

## The reflective abilities of expert and novice learners in computer programming

**Betty Breed**

Faculty of Educational Sciences, North-West University  
sneab@puknet.puk.ac.za

### **Abstract:**

I report the results of a pilot study done to determine the differences in reflective abilities between expert and novice learners in computer programming. Although the pilot study was primarily aimed at testing the reliability of the questionnaire, the research provided a number of interesting results about the degree of reflective thinking in which the learners engaged, before beginning to write the program, while working on it, and after finishing the task. The most outstanding result was that neither of the groups tended to do any reflection after they had completed the task.

### **Abstrak:**

*My doel is om die resultate weer te gee van 'n loodsstudie wat gedoen is in 'n ondersoek om te bepaal hoe die effektiewe leerder en die oneffektiewe leerder in rekenaarprogrammering verskil ten opsigte van hul reflektiewe denke tydens 'n programmeringsaktiwiteit. Alhoewel die loodsstudie primêr bedoel was om die betroubaarheid van die vraelys te toets, het die ondersoek 'n aantal interessante resultate gelewer rakende die mate waarin leerders by reflektiewe denke betrokke is voordat hulle begin om 'n rekenaarprogram te skryf, terwyl hulle besig is met die program, en nadat die taak voltooi is. Die mees uitstaande resultaat is dat beide groepe baie min refleksie gedoen het nadat die taak voltooi was.*

### **Tshoboloko:**

*Maikaelelo a athikele e ke go neela gape dipholo tsa thuto ya phaeloto e e dirilweng mo patlisisong ya go tlamamisa gore mo go moithuit yo o nonofileng le yo o sa nonofang mo lenaneng la khomputara pharologanyo ke efe mabapi le kakanyo e e nonofileng mo go bona ka nako ya tiragatso ya lenane. Le fa maikaelelo-potlana a thuto ya phaeloto e ne e le go tlhatlhoba boikaneyo jwa dipotsololo, dipatlisiso di neetse dipholo tse mmalwa tse di kgatlang mabapi le se baithuti mo kakanyong e e nonofileng ba tsayang karolo pele ba simolola go kwala lenane la khomputara, fa ba sa tshwaregile mo lenaneng le fa morago fa tiro e feditswe. Pholo e e bonagalang go feta ke gore mo ditlhopheng tsoo pedi go dirwa nonofo e nnye thata morago ga gore tiro e khutle.*

### **Background and problem statement**

Computer programming is the component of Computer Science that demands a great deal from learners in terms of both cognitive and metacognitive skills. Cognitive abilities that play an important role include critical and creative thinking, and problem-solving skills. For example, learners must be able to use unstructured data, devise the most effective solution for the problem, and then encode it in a computer language to get logical and useful results. Computer programming also requires of learners to plan continuously, monitor their progress, and evaluate their efforts, which implies typical metacognitive skills (Gourgey, 2001:18).

The degree to which learners possess these necessary or required cognitive and metacognitive skills influences their effectiveness in computer programming to a large extent and consequently determines their performance as programmers. According to Ertmer and Newby's (1996:4) view of expert learners, those learners who have no problems with computer programming, who reach their goals easily, and who attain outstanding learning outcomes, can be considered expert learners or programmers. On the other hand, novice learners in computer programming are considered to be those learners who find computer programming difficult, do not reach their goals, and often do not have any goals at all.

An expert is usually considered as someone who knows a good deal more about a specific topic than most other people do. For many years this view also prevailed in education regarding the expert learner (Weinstein & Van Mater Stone, 1993:31). However, research (Ertmer & Newby, 1996:4; Weinstein & Van Mater Stone, 1993:32) has revealed that the difference between an expert learner and a novice learner entails more than the quantity of knowledge the learner possesses. There are also qualitative differences between expert and novice learners. Weinstein and Van Mater Stone (1993:32) identify the following qualities, among others, of expert learners: their knowledge is better organised and integrated than that of novice learners; they possess better learning strategies and methods than novice learners; they are better motivated than novice learners; and they are more self-regulated than novice learners.

