--> deductive learners (prefer explanations that move from the general to the specific)

The emphasis in Felder's work is on *preferred* learning style, not ability. He notes that

> "A student's learning style profile provides an indication of probable strengths and possible tendencies or habits that might lead to difficulty in academic settings. The profile does not reflect a student's suitability or unsuitability for a particular subject, discipline, or profession." [2]

The purpose here is to identify what may be *easy* for the students as opposed to the areas in which they may be more challenged.

> "Functioning effectively in any professional capacity, however, requires working well in all learning style modes. ... If professors teach exclusively in a manner that favors their students' less preferred learning style modes, the students' discomfort level may be great enough to interfere with their learning. On the other hand, if professors teach exclusively in their students' preferred modes, the students may not develop the mental dexterity they need to reach their potential for achievement in school and as professionals. An objective of education should thus be to help students build their skills in both their preferred and less preferred modes of learning." [2]

Felder notes that for at least the past decade, engineering instruction has been biased heavily towards the intuitive, verbal, reflective, sequential learning styles, yet few students fall neatly into all of these five categories.

Whichever model of the learning styles one favours or accepts, Felder's crucial observation for us is that there is and has been a profound mismatch between learning and teaching style preferences. Such a mismatch may result in student disinterest and apathy on one hand, manifesting in lower grades, or on the other hand in extreme circumstances students may even drop out of courses altogether, denying the profession some potentially great contributors. He therefore advocates teaching strategies that incorporate all of the learning style preferences.

3 Implications

We were interested in this topic at Aberystwyth because we are in the process of designing an environment, MAP (Monitoring, Assessment and Provision) [11], that encourages students to reflect on their own learning preference and then pick revision activities that suit this preference. This work is ongoing, but initially and to obtain a baseline, we have tested the entire first year introductory programming class to determine their preferred learning styles. We then correlated these results with performance on the course work and exam portion of the course. The results are being used in the design of revision materials that form part of the MAP system.

4 The Experiment

We conducted an experiment to test the hypothesis that different kinds of learners perform differently on the introductory programming course. We had no preconceived ideas about how learning style preference would be related to achievement (although a recent paper has suggested that students who prefer abstract conceptualisation and active experimentation on the Kolb test may perform better on this kind of course [1].)

The introductory programming sequence at Aberystwyth consists of two modules that use Java as the programming language. Students are divided into two sections, but both sections have the same exams and assignments and are taught by the same instructor.

Students learning styles were obtained by a multiple choice test that used the Felder-Silverman scale [7]. This test tested students along 4 axes and grouped them into 16 separate categories. (Felder believes that the best approach to teaching undergraduates is always inductive and thus the test does not differentiate between inductive and deductive learners.) The results of the learning style test were then correlated with achievement in the course

5 The Results

Results were taken from the second of the two-course sequence, omitting students who had dropped out of the course or had not completed the learning styles questionnaire.

Overall results of the course were typical for British institutions. One hundred and seven students were tested, the average mark on the final exam was 56.21% with a standard deviation of 15.65. The average mark on the programming assignment was 62.86% with a standard deviation of 17.44.

category	course mark	exam mark	n
overall	62.86	56.21	107
active	61.59	52.80	58
reflective	64.37	60.26	49
sensing	62.11	54.54	54
intuitive	63.62	57.92	53
visual	61.43	54.29	82
verbal	67.56	62.52	25
sequential	64.07	54.48	54
global	61.62	57.98	53

Table 1: Summary of results for each categorisation

Our first analysis was to look at each of Felder's categorisations and see if there were any clear differences between groups of students. The results are summarised in Table 1.

The two statistically significant differences (using the t-test) that emerged from this analysis were that, on the exam portion of the course, reflective learners scored higher than active learners (p=.015) and verbal learners higher than visual learners (p=.027). We also noted the small number of verbal learners and that sequential learners scored higher than global learners on the course work but lower on the exam

Felder's observation that engineering instruction has been biased heavily towards the reflective, intuitive, verbal, sequential learning styles is born out in our results. (Assuming that the result of such bias would be higher achievement of students with those learning styles.)

Our next analysis was to look at the 16 individual groups of students as determined by the four learning style categories. We compared each category against the group as a whole.

group	course mark	exam mark	n	notes
overall	62.86	56.21	107	
reflective intuitive verbal global	68.89 (p=.34)	67.44 (p=.05)	9	higher overall, *much* higher on exam
reflective intuitive visual global	58.55	60	11	the only group that did better on the exam than the course work
reflective sensing verbal sequential	82.33 (p=.015)	79.5 (p=.011)	3	small sample but very high results
reflective sensing visual sequential	67.23	54.42	13	did better than average on course work, worse on exam
actual sensing verbal sequential	73.6 (p=.052)	54.10	5	much better on course work

Table 2: Smaller group analysis

Some groups were very small, and these and groups with little variation from the overall sample are omitted. Other results are summarised in Table 2.

