Strategy Transfer between Computer Programming and Mathematical Problem Solving

Caroline Walta
University of Melbourne
<waltacj@mlc.vic.edu.au>

This paper presents part of the findings of a study into the effect of teacher intervention, designed to improve metacognitive skills, on learning transfer between MicroWorlds programming and mathematical problem solving. The study compared two classes of Year 6 students both extensively involved with programming. Students who were exposed to the teacher intervention activities in addition to programming outperformed the group who did not receive the intervention activities on a test of mathematical problem solving.

This paper presents part of the findings of a study into the cognitive and affective outcomes of programming using MicroWorlds software in upper primary classes (Walta, 1999). One aspect of the study, outlined here, sets out to examine the effect of a teacher intervention designed to improve metacognitive skills, on transfer of learning between programming in MicroWorlds (LCSI, 1993) and mathematical problem solving. MicroWorlds is a programming package for children based on the language of Logo, incorporating a range of features, such as drawing, animation, sound and text processing.

The study is underpinned by the writing of Papert (1980) and Polya (1973). Papert created Logo, a language for children to use to make individualised creations using computers. Papert suggested that, through programming, children would acquire problem-solving skills that would transfer into other areas of learning. Papert's initial publication of *Mindstorms* in 1980 led to a proliferation of studies in the 1980s which set out to demonstrate the effect of Logo use on a range of areas, including the development of mathematical problem-solving skills. Many of these studies were later found to have limitations in their research design.

Later studies (Swan, 1991; Clements, 1987), which continued to involve small groups in experimental situations, found that learning transfer did not occur unless teachers were proactive in teaching for transfer. Among such studies was that of Lehrer (1989) who highlights the need for instructional practices to enhance the likelihood that strategies acquired when children learn to program will transfer to other areas. Both instructional method and the type of transfer task (analogical or similarity based) were found to have an influence on the degree and quality of transfer. Analogical transfer occurs when strategies developed in one context are spontaneously applied to a different domain of learning. In this study transfer between programming and mathematical problem solving is one such example. Lehrer concludes that to facilitate analogical transfer, teachers need to assist children to construct and reflect on different ways of composing problems, evaluate the consequences of their choices and establish correspondence between programming and non-programming contexts.

The work of Polya (1975) provided a framework for considering the complexity of mathematical problem solving behaviour. He divided problem solving into four stages; namely (a) Understanding the problem, (b) Devising a plan, (c) Carrying out the plan and (d) Looking back. This framework provided a generic model for considering problem-solving behaviour and structuring the teacher-intervention strategy.

The Classroom Programming Context

The subjects of the research were two classes of Year 6 students at one Melbourne school, who were working extensively with MicroWorlds, with lap-top computers. The study took place over the duration of an entire school year. One class was taught by the researcher and the other by an experienced colleague.

The term programming, as it is used in this study describes such activities as problem definition, design development and organisation, code writing and debugging. It also refers to the specific behaviours associated with the use of a code or computer language to generate outcomes not predetermined by the software. These outcomes include animation of objects, movement of objects within a page to achieve special effects and the use of text boxes 'programmed' to appear at critical points and report specific text. Other techniques described as 'programming' are the use of buttons and sliders, which are devices to which code can be attached to execute a specific effect.

All students involved in the study were taught programming in the same way over a term through a problem-solving, discovery-based learning approach. Students were assisted in the mastery of basic programming syntax through a series of class-based activities, which in total occupied about an hour a week over term one of Year 6.

After basic competence in MicroWorlds was attained in term one, students worked on five Lap-T tasks during the rest of the year. Each Lap-T occupied the students for around four weeks. These tasks had evolved with the previous Year 6 class in the year before the study as a way of integrating learning outcomes across the key learning areas while developing the problem-solving strategies associated with Logo programming. Presentation of each completed task to the class enabled students to share knowledge about programming as well as the content knowledge that was an essential part of the task. The idea of teaching for transfer of these strategies was linked to the desire to improve mathematics problem-solving outcomes for Year 6 students. The topics of the five Lap-T tasks were:

- writing a program to illustrate both problems and solutions associated with a maintaining a balance in a particular ecosystem,
- writing instructional software to teach the calculation of perimeter or area,
- constructing a time-line about women in Australian history,
- representing issues associated with a particular endangered Australian animal and
- using LOTE to construct an interactive game to reinforce certain vocabulary.