Because expert learners are more self-regulated they more often use metacognitive skills such as self-monitoring of their own thoughts and learning. They know when to reflect and how to evaluate their progress (Lieberman & Linn, 1991). They learn from their mistakes, make adjustments if necessary, and are able to apply successful solutions and strategies in different problem situations.

The importance for learners to continuously reflect while doing a programming activity is supported by research on programming (Lehrer, Lee & Jeong, 1999:248; Van Mierenboer, 1990:46). The interactive nature of modern programming languages can easily lead to the perception that planned and reasoned action is not important (Lehrer, Lee & Jeong, 1999:250). Learners often rely on the programming language itself to help them solve a problem, without themselves planning a solution beforehand and then using a computer language to implement the solution. This approach usually leads to the development of bad programming techniques, resulting in unstructured programs or the learner is rendered unable to solve the problem.

Expert learners approach problem-solving in three phases: (a) the phase in which the solution is planned, (b) the actual problem-solving process, and (c) evaluation of the solution (Artzt & Armour-Thomas, 2001:127). Reflection in problem-solving is necessary because it allows learners to consider the planning that was done before the task was started, the evaluations and changes that occurred while the task was being executed, and evaluations that were done after the task was completed (Ertmer & Newby, 1996:14). Reflection therefore enables learners to come to conclusions regarding the effectiveness of their planning and of the strategies used, so that possible adjustments or changes can be made for future learning activities.

In essence computer programming is nothing but problem-solving and it can also be approached in these three phases. The first phase is the period before the program is coded into a computer language, the second phase is the period while the program is being coded, and the last phase is the period after the program has been completed. This leads to the issue of the role of reflective thinking in each of these respective phases. By analysing the reflective abilities of expert and novice learners in computer programming in each of the three phases, possible conclusions could be made regarding the extent to which continuous self-monitoring and self-evaluation are necessary in each phase to enhance the effectiveness of learners as program-

mers. This knowledge can provide guidelines for educators on how to adjust their teaching approaches and strategies in order to develop learners' reflective thinking. Based on the above arguments, the research revolved around the following questions:

- How do expert and novice learners in computer programming differ regarding their reflective abilities in each of the following phases:
  - before starting the computer programming?
  - while performing the programming task?
  - after finishing the computer programming?
- How should learners manage their thoughts in order to be effective programmers?
- What role should the educator play in the development of reflective thinking of learners in computer programming?

Two methods of research were used. Firstly, a literature study was undertaken to look into the role of reflection in effective learning and, secondly, an empirical study regarding the reflective abilities of expert and novice learners in computer programming was carried out.

I now report on the results of a pilot project done to determine the differences in their reflective abilities, between expert and novice learners in computer programming, in each of three phases: before beginning to write the program, while working on it, and after finishing the task.

### **The role of reflection in effective learning**

In the previous two to three decades much research was done on the role of metacognition in effective learning. Most researchers distinguished two fundamental aspects of metacognition, namely, metacognitive knowledge and metacognitive control (Hartman, 2001:34; Schraw, 2001:3; Everson & Tobias, 2001:69; Gourgey, 2001:18). Metacognitive knowledge refers to learners' knowledge of themselves, of the learning task and of specific learning strategies (Schraw, 2001:4), whilst metacognitive control relates to planning, monitoring and evaluation of the learning process (Hartman, 2001:35).

The work of Zimmerman (1990; 2000) and that of Weinstein and Van Mater Stone (1993) also emphasise the role of metacognition in effective learning. According to Zimmerman's (1990:4) view of self-regulated learning, expert learners are actively involved in their own learning through metacognitive, motivational and behavioural processes in order to reach their goals. The metacognitive control processes used by expert learners in different stages of the learning process include planning, self-monitoring and self-evaluation. Zimmerman (1990:8) attributes the difference between expert and novice learners to the fact that expert learners use self-regulating learning strategies to a greater extent, whilst novice learners usually do only as instructed by their teacher.