As previously mentioned, there are two ways that we could use these results. We *could* continue to teach as at present and give entering students a learning style preference test and admit them to our program based on their results. This would, however, do nothing for the shortage of qualified people in our field and stand against the purpose of a university education, which is to extend opportunities rather than narrow them.

Instead, we are choosing to create materials that present the information and skills of the introductory programming course in ways that appeal to different kinds of learners' preferences. As well as learning the material, students can also reflect on their own learning styles and work on strengthening their less preferred modes of learning.

6 Recommendations and Future Work

So, what have we learnt from this exercise and where does it take us?

The first thing is that Felder's observations about engineering education may well be true in Software Engineering education. Judging by student performance, we are probably favouring students with certain kinds of learning style preference in our provision and we need to provide opportunities for *all* types of learners. By providing scaffolding for students with different learning style preferences and encouraging students to be aware of their own individual needs and preferences, we can only enhance the learning experience.

With this in mind, the MAP environment is designed so that students have control of their learning experience. Students have individual profile pages from which links are set up to direct them to the learning resources that best suit their needs. For instance, students with a verbal preference who prefer learning from lectures may be directed to lectures available on RealVideo. Visual learners are directed to animations and diagrams that are also available on line.

Secondly, we have obtained a baseline against which we can test the emerging MAP environment. This academic year we again intend to test students for their learning style preference. We then provide activities, focused towards different learning styles, which are available to students through the environment. It will be interesting to see if there is a change in students' performance.

Thirdly, it appears from the results of this experiment that active, sensing, and visual learners may be particularly disadvantaged by current methods of teaching. To this end we have provided Web-based interactive revision aids, for example, in the form of formative objective tests that are designed to appeal to active (learn by trying things out) and sensing (concrete, practical) learners. We are also currently developing, with the aid of postgraduate students, several projects including simple packages explaining the basic concepts associated with traditional programming skills and hardware. Again the emphasis in these educational aids is on a place for students to experiment and work with others in a practical, visual environment. In particular, we are developing an environment for collaboratively creating UML designs that allows students to observe the behaviour of their designed classes in a visual, animated and interactive way.

One of the fundamental design features of the MAP environment is that students themselves need to take control of their own learning. Research has shown that methods of teaching and learning which combine significant student autonomy, conjecturing and articulation with dynamic scaffolding by the teacher are highly effective [8]. In the MAP environment we provide students with the opportunity to reflect on their own learning and choose materials which enhance it in their own preferred manner.

We have learnt a lot about our students from performing this experiment and we will be interested to see whether the MAP environment indeed successfully caters to the wide variety of student learning preferences with which we are presented.

References.

[1] Byrne, Pat and Gerry Lyons, The Effect of Student Attributes on Success in Programming, *Proceedings of ITiCSE 2001*, Canterbury, Kent, ACM Press, 2001.

[2] Felder, Richard M., Matters of Style, *ASEE Prism,6* (4), 1996.

[3] Markova, Dawna, *The Open Mind: Exploring the Six Patterns of Natural Intelligence*, Conari, 1996.

[4] McCracken, M. et al A multi-national, multi-institutional study of assessment of programming skills of first year CS students, Report by the *ITiCSE 2001* Working Group on Assessemnt of Programming Skills (to appear).

[5] Reid, Joy, *Understanding Learning Styles in the Second Language Classroom*, Prentice Hall, 1998.

[6] Rohr, G., Using Visual Symbols in *Visual Languages*, S.Chang (ed.), Plenum, 1986.

[7] Soloman, Barbara and Richard M. Felder, *Index of Learning Styles*, <http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html>

[8] Tanner, H. and S. Jones, Dynamic Scaffolding and Reflective Discourse: the Impact of Teaching Style on the Development of Mathematical Thinking, *Proceedings of the 23rd Conference of the International Group for the Psychology of Mathematics Education*, Haifa, 1999.

[9] Thomas, L.A. The Implication of Different Learning Styles on the use of UML Notation, poster presented at *CSEET 99*.

[10] Tobias, Sheila, *They're Not Dumb, They're Different: Stalking the Second Tier*, Research Corporation, Tucson, 1990.

[11] Woodbury J, Ratcliffe M, and Thomas L, Building and Deploying an Extensible CAA System: from theory to practice, *CAA 2001*, 2001.