African Journal of Research in Mathematics, Science and Technology Education, 2013
Vol. 17, No. 3, 206–219, http://dx.doi.org/10.1080/10288457.2013.839154

Views of the Use of Self-directed Metacognitive Questioning during Pair Programming in Economically Deprived Rural Schools

Betty Breed[1*], Elsa Mentz[1], Marietjie Havenga[1], Irene Govender[2], Desmond Govender[2], Frank Dignum[3] and Virginia Dignum[4]

[1] *North-West University, Potchefstroom, South Africa*
[2] *University of KwaZulu-Natal, Durban, South Africa*
[3] *Utrecht University, Netherlands*
[4] *Delft University of Technology, Netherlands*
**Corresponding author. Betty Breed, North-West University, Potchefstroom, South Africa. Email: Betty.Breed@nwu.ac.za*

The research reported in this article formed part of an internationally funded project about the empowerment of Information Technology (IT) teachers in economically deprived rural schools in the North-West and KwaZulu-Natal provinces in South Africa. The current paper focused on the use of self-directed metacognitive (SDM) questioning in a pair programming context. The study sample consisted of five IT teachers and 99 Grade 10 IT learners at five schools. The teachers were trained to implement pair programming and to guide learners in the application of metacognitive regulation while doing pair programming. The learners used SDM questions during their subsequent pair programming tasks. Data-gathering was done through interviews with the teachers regarding their views on the use of the SDM questions, and the learners' journals with their views on how they experienced the SDM questions to direct their thinking during execution of pair programming tasks. The results indicated that the teachers viewed the implementation of the SDM question difficult and time-consuming, and that they experienced the learners to be either reluctant or unwilling to engage in SDM questioning. However, the results of the learners' journals indicated that the learners experienced the SDM questions to be helpful in directing their thinking during pair programming tasks.

Keywords: pair programming; metacognitive skills; metacognitive regulation; problem solving; self-directed questioning

Introduction

Purposeful teaching and learning of metacognitive skills, especially in problem-solving contexts, promotes learners' ability to think about how and why they work in specific ways (Williams & Upchurch, 2001a). Research done by Mevarech (1999), Kramarski, Mevarech, and Lieberman (2001), Kramarski, Mevarech, and Arami (2002), Kramarski (2004), and Mevarech and Fridkin (2006) yielded positive effects of purposeful teaching and learning of metacognitive skills on mathematical problem solving, for both individual learners and learners working in groups. The positive outcomes of metacognitive instruction in mathematics while learners are working in groups (Kramarski, 2004) emphasise the importance of purposeful development of learners' metacognitive skills during pair programming.

Pair programming is currently often used in educational contexts to enhance learning in computer programming with the goal of solving problems using a programming language.

Little research on the explicit teaching and learning of metacognitive skills during pair programming has been done. In addition, the relevant previous research of Breed (2010) dealt with learners in affluent suburban schools in South Africa, while the teachers and learners involved in this project hail from economically deprived rural schools. These rural schools generally lack experienced, skilled teachers (Nonyane & Mlitwa, 2008), especially in Information Technology (IT) (Mentz et al., 2012). The medium of instruction in these schools is English, which very often causes difficulties for teachers and learners whose home language is not English.

The study reported in this paper aimed at investigating the views of teachers on the use of self-directed metacognitive (SDM) questions, and the learners' experiences in using the SDM questions to direct their thinking during the execution of pair programming tasks.

Conceptual-theoretical Framework

This section firstly briefly discusses pair programming as a teaching-learning strategy, highlighting the metacognitive involvement of the members of the pair. Following is an argument for the explicit development of metacognitive skills during pair programming. The section ends with an explanation of the development of the SDM questions that were used in the intervention phase of the research.