Initial observations of student response to these work-tasks indicated that students increasingly showed motivation and persistence to engage in and solve associated problems as the year progressed. In addition, students appeared to demonstrate increasing skill, creativity and sophistication in their ability to interpret problems posed and to write complex MicroWorlds programs in response.

Strategy Training

Central to the research was the teacher intervention strategy adopted in one class and known as strategy training. This was a series of lessons designed to improve students' metacognitive awareness and assist them to make connections between skills and strategies developed in a programming context and those needed for mathematical problem solving and to improve the likelihood of transfer of these skills and strategies. Over a period of eighteen weeks, students in one class (called the Strategy Training Group) were involved in a weekly lesson of approximately forty-five minutes. Activities undertaken in strategy training sessions were largely computer based, involving students working on a short challenging programming task using MicroWorlds software. As part of the session, time was also spent in

planning, discussion, reflection and application of skills into a far-transfer context. There were eighteen sessions, conducted in terms two and three. Each of the sessions followed the same format, which involved five different sections summarised below.

Awareness raising. Students were told that as they undertake short programming tasks they would be considering the strategies they used and their possible application to other learning contexts. This section was included to raise awareness of the types of strategies common to problem-solving contexts and to introduce a metacognitive component to the discussion.

Collective reading and planning. This section was included to highlight the value of understanding the problem and planning for a solution. Students also needed to verbalise their thinking on why their particular solutions were more likely to succeed. This step was included in acknowledgement of research which indicates that encouraging children to use 'think aloud' techniques provides them with a useful strategy to aid in reflecting, and by implication improving, their own thinking (Short & Weissberg-Benchell, 1989).

Group co-operation in the writing and debugging of the program. Students working in pairs were asked to reflect on progress at intervals. This section was included to reinforce the processes of reflecting and debugging. Rowe (1993), in discussing metacognitive outcomes of computer programming, notes the value of students working in pairs to encourage reflection through thinking aloud about their ideas and strategies.

Construction of a similar problem. Children then suggested tasks they could undertake which involved use of similar strategies for seeking solutions. This section was designed to assist in raising awareness of the idea that skills and processes used in one context could be applied in another.

Training to facilitate analogical-based transfer. At the conclusion of each lesson a sample mathematics problem was briefly presented to assist students to generalise generic problem-solving skills learned in the programming context to a mathematical problem-solving example. The class was encouraged to reflect verbally on how the strategies utilised in the programming task could be applied to it.

In total nine different programming tasks formed the subject matter of the strategy training sessions. Each was selected because it was complex enough to require careful planning and consideration and had the potential to be solved using a number of processes and strategies. The activities were designed so that one strategy seemed the most likely approach to solving the problem posed by the wording of the task. For example, the strategy "Draw a picture or diagram" was discussed through an activity of designing a car rally and programming one car to win the race. Reproducing a complex geometric design on the computer illustrated "Divide the problem into smaller related parts/work backwards from a given problem". Running a pre-written program and then rewriting the program using another approach to produce the same outcome illustrated "Relate the problem to a similar related problem".

Method

The study involved a comparison between the mathematical problem-solving performance of the two Year 6 classes. One class (the Strategy Training Group or STG), received explicit teaching for transfer of strategies acquired in programming as outlined above. The other class (the Independent Learning Group or ILG) was taught programming and mathematics in exactly the same way, but did not receive explicit training for strategy transfer. Each class had 18 students. Students in both the STG and the ILG took pre-tests and post-tests in mathematical problem-solving and general mathematical achievement (Progressive

Achievement Test, ACER, 1990). The strategy training took place in the middle terms of the year, but the post-test was not conducted until the end of the year, when students had finished all of the Lap-T tasks.