Weinstein and Van Mater Stone (1993:32) view effective learning from the angle of metacognitive knowledge and describe expert learners as learners who use metacognitive knowledge of themselves, of the task, of specific learning strategies, and of the content, to monitor the success of the strategies they use in achieving their desired goals. Expert learners continuously have to make decisions regarding the strategies they use to determine whether or not a strategy should be continued, changed, or suspended.

The work of Weinstein and Van Mater Stone (1993) and that of Zimmerman (1990) form the base of Ertmer and Newby's (1996) view of effective learning. However, Ertmer and Newby (1996) emphasise the critical role of the learners' thoughts (reflection) in effective learning. According to them (1996:14) reflection is the important link (a) between the components of metacognitive control, namely, planning, monitoring, and evaluation, and (b) between metacognitive knowledge and metacognitive control, which enables learners to connect their know-

ledge of themselves as learners, of the task, and of specific learning strategies to the current learning activity. This corresponds with Dewey's (1933) description of reflection as being the ability to stand back from (learning) experiences, to look at them and to try and understand them in the context of previous experiences and present knowledge (Kusnic & Finley, 1993:12; Lehrer, Lee & Jeong, 1999:246).

Ertmer and Newby (1996) base their view of the dual role of reflection in effective learning on the fact that reflection can occur on different levels of the learning process. The first level of reflection forms a feedback loop within a specific learning activity. It means that learners must execute the activity and at the same time reflect on what they are doing (Lehrer, Lee & Jeong, 1999:247). During the phases of a learning activity, namely, planning beforehand, application of strategies during the activity, and evaluation afterwards, expert learners will reflect on content and process continuously to enable them to utilise their existing knowledge structures, decide on appropriate strategies, determine their progress, and make adjustments or changes, if necessary, to reach their goals. Adjustments and changes are necessary because of continual changes in personal, behavioural, and environmental factors during the learning process (Zimmerman, 2000:14).

The second form of reflection is a feedback loop outside the specific activity. Expert learners reflect on previous experiences to orientate themselves for current or future thought and action (Dewey, 1933:12; Ertmer & Newby, 1996:16). In this way they connect their existing knowledge and metacognitive knowledge with their current learning activities and it becomes easier for them to integrate and organise their knowledge.

Reflective thinking is a generic construct and not inherent in particular subjects (e.g. computer programming) (Kember *et al.*, 2000:390). Consequently educationists have for some time been advocating a model of instruction aimed at equipping learners to work reflectively so that they will be able to handle problems in any subject. Learners who do not use reflective thinking in problem-solving tend to be less systematic in collecting information and data, spend less time on planning the solution beforehand, and usually do not consider alternative methods (Van Mierenboer, 1990:45). In general, these learners reach lower learning outcomes than learners who function reflectively while solving problems. Learners who use reflective strategies in computer programming usually plan properly in advance, divide a problem into sub-problems if possible, collect data systematically and accurately, consider more than one solution for the problem, and only then start coding the program lines (Van Mierenboer, 1990:46).

Reflection is necessary because it allows learners to consider the planning that was done before the task was started, the evaluations and changes that occurred while the task was being executed, and evaluations that were done after the task had been completed (Ertmer & Newby, 1996:14). Reflection therefore enables learners to come to conclusions regarding the effectiveness of their planning and of strategies that were used, so that possible adjustments or changes can be made for future learning activities. In other words, reflection leads learners to strategic and self-regulated learning.

## **Pilot study**

### **Questionnaire**

Based on the literature review, a questionnaire was designed. It was specifically aimed at determining the difference regarding reflective thinking between expert and novice learners in computer programming. The questionnaire consisted of three sections:

- Section A contained nine questions requiring biographical information of the learner.
- Section B consisted of 19 items with the specific goal of determining the extent to which

the learner uses reflective thinking when performing a computer programming activity. This Section was based on the principle that problem-solving can be viewed in three phases, namely, planning before starting to code the program, application of strategies while executing the task, and evaluation after the activity has been completed (Fortunato *et al.*, 1991:39; Ertmer & Newby, 1996:14). These items were divided into three phases:

- four items on reflective thinking before the learner started coding the computer program
- nine items on reflective thinking while the learner was coding the computer program
- seven items on reflective thinking after the learner had completed the computer programme.
- Section C consisted only of two general items on the programming that had been completed by the learner. The aim of these items was to determine
  - whether the learner had relied on the error messages of the compiler to solve the problem, and
  - whether the learner could connect this problem to any previous experience, or not.

### Participants

The questionnaire will be used for research involving about 350 learners in Grade 11 in the North-West Province of South Africa.

In the pilot study reported on here, second-year education students majoring in Computer Science at the Potchefstroom University were used. This class consisted of 30 students. For the purposes of the pilot study these 30 students formed the population, and the whole population was used in the study.

### Procedure

On a given day these 30 students received a problem that had to be solved by means of a computer program. All 30 attempts were evaluated according to a preset memorandum. The five learners with the highest marks and the five learners with the lowest marks were regarded as the "expert" and "novice" groups, respectively.

On a second occasion the 30 students again had to write a computer program to solve a problem. Afterwards they had to complete the questionnaire. The numbers on the questionnaires distinguished the experts from the novices, and from the rest of the class.

The responses of the students were processed by using the SAS computer program (SAS Institute, 1990) to calculate means, standard deviations, and other relevant statistics. The following statistical techniques and methods were used to analyse the data:

- Determining the reliability of the items in each of the phases in Section B of the questionnaire by computing the Cronbach alpha values (Anastasi, 1988:124).
- Determining the effect sizes of differences between experts and novices by calculating Cohen's *d* values (Cohen, 1988:20).

It is important to note that the 30 students used in the pilot study were not selected randomly. Because the whole population was used in the pilot study the *p* value could not be used.

### Results

#### Biographical information

The frequencies of the questions on the respondents' biographical information were not used

because each group in the pilot study consisted of five learners only.

#### Computation of the Cronbach alpha values for Section B

To determine the reliability of the items in each of the three phases in Section B of the measuring instrument, in the context in which it was used, the Cronbach alpha reliability coefficient was used. This value was calculated for each of the three phases, namely, Before, While, and After. The values are given in Table 1:

**Table 1** Determination of the reliability of the items grouped together in the phases *Before, While, and After*

Reflection	Cronbach alpha value
<i>Before</i> the learner started coding the computer program	0.67
<i>While</i> the learner was coding the computer program	0.54
<i>After</i> the learner had completed coding the computer program	0.73

The phases all reached acceptable values of larger than 0.5, indicating that the items in each of the three phases had been grouped together in a reliable way and that the items in each of the phases of Section B could be interpreted as reliable for the population for which they were used.

#### Computation of Cohen's $d$ values to determine effect sizes of differences between the reflective thoughts of experts and novices in each of the three phases of Section B

Cohen's effect sizes were then calculated to determine the practical significance of the differences. All items in each of the phases were grouped together for calculation of the mean for each phase.