Pair Programming

Pair programming involves two programmers working together at the same computer on the same design, algorithm, code and test (Williams & Upchurch, 2001a). Research on pair programming as a teaching-learning strategy in acquiring programming skills found pair programming to be highly advantageous in the educational context (Katira et al., 2004), especially with regard to the resulting program quality. During pair programming one of the members of the pair acts as the 'driver' who is responsible for typing on the computer or coding a design (Nicolescu & Plummer, 2003). The other member acts as the 'navigator' who is constantly observing the driver's work, identifying problems, asking questions, making proposals to improve the program design or code (Berenson, Slaten, Williams, & Ho, 2004), suggesting alternative strategies and using relevant resources (Bipp, Lepper, & Schmedding, 2008). All these processes require metacognitive thinking. In their collaborative effort to complete a programming task, the driver and the navigator have to apply metacognitive activities when planning and designing the algorithm before coding, and afterwards when evaluating the process and product.

Explicit Development of Metacognitive Skills

Successful learners have, in addition to cognitive knowledge and skills, the ability to implement metacognitive skills. These skills include the ability to plan a strategy, monitor a process and its progress, make adjustments when necessary and reflect on the whole process with the aim to evaluate and improve their product (Williams & Upchurch, 2001b). According to Joseph (2009), most learners need to be taught and encouraged to develop these skills. Effective development of metacognitive skills requires that teachers themselves function metacognitively. Teachers who are aware of their own metacognitive functioning can meaningfully contribute to the development of learners' metacognitive skills (Sternberg, 2001). Such teachers realise the influence of their methods on their learners' metacognitive awareness (Paris & Paris, 2001) and capacity to engage in effective learning (Anderson & Nashon, 2007).

The development of the metacognitive skills of learners working in groups influences both their knowledge and the contribution that each member of the group can make to reach the learning outcomes (Chalmers & Nason, 2003). Explicit instruction of metacognitive skills enhances a group's ability to consider, suggest and compare different appropriate strategies to solve a problem, and to analyse the advantages and disadvantages of each strategy before applying it (Kramarski et al., 2002). Considering these aspects of the use of metacognition in group work necessitates the development of metacognitive skills of learners programming in pairs, albeit that the group now only has two members.

SDM Questioning during Pair Programming

Previous research in which a metacognitive teaching-learning strategy for pair programmers was developed (Breed, 2010) produced a set of SDM questions that a programming pair had to answer for themselves during the execution of a pair programming task. The development of these questions was based on a comprehensive literature study on metacognition during problem solving, and existing examples of metacognitive teaching-learning strategies. From the literature three generally accepted elements of metacognitive regulation during the execution of a task were identified: planning, monitoring, and evaluating (Veenman & Spaans, 2005; Williams & Upchurch, 2001b), with reflection as link between these elements. These could be related to the three phases generally associated with writing a computer program, namely the pre-coding, intra-coding and post-coding phases. The pre-coding phase of a programming task involves analysing the problem and designing an algorithm to solve it, the intra-coding phase concerns the actual coding of the planned solution and making adjustments if necessary, and the post-coding phase concerns evaluating the product and the process of the programming task. Figure 1 shows the metacognitive aspects relevant in each of the three metacognitive elements.

The SDM questions were then developed with the purpose to guide learners to execute the relevant metacognitive activities in each of these phases (see Appendix). The results of the previous research by Breed (2010) indicated that using the SDM questions enhanced the metacognitive awareness of the participating learners programming in pairs in terms of their planning, monitoring and evaluation.