The ten items contained in the Maths Problem-solving pre and post-test had been selected after trials with students in the previous year. Some items were from *Profiles for Problem Solving* (Stacey, Groves, Bourke, & Doig 1993) and others were constructed by the researcher. These problems were considered to conform to three criteria: they readily showed evidence of use of problem-solving strategies when successful outcomes were achieved, they were complex enough to require planning, reflection and debugging skills and they could be resolved by at least two thirds of the trial group in the allotted time.

The problem-solving tests were analysed to assess student success in three main dimensions of mathematical problem-solving. These were obtaining correct answers, selecting appropriate strategies and articulating strategy choice. These dimensions were analogous to the dimensions of correct answers, method used and quality of explanation used in Profiles for Problem Solving. It was hypothesised that the STG would be better than the ILG in overall problem solving and in each of the dimensions above. Furthermore, it was hypothesised that improvement would mainly be evident in the students who were initially low That is, that students in the STG with relatively low initial mathematical skills, will show greater improvements in overall problem-solving performance than will their counterparts in the ILG. There will be no difference in the improvement of students with high initial mathematical skills. This was hypothesised because it was judged that able problem solvers probably already had knew and used useful strategies and had more developed metacognitive skills. Schoenfeld (1985) establishes that expert adult problem solvers have these characteristics: direct data on young children is not available.

Results

Table 1 shows the means and standard deviations on both pre-test (T^1) and post-test (T^2) for both groups of students. It shows that that the ILG made more improvement on the PAT Maths test than the STG and the STG made more on all three dimensions of problem solving. The pre-test differences between the STG and the ILG meant that regression analysis rather than first difference scores (i.e. pre-test minus post-test scores) is an appropriate technique for comparing improvements in the two classes.

Hypothesis1 states that the STG will improve more between the beginning and end of the year at finding correct solutions to problems than the ILG. In the regression equation which follows, the dependent variable is Obtaining correct answers at time 2 (OCA_2) and the independent (explanatory) variables are Obtaining correct answers at time 1 (OCA_1) and Group (STG = 1, ILG = 0). The independent variables were entered simultaneously, although it should be noted that stepwise entry (Obtaining correct answers 1 entered first) yielded an almost identical result. The value of R^2 is 0.45.

Equation 1
$$OCA_2 = 36.42 + 0.61\,OCA_1 + 6.53\,Group$$
$$(2.90) \quad (5.07) \quad (0.95)$$

This equation, which is similar in form to all subsequent equations, should be read in the following way. OCA_2 is first a function of a constant (the intercept term in the regression equation). Secondly, OCA_2 is positively related to OCA_1, the metric regression coefficient being 0.61. OCA_2 is also positively related to Group where the STG was coded 1 and the ILG coded 0. The coefficient of 6.53 for Group indicates that, controlling for scores at time 1, the STG improved by 6.53 points (on a 0-100 scale) more than the ILG.

If we treat the two Groups as populations this difference in improvement may be regarded

as 'real'. However, if we treat the two Groups as samples of Year 6 girls and so take notice of the t values (i.e., the standard errors of the estimates, which are shown in parentheses beneath the regression coefficients), we would infer that the difference in improvement between the two classes was not significant at the conventional 0.05 level. (It should be noted that t values larger than 2.0 are required for significance at this level).

Overall, it can be inferred that there is some confirmation of hypothesis 1, although it can also be argued that the null hypothesis of no difference between the Groups cannot be decisively rejected. This latter caveat essentially springs from having a small number of subjects in the population, as well as considerable variance of scores within both Groups.

Table 1

Means and Standard Deviation of STG and ILG on Problem Solving Dimensions and PAT Maths Test and Mean Improvement: Pretest (T^1) and Posttest (T^2)

Variable	Test T^1 (Feb) T^2 (Dec)	Means/Std Deviations STG (n = 18)	Means/Std Deviations ILG (n = 18)
OCA	T^1	45.6 (25.9)	39.1 (30.7)
OCA	T^2	58.2 (22.8)	47.8 (28.8)
Improvement		12.6	8.7
Strategy choice	T^1	47.2 (29.6)	44.1 (29.2)
Strategy choice	T^2	70.2 (18.4)	51.2 (32.8)
Improvement		23.0	7.1
Articulation	T^1	41.4 (19.5)	39.8 (24.9)
Articulation	T^2	58.9 (20.5)	51.0 (23.1)
Improvement		17.5	11.2
PAT Maths test	T^1	74.5 (18.10)	58.2 (24.3)
PAT Maths test	T^2	78.1 (21.7)	69.4 (24.1)
Improvement		3.6	11.2

Note. Abbreviations: OCA = Obtaining correct answers; Articulation = Strategy articulation

Hypothesis 2 predicts that the STG will show a greater improvement in selecting appropriate strategies than the ILG:

Equation 2

Strategy Choice$_2$ = 58.16 + 0.58 Strategy Choice$_1$ + 16.21 Group
(4.41)　　　(4.63)　　　　　　　　　(2.27)

Equation 2 confirms the hypothesis and indicates that the difference between the two groups of 16.21 points on the 0-100 scale was significant at the 0.05 level (t = 2.27). The value of R^2 is 0.44. The STG was making better strategic choices at the end of the year than the ILG. This indicates possible transfer of skills, so is a promising result.

Hypothesis 3 predicts that the STG will improve more than the ILG during the year in their ability to articulate the strategies they used to solve problems.

Equation 3

Articulation$_2$ = 44.10　0.58Articulation$_1$　8.10 Group
(4.17)　(4.54)　　　　　　(1.44)

This hypothesis is weakly confirmed, but with the coefficient for Group not being significant at the conventional 0.05 level (t = 1.44).

Hypothesis 4 also compares the two groups of students and is based on an *overall performance measure* (Performance$_1$ = Performance at time 1, and Performance$_2$ = Performance at time 2). This scores was obtained by giving a 50% weight to Obtaining correct answers, and 25% each to Strategy choice and Strategy articulation.

Equation 4

$$\text{Performance}_2 = 41.59 + 0.62\,\text{Performance}_1 + 10.65\,\text{Group}$$
$$\quad\quad\quad\quad (3.35)\quad\quad (5.42)\quad\quad\quad\quad\quad (1.59)$$

Equation 4 shows that the overall performance of the STG (controlling for performance at time 1) improved by 10.65 points on the 0-100 scale more than the performance of the ILG. Again, the result was not quite significant at the 0.05 level (t = 1.59). The value of R^2 is 0.50.

Table 2 shows the results of students with high and low PAT Maths scores at the pre-test. The low group contains students at stanines four or below and the high group contains students at stanines 8 and 9. The low students from the STG recorded a high average improvement in problem solving performance, while students in the ILG class improved only minimally. To assess statistical significance, Equation 4 was re-run just for students in the low initial maths scoring group. The metric regression coefficient of 25.3 for the variable 'Group' had a significance level of p = 0.029 so the hypothesis that initially low students improved more in the STG group was accepted.

Among students in the high group, the ILG students actually registered a greater improvement in problem-solving but a t-test indicated that this difference was not statistically significant at the 0.05 level. It is however possible to speculate that the strategy intervention was not useful in improving problem-solving skills for students with established mathematical skills. Students in the ILG actually made greater gains.

Table 2
Improvement in Overall Performance During the Year for High and Low Performers in STG and ILG)

STG groups	STG Performance			ILG groups	ILG Performance		
	T^1	T^2	Improv.		T^1	T^2	Improv.
Low (n = 3)	26.6	53.1	26.5	Low (n = 7)	30.0	30.4	0.4
High (n = 5)	63.6	67.3	3.7	High (n = 4)	62.7	71.4	8.7

The results are of course based on small numbers of students who comprised the 'low' and 'high' achievers in both classes. However the results do indicate the strong possibility that students who are overall weak performers in mathematics might be helped to better problem solving performance with strategic training.

Discussion

Considering the results obtained for hypothesis 5 in conjunction with the overall positive findings for the effects of strategy training on students' mathematical problem solving, it is reasonable to say that the greatest benefit was obtained by lower maths performers. From a teaching standpoint it is significant that students of low mathematical ability registered much greater improvements in problem-solving as a result of strategy training than did their counterparts in the ILG, who did not receive strategy training. It appears that explicit strategy training in a variety of contexts in the area of problem-solving may well be a promising method of overcoming deficits in mathematical problem-solving performance, especially for students with a low level of general maths skills.

However, in general terms, the differences found in the greater ability of the STG to choose and articulate strategies are particularly encouraging, because it is reasonable to suppose that students who are conscious of the strategies they are using will, in general, be better able to transfer their skills to a wide range of analogous tasks. This may well be the most important factor in achieving far-transfer to mathematical problem solving for the STG.

More broadly the research addressed in a small way the issue of how teachers can utilise time effectively by ensuring that learning is transferred and students are challenged. Programming tasks, in the form that they were presented to the students, were successful in engaging, motivating and challenging a range of students. Their subject matter ensured that students engaged in higher-order thinking processes as they researched, classified, processed and presented the solutions to their programming tasks. As such they were universally successful as one way of reinforcing content knowledge associated with a subject area or areas and contributing to the development of a range of important learning skills. Such outcomes are minimal expectations for teachers wishing to model learning with Logo-based software around the Lap-T(asks).

Programming has been shown in this study to improve mathematical problem-solving processes in some upper primary students with specific teacher mediation to systematically reinforce cognitive and metacognitive skills and strategies associated with programming and problem-solving. Teacher mediation which encourages application to knowing the problem, persistence with the task and constant editing and reflection seems to make a contribution to bridging the gap for students, resulting in transfer of useful problem-solving aspects of programming into a maths problem-solving context. This research also found that some levels of metacognitive awareness, namely being able to verbalise relational features of analogous contexts was not evident despite the apparent strategy transfer between two contexts (Walta, 1999). Further research into the relationship between the application of useful strategies to mathematical problem solving tasks and a student's ability to verbalise these understandings would be interesting.

References

ACER (1990). Progressive Achievement Tests in Mathematics (PAT). Melbourne: Australian Council for Educational Research.

Clements, D.H. (1987). Longitudinal study of the effects of Logo programming on cognitive abilities and achievement. *Journal of Educational Computing Research, 3*(1), 73-94.

Lehrer, R. (1989). Computer-assisted strategic instruction. In C. McCormick, G. Miller, & M. Pressley (Eds.) *Cognitive Strategy Research* (pp. 303-320). New York: Springer-Verlag.

Lester, F.K. (1994). Musings about mathematical problem solving research 1970-1994. *Journal for Research in Mathematics Education, 25*(6), 660-675.

MicroWorlds (LCSI, Logo Computer Systems Inc., 1993)

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. Brighton: Harvester Press.

Papert, S. (1994). *The children's machine*. New York: Harvester Wheatsheaf.

Polya, G. (1973). *How to Solve It*. Princeton, New Jersey: Princeton University Press.

Rowe, H.A.H. (1993). *Learning with personal computers: Issues, observations and perspectives*. Melbourne: Australian Council for Educational Research.

Schoenfeld, A.H. (1985) *Mathematical problem solving*. Orlando: Lawrence Academic Press

Short, E.J., & Weissberg-Benchell, J.A. (1989). The triple alliance for learning: cognition, metacognition and motivation. In C.B. McCormick, G. Millar, & M. Pressley (Eds.) *Cognitive strategy research* (pp. 6-23). New York: Springer-Verlag.

Stacey, K., Groves, S., Bourke, S. & Doig, B. (1993) *Profiles of problem solving*. Melbourne: Australian Council for Educational Research.

Swan, K., (1991). Programming objects to think with: Logo and the teaching and learning of problem solving. *Journal of Educational Computing Research, 7*(1), 89-98.

Walta, C. J. (1999) *Microworlds programming, mathematical problem solving and teaching for transfer: A year-long classroom study* Unpublished Ed. D. thesis, University of Melbourne.