**Table 2** Determination of Cohen's effect sizes

Phase	Group of students	Mean	SD	$d$ value	Effect size of difference
Before	Novices	2.7	0.4108	2.56	Large
	Experts	3.75	0.25		
While	Novices	3.025	0.163	1.26	Large
	Experts	2.4	0.4953		
After	Novices	1.4857	0.4802	0.83	Large
	Experts	1.8857	0.2928		

Cohen's  $d$  values indicate the practical significance of differences as follows:

0.2: small effect size; 0.5: medium effect size; 0.8: large effect size

From Table 2 it may be concluded that the  $d$  value is bigger than 0.8 in each of the three phases, indicating a practically significant difference between the experts and novices in each of the three phases. When comparing the means of the two groups in each of the phases, the following observations can be made:

1. There is a practically significant difference between expert learners and novice learners regarding the extent of reflection before they start coding the computer program. In this phase experts reflect far more than novices before starting to code the computer program. The individual items with a practically significant difference between experts and novices in this phase indicate that

- experts are more inclined than novices to read through the problem once before starting to develop a computer program ( $d = 2.56$ );
  - experts are more inclined than novices to try recalling experience in developing a similar program before ( $d = 1.79$ );
  - experts are more inclined than novices to divide a problem into sub-problems, recalling previous experience, techniques and strategies used to solve the similar problems ( $d = 1.83$ ).
2. There is a practically significant difference between expert learners and novice learners concerning the extent of reflection while they are coding the computer program. In this phase experts reflect far less than novices do. The individual items in this phase with a practically significant difference between experts and novices indicate that
- experts are less inclined than novices to stop and make a decision to divide the program into sub-parts ( $d = 2.19$ );
  - experts are less inclined than novices to stop and think about a technique/procedure that they have used in another program ( $d = 0.89$ );
  - experts do not frequently find it necessary to go back to previous steps in order to rectify a mistake ( $d = 0.89$ ).
3. There is a practically significant difference between expert learners and novice learners concerning the extent of reflection after the computer program has been completed. In this phase experts reflect a little more than novices do. The individual items in this phase with a practically significant difference between experts and novices indicate that
- experts are more inclined than novices to read the question again to determine if they had followed the instructions of the problem ( $d = 1.79$ );
  - experts are more inclined than novices to double-check for correct techniques and/or procedures ( $d = 2.56$ );
  - experts are more inclined than novices to double-check their output and solutions to see if these make sense ( $d = 1.05$ ).

Computation of significant differences between experts and novices in each of the two questions of Section C

Cohen's effect sizes were then calculated to determine the practical significance of the differences. One of the questions revealed a practically significant difference between experts and novices, indicating that in this specific case, experts were more inclined than novices to connect the task to previous experience ( $d = 1.8$ ).

### **Discussion**

The result showing that the experts in the pilot study reflected more than the novices in the pilot study before starting to work on a computer program is an indication that experts do proper planning before starting to code the program lines. They divide a problem into sub-problems if possible, try to connect each of the sub-problems with a previous experience, and consider techniques and procedures they have used before. What is quite surprising is that some experts have reported reading the problem only once before starting to work on the solution. This could be an indication that they read the problem attentively the first time to make sure that they understand it from the very start.

The fact that the novices in the pilot study were more inclined than the experts in the pilot study to be engaged in reflective thinking while working on the programming task could be because they did not plan properly in the first instance. The results show that it is only in this

phase that the novices start considering dividing the problem into sub-problems and try to connect these sub-problems with previous experiences. This could be because they are forced to do so by necessity once they start struggling with the program. Experts, on the other hand, plan ahead and decide on what strategies and/or procedures to use before even starting the task. According to Kember *et al.* (2000:384) the fact that experts are able to connect a problem (or sub-problem) to a previous similar problem could lead to habitual action so that reflection becomes less necessary.

Although a practically significant difference between the experts and the novices in the pilot study was found for the phase after they had completed the program, it must be noted that in both cases the means were low. The novices had a mean of 1.48 and the experts a mean of 1.88. Keeping in mind that the average was 2.5, these means indicate that in this specific population the novices as well as the experts do not tend to do much (if any) reflection after the programming task has been completed.

### Concluding remark

There are still several unanswered questions arising from this pilot study. Although the pilot study was primarily meant to test the reliability of the questionnaire, it produced interesting results regarding the differences in reflective thinking between expert learners and novice learners in computer programming. It remains to be seen if the same tendencies will emerge once the final research has been done with the 350 Grade 11 learners doing computer programming.

### References

- ANASTASI A 1988. *Psychological testing*. 6th edn. New York : Macmillan.
- ARTZT AF & ARMOUR-THOMAS E 2001. Mathematics teaching as problem solving: A framework for studying teacher metacognition underlying instructional practice in mathematics. In: Hartman HJ (ed.). *Metacognition in learning and instruction. Theory, research and practice*. Dordrecht : Kluwer Academic Publishers.
- COHEN J 1988. *Statistical power analysis for behavioural sciences*. 2nd edn. Hillsdale, NJ: Erlbaum.
- DEWEY J 1933. *How we think*. Boston, MA: D.C. Heath & Co. Publishers.
- ERTMER PA & NEWBY TJ 1996. The expert learner. Strategic, self-regulated and reflective. *Instructional science*, 24:1-24.
- EVERSON HT & TOBIAS S 2001. The ability to estimate knowledge and performance in a college: A metacognitive analysis. In: Hartman HJ (ed.). *Metacognition in learning and instruction. Theory, research and practice*. Dordrecht : Kluwer Academic Publishers.
- FORTUNATO I, HECHT D, TITTLE CK & ALVAREZ L 1991. Metacognition and problem solving. *Arithmetic teacher*, 38-40.
- GOURGEY AF 2001. Metacognition in basic skills instruction. In: Hartman HJ (ed.). *Metacognition in learning and instruction. Theory, research and practice*. Dordrecht: Kluwer Academic Publishers.
- HARTMAN HJ 2001. Developing students' metacognitive knowledge and skills. In: Hartman HJ (ed.). *Metacognition in learning and instruction. Theory, research and practice*. Dordrecht : Kluwer Academic Publishers.
- KEMBER D, LEUNG DYP, JONES A, LOKE AY, MCKAY J, SINCLAIR K, TSE H, WEBB C, YUET WONG FK, WONG M & YEUNG E 2000. Development of a questionnaire to measure the level of reflective thinking. *Assessment & evaluation in higher education*, 25:381-396.
- KUSNIC E & FINLEY ML 1993. Student self-evaluation: An introduction and rationale. In: MacGREGOR J (ed.). *Student self-evaluation: Fostering reflective learning*, 56. San Francisco : Jossey Bass.
- LEHRER R, LEE M & JEONG A 1999. Reflective reaching of Logo. *The journal of the learning sciences*, 8:245-289.
- LIEBERMAN DA & LINN MC 1991. Learning to learn revisited: Computers and the development of self-directed learning skills. *Journal of research on computing in education*, 23:373-396. [In EBSCOHost: Academic Search Premier, Full text: <http://search.epnet.com>] [ Date of access: 10 September 2003].

- SAS INSTITUTE Inc. 1990. *SAS/STAT Procedure's Guide, Version 6*, 3rd edn. Cary, NC: SAS Institute Inc.
- SCHRAW G 2001. Promoting general metacognitive awareness. In: Hartman HJ (ed.). *Metacognition in learning and instruction. Theory, research and practice*. Dordrecht : Kluwer Academic Publishers.
- VAN MERRIENBOER JJG 1990. Instructional strategies for teaching computer programming: Interactions with the cognitive style reflection-impulsivity. *Journal of research on computing in education*, 23:45-54.
- WEINSTEIN CE & VAN MATER STONE G 1993. Broadening our conception of general education: The self-regulated learner. In: *New directions for community colleges*, 81:31-39. San Francisco: Jossey Bass.
- ZIMMERMAN BJ 1990. Self-regulated learning and academic achievement: An overview. *Educational psychologist*, 25:3-17.
- ZIMMERMAN BJ 2000. Attaining self-regulation. A social cognitive perspective. In: Boekaerts M, Pintrich PR & Zeidner M (eds). *Handbook of self-regulation*. New York: Academic Press.