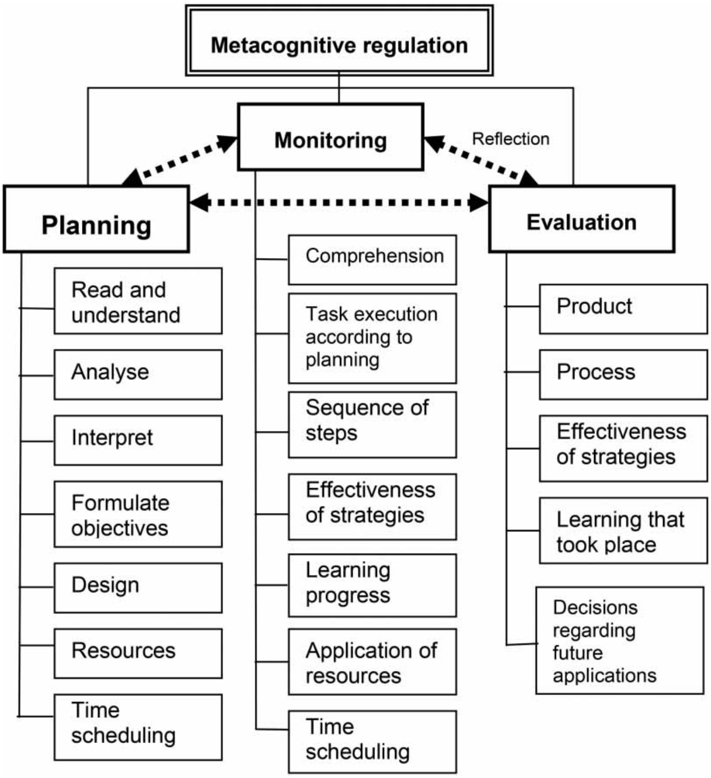

Figure 1: Schematic representation of the elements of metacognitive regulation

Research Methodology

The Intervention

The investigation focused on an intervention aiming at the empowerment of IT teachers in economically deprived schools in the North-West and KwaZulu-Natal provinces in South Africa. The main objectives of the intervention included the support of these IT teachers by developing their teaching competencies and pedagogical content knowledge in order to improve learners' programming skills. This was done by providing the teachers with knowledge and skills to (a) implement pair programming as a cooperative learning strategy in their classes, (b) teach and apply various problem-solving skills and strategies, and (c) develop the learners' metacognitive skills required for problem-solving and computer programming. The authors were responsible for training the teachers to implement each of these strategies in their IT classes at different stages of the intervention.

Before the intervention regarding the development of metacognitive skills, the learners already had experience with individual programming, as well as with the implementation of pair programming. At the start of this phase the teachers were trained in the implementation of the metacognitive approach to pair programming. Training of the teachers entailed a workshop on metacognition, with special emphasis on metacognitive regulation. The researcher modelled the use of metacognitive activities by using an applicable programming problem and by answering the SDM questions while executing the task. The teachers were then allowed a three-month period to implement this intervention in their pair programming classes, after which the teachers were interviewed. During the period of intervention, the learners had to complete a journal individually after completing a pair programming task, providing them with the opportunity to reflect individually on their own thinking processes and activities during the execution of the programming task.

The research reported below focused on the effect, within the intervention, of the use of SDM questioning to enhance the learners' metacognitive regulation in a pair programming context. The purpose of the investigation therefore was to determine the teachers' views on the use of SDM questions in the teaching of programming, and the learners' views on how they experienced the SDM questions to direct their thinking during execution of pair programming tasks.

Sampling

Since the research project focused on the empowerment of IT teachers in economically deprived rural schools in North-West and KwaZulu-Natal Provinces, only the rural schools situated in these provinces that were classified as economically deprived and that offered IT as a subject in Grade 10, formed the study population for the study. Only a small number of schools in both provinces met these criteria. Three schools per province were selected from those that did qualify. Finally, the IT teachers and Grade 10 IT learners at five of these schools participated in the study. The participating schools all experienced problems with their computers and a lack of resources. Very few learners had access to computers outside the classroom.

Because of the low numbers of learners taking IT as a school subject in South African secondary schools, all the Grade 10 IT learners at the particular schools were involved (N = 99). A total of five teachers took part in the investigation. Since all of the participating schools have English as medium of instruction and the learners have to write their examination papers in English as well, it was assumed that interviews and journals in English would not pose problems. However, it should be noted that none of the teachers or learners use English as home language.

Data Gathering and Research Instruments

- *Individual interviews*

The qualitative, semi-structured interviews conducted with the teachers had an informal conversational character. The participants were expected to answer a set of predetermined questions according to a prepared question schedule. When necessary, follow-up questions were used to clarify the participants' answers (Nieuwenhuis, 2007). The questions used during the interviews were meant to determine the teachers' views on the SDM questions and their experiences regarding the use thereof in their classes during pair programming.

- *Individual journals*

The learners' individual journals comprised of a file containing a summary of the metacognitive activities that had to be attended to in the different phases of a pair programming task, a number of lists with the SDM questions which the pairs collaboratively had to tick off each time they did a pair programming task, and a final part where the learners individually had to describe their thought processes and actions during the pre-coding, intra-coding and post-coding phases of each pair programming task that had just been completed. These individual entries were analysed to determine how the SDM questions directed the learners' thinking in each phase and helped them to execute the programming task.

Ethical Issues

Approval to conduct the research was obtained from the relevant ethical committee, education departments, principals, teachers, parents and learners. All participants had the right to withdraw from the research at any stage. They were assured that it would not be held against them if they withdrew. The schools and all participants were assured of anonymity with regard to data-gathering and dissemination of results.

Data Analysis

The data derived from the teachers' interviews were transcribed and analysed by using Atlas.ti™ and implementing the applicable principles of qualitative data analysis. The transcriptions were coded into themes and sub-themes, and then analysed for dominant themes. The aim of this analysis was to determine the teachers' views and experiences regarding the implementation of the SDM questions. From the data analysis, an overall impression of the teachers' own views and experiences, as well as their views on the learners' experiences was gained. The data derived from the learners' individual journals were analysed correspondingly, using the three phases *pre-coding, intra-coding* and *post-coding* as main themes. The learners' entries were coded into sub-themes, using a combination of *a priori* and emerging codes. The *a priori* codes used were in accordance with the metacognitive aspects identified for each phase (see Figure 1). Other emerging codes allowed for further understanding and explanation of the results.

Validity

For qualitative research (as reported here) reliability is not quite a relevant concern. The validity of the analysis was ascertained by having the transcribed interviews and data from the learners' individual journals triangulated, and the validity of the codes was increased by cross-checking these by another expert in the field of study.

Results

Teachers' Experiences and Views on the Use of the SDM Questions

The results of the interviews are presented below according to the themes that emerged, with quotes from the participant interviews. Where necessary, the authors inserted a word, or words, in round brackets to make the quote more understandable.

- *Own knowledge of metacognition*

Analysis of the interviews indicated that the notion of metacognition was totally new or just vaguely known to the teachers. The comment of one of the teachers during the interview says it all:

> It (the use of the metacognitive approach) was an eye-opener for me.

Another teacher commented:

> Even me, ... I don't take the time to really step back and see what is involved in thinking.

- *Time-consuming*

The teachers were in unison in their respective views that they had found the explicit teaching of metacognitive skills difficult and time-consuming.

> Teaching the concept (metacognition) is not easy. Yeah, it is taking time.
> ... the time is a major factor, because I need to complete the other topics, so I can't just spend more time on that specific strategy (metacognitive approach).

- *Advantages during problem solving*

Although the teachers emphasised the fact that the metacognitive approach was time-consuming, they were of the opinion that the approach could save time solving a programming problem when the learners have become experienced in executing the metacognitive activities.

> But if you think about it (solving the problem) step by step, you will find that you save a lot of time solving that problem.
> I hope that they will learn and it (the metacognitive approach) will become automatic.

One of the teachers recognised an advantage of the process:

> Because you will spend less time, coding and reflecting back is going to increase your knowledge as well.

- *Necessity of developing metacognitive regulation skills*

The teachers seemed to be convinced about the necessity of explicitly teaching metacognitive activities and developing the learners' metacognitive regulation skills.

> The first stage is very important. Pre-programming is ... gonna help you so that when you are coding you have a clear mind set what you are gonna do and what you are expecting at the end of the program.
> It is important to reflect on what was learned from solving a specific problem, like with every problem you must make sure that you caught it (understood what you did) and what is new (what you have learned from it).

One teacher experienced that some of his learners gained from using the strategy:

> Those who I could really see gained something out of the exercise are those who have generally been trying hard (to use the SDM questions).

- *Learners lack skills*

The teachers all indicated that the learners did not demonstrate any metacognitive regulation and that they lacked thinking skills.

> Yeah, so far learners are not even aware of these things (metacognition).
> They (the learners) do not understand that they are supposed to reflect when they are thinking.

One teacher later added:

> I think they are not used to thinking that way.

- *Learners' unwillingness*

The teachers experienced that the learners were unable or unwilling to use the SDM questions to direct their thinking when they had a programming problem to solve. The learners just wanted to start with the coding of the program immediately without any planning. One teacher explained:

> I gave them all a piece of paper and I said: First put down on paper what you want to do and you must hand it in at the end of the session. But immediately when they start doing the task, not even one of them writes on the piece of paper.

Another teacher added:

> When you get a question, you think you already know what needs to be done and you just go straight to it (In this quote 'you' refers to a learner).

The third teacher commented:

> They just can't follow all these steps, the pre-coding, the intra-coding and all these things. They just want to do the programming on their own.

- *Learners view process as waste of time*

According to the teachers, the learners initially viewed the use of the SDM questions as wasting the time they had to complete a task. A teacher described his learners' attitude as:

> For the start, it seems like you are wasting time with them, they think you are delaying them to solve the problem,

and explained:

> When they are programming, they do not want to spend all this time on writing all these steps, (referring to the journal the learners had to complete about their use of the SDM questions).

Another teacher confirmed that the students were very reluctant to do the metacognitive activities, especially the evaluation of the program:

> This post-coding phase – the learners, when they are doing a program, normally after the program is running, they don't go back. So they think they are done, and once they are successful they think, why go back and check?

Learners' Views on How they Experienced the SDM Questions

The results of the analysis of the learners' individual journals are reported with regard to the three phases of programming and the aspects of metacognitive regulation relevant in each phase (see Figure 1).

Pre-coding phase

- *Reading and understanding*

The SDM questions focused most learners' attention on the importance of reading a problem, and analysing it to identify the most important information and to determine the expected outcomes. One learner described it as follows:

> The questions did direct my way of thinking and assisted me to execute the task in a manner that I first read through the problem to get an overall view.

- *Formulating objectives*

The SDM questions directed learners at formulating the goals to be reached with a task.

> ... because of the self-directed questions I think twice or three times what I should do before I jump to the keyboard. I also think about the outcomes of the program before it could be coded.
>
> The questions did direct my way of thinking and assisted me to ... formulate the goals to be reached.

- *Reflecting on previous knowledge*

A number of learners indicated that the SDM questions had led them to reflect on aspects of the problem that were familiar to them and on their previous knowledge. A quote to illustrate this reads:

> The questions did direct my way of thinking and assisted me to ... reflect on previous knowledge.

- *Using available resources*

Learners indicated that they were directed to make use of available resources.

> Consulting available resources is a great technique, and using previous activities.
>
> We reflected back on our previous notes and Delphi code that is in the textbooks as guidelines to help us with an idea of how we can go about solving the problem and attaining the expected outcome.

- *Designing an algorithm*

Very few learners indicated that they actually designed an algorithm before coding, but one of those who did seemingly experienced success when completing the program said:

> I tried to plan with different strategies. At first it didn't work; then I started writing an algorithm which eventually worked for me.

Another wrote:

> At first we didn't plan well for creating this program, but after designing an algorithm and checking the steps we need to undertake to do this program, everything went well.

There were learners who indicated that the SDM questions made them think deeper about appropriate strategies, procedures or methods to solve a problem.

> First I read the question, then analyse it and I think about it, what method I should use to solve this problem.

- *Planning time*

In only one of the journals that were returned, a comment was made in the pre-coding phase about time planning:

> We planned the program, discussed it further and agreed that we should spend at least 30–40 minutes on this.

Intra-coding Phase

- *Understanding*

The first of the SDM questions to be used during this phase prompted learners to consider regularly whether they still understood what they were doing. The results show that this did not get much attention. Two entries only bore evidence to this regard, but it seems as if this was forced by errors rather than metacognitive activity.

> During the coding process the program gave me numerous errors which clearly showed me that I didn't understand what was required.
>
> I was not sure if what I was doing was correct. I had to go back and check with the problem to see if I was doing the right things.

- *Still on right track*

Quite a number of entries referred to regular consideration if they were still on the right track, but it is not clear whether they had, while coding, really thought about how the steps fitted in or what should be done next.

> We continuously checked that we were on the right track by assessing our coding thus far.

- *Monitoring progress*

The SDM questions directed the learners to monitor their progress, identify and correct errors and consider whether the procedures used were suitable to solve the problem.

> Then while I am coding I also have to ask myself whether I am doing the correct thing and also analyse errors and correct them. I continuously ask myself whether I have chosen the correct and most suitable techniques.

- *Changed planning if necessary*

If necessary, the initial planning of the solution was changed, sometimes leading to changes of the solution strategies.

> As we went on we checked if we were on the right track, until we ran the program and a whole list of errors appeared! We decided to start again and carefully think this over and use a different approach.

Some used the approach of dividing the problem into smaller units.

> After it gave us a few problems we divided the problems into sub-parts to identify the errors.

- *Consulting resources*

When necessary, learners consulted resources or sought help.

> Some programs are hard for me; then I look for help in the textbooks and internet, or ask people.

As in the case of the pre-coding phase, time-management was not attended to in this phase. Not a single entry was made in this regard.

Post-coding Phase

- *Effectiveness of strategies*

Some learners indicated that they had evaluated their programs to determine if the expected outcomes had been attained and their aims had been reached by using the best strategy.

> The program worked in the end and the expected outcomes were attained and we evaluated the way we solved the problem and tried to find a better, simpler way.

- *Learning that took place*

Reflection on what was learned by doing the task and the future use of the strategy was touched on:

> We reflected back and discussed what other problems can be solved using the statements and other problem-solving skills we attained when solving the problem.

- *Evaluation of process*

Some entries related specifically to the learners' experiences of the use of the SDM questions.

> This phase makes me check if everything is working correctly and if I am able to use the same method for other problems.
>
> With the help of this journal I reflected on some things that had been done in this activity.

Most learners still viewed evaluation in the post-coding phase to be the mere testing of the program and correction of errors.

> The program should be checked and tested if it meets the requirements and runs properly.

Discussion and Recommendations

The results underline Joseph's (2009) concern that teachers commonly do not implement practices to develop metacognitive awareness. The teachers involved in this study were either ignorant about metacognition or fitted in with Joseph's (2009) view that the teaching of metacognitive skills does not come naturally and teachers therefore find it difficult, and that teachers neglect it mostly due to limited instructional time. These findings uphold the researchers' view that training in metacognitive skills could contribute to empowering teachers in terms of their own metacognitive awareness, as well as their ability and commitment to create opportunities to develop learners' metacognitive skills. Considering the time of training the teachers with regard to the development of metacognitive skills as a limitation in the current research, more attention should be given in future research to the training of the teachers. They themselves need some time to assimilate the process before they can be committed to develop their learners' metacognitive skills. A positive aspect of the findings was the teachers' realisation of the importance of metacognitive awareness and how learners gained from it. The teachers also realised that practice of metacognitive strategies was necessary to make it an integral part of the learners' learning activities.

Another important issue emanating from the results was the teachers' experience of their learners' lack of metacognitive awareness and thinking skills, and the learners' unwillingness to implement the SDM questions. It could be that the learners' attitude was a result of their teachers' instructional methods and needed to be addressed. The teachers' observation that the learners just wanted to get on with the coding of a computer program and did not want to '*waste time*' implementing the relevant metacognitive activities in the different phases of solving a programming problem should encourage teachers to consciously nurture the metacognitive awareness and the development of the metacognitive skills of their learners. These objectives can be reached, as was proved by previous research and implemented in this research, through teacher modelling, discussions about metacognition, and active implementation of metacognitive and reflective activities (Joseph, 2009; Kramarski & Mevarech, 2003). Reflective activities, such as the learners' journals, afford learners the opportunity to explicitly consider their thinking and learning processes. In the case of this research, the activity took a lot of time because of the learners' difficulty to describe their thinking processes in a language that was not their home language.

Results further suggest that learners involved in this study need a longer time span, more encouragement and focused support when implementing the SDM questions. Teachers should guard against a situation where learners perceive the use of metacognitive activities just as superfluous. The SDM questions will be more effective when used over a longer period and with more experience. This could be supported by introducing the method gradually and letting learners experiment with it. If they also get dedicated time for this while not taking time away from their usual problem-solving time, they would probably see it as quite useful after six months to a year.

What became clear from the results of the learners' journals is that they had found the SDM questions useful in the pre-coding phase of the programming task. Their attention was focused on making sure that they understood what the task required, they could identify relevant information, and they had to think about a solution before 'jumping to the keyboard'. Consultation of resources, consideration of alternative strategies and time management received less attention. This tendency indicates the types of activities teachers specifically will have to attend to in future efforts to develop metacognitive skilfulness. It was also found in the intra-coding and post-coding phases that some metacognitive activities received attention while others were neglected. Activities that received most attention in the intra-coding phase included checking whether they were still on track to get to the goals they had set for themselves, monitoring their progress, correcting errors as they went along, and changing their planning or procedures when necessary. It would be necessary to ensure that learners do these activities not just for the sake of getting the program to run, but because they understand how each step fits in with what has already been done and what has to be done next in their endeavour to solve the problem. The use of metacognitive activities in the post-coding phase would also need a lot of attention since most learners see this phase merely as a time to correct errors and to get the program to run, without considering what could have been done better or what was learned from it.

Development of learners' metacognitive regulation requires purposeful development of the metacognitive elements of planning, monitoring and evaluation, encompassing the metacognitive aspects relevant in each element. The schematic representation given in figure 1 can be used to direct purposeful development of the elements of metacognitive regulation by attending to each of the relevant aspects. Aspects that do not feature in learners' metacognitive thinking can then be specifically addressed. In this investigation, for example, the three aspects of planning that received little attention were consideration of alternative strategies (during the design of the algorithm), use of extra resources, and time scheduling. The authors speculate that this might be due to the time constraint normally experienced in a classroom situation. Aspects of monitoring that seemingly were neglected were making sure that you really understand what you are doing and, again, time scheduling. For the learners in this investigation the aspects of the evaluation element would all require special attention. Though most of the aspects were touched on, it was mentioned by only a small number of learners. It seemed that most learners were unconcerned about evaluation of their tasks. As soon as they had some kind of product they did not pay further attention to the quality, the process, what they had learned from it, or how they would use it in future applications.

To address the language problem, as experienced in this investigation, alternative data collection methods may have to be considered. Future research based on personal interviews with learners and real-time observation of their activities during pair programming might shed more light on the development of learners' metacognitive regulation, and bring about more insight regarding the influence of the SDM questions during pair programming on all aspects.

Conclusion

In this article, we have reported on the use of SDM questions during pair programming in economically deprived rural schools as part of an internationally funded project about the empowerment of IT in North-West and KwaZulu-Natal Provinces. Results revealed that the teachers were quite uninformed regarding the notion of metacognitive regulation, that they found the development of these skills difficult and time-consuming, and that they experienced learners to be reluctant and unwilling to use the SDM questions. Results emanating from the learners' journals suggested that the learners had found the SDM questions useful in the pre-coding phase of their programming tasks, but that they tended to neglect the metacognitive activities during the other phases. A positive trend was that both teachers and learners realised the advantages of the metacognitive approach to problem solving during pair programming. Yet, a different procedure of implementation is required making the metacognitive approach an integral part of the teaching and learning activities, and thus less time-consuming.

The research indicated that learners in economically deprived rural schools could benefit from purposeful and timely development of metacognitive skills during pair programming. Implementing a metacognitive approach to pair programming does not require many resources, but rather a disposition of commitment and willingness. Both teachers and learners involved in this research expressed their realisation of how learners could gain from a metacognitive approach to problem solving and programming. The teachers also acknowledged that metacognitive strategies can become an integral part of learners' learning activities, if explicitly studied and practiced.

Acknowledgement

This research is based upon work financially supported by South Africa – Netherlands Research Program on Alternatives in Development (SANPAD). Any opinion, findings, conclusions or recommendations expressed in this article are those of the authors, and therefore SANPAD does not accept any liability in regard thereto.

References

Anderson, D., & Nashon, S. (2007). Predators of knowledge construction: Interpreting students' metacognition in an amusement park physics program. *Science Education*, *91*(2), 298–320.

Berenson, S., Slaten, K., Williams, L., & Ho, C. (2004). Voices of women in a software engineering course: Reflections on collaboration. *ACM Journal of Educational Resources in Computing*, *4*(1), 1–18.

Bipp, T., Lepper, A., & Schmedding, D. (2008). Pair programming in software development teams – an empirical study of its benefits. *Information and Software Technology*, *50*(3), 231–240.

Breed, E. A. (2010). *'n Metakognitiewe onderrigleerstrategie vir paarprogrammeerders ter verbetering van kennisproduktiwiteit*. Doctoral thesis. Potchefstroom, RSA: North-West University.

Chalmers, C., & Nason, R. A. (2003). *Developing primary students' group metacognitive processes in a computer supported collaborative learning environment*. Paper presented at the AARE – NZARE Joint Conference, November 29–December 3, Auckland, New Zealand. Retrieved October 20, 2010, from http://www.aare.edu.au/03pap/nas03501.pdf

Joseph, N. (2009). Metacognition needed: Teaching middle and high school students to develop strategic thinking skills. *Preventing School Failure*, *54*(2), 99–103.

Katira, N., Williams, L., Wiebe, E., Miller, C., Balik, S., & Gehringer, E. (2004). On understanding compatibility of student pair programmers. In *Proceedings of the 35th SIGCSE Symposium on Computer Science Education*, March 3–7 (pp. 7–11). New York, NY: ACM.

Kramarski, B. (2004). Making sense of graphs: Does metacognitive instruction make a difference on students' mathematical conceptions and alternative conceptions? *Learning and Instruction*, *14*(6), 593–619.

Kramarski, B., & Mevarech, Z. R. (2003). Enhancing mathematical reasoning in the classroom: The effects of cooperative learning and metacognitive training. *American Educational Research Journal*, *40*(1), 281–310.

Kramarski, B., Mevarech, Z. R., & Arami, M. (2002). The effects of metacognitive instruction on solving mathematical authentic tasks. *Educational Studies in Mathematics*, *49*, 225–250.

Kramarski, B., Mevarech, Z. R., & Lieberman, A. (2001). Effects of multilevel versus unilevel metacognitive training on mathematical reasoning. *The Journal of Educational Research*, *94*(5), 292–300.

Mentz, E., Bailey, R., Havenga, M., Breed, B., Govender, D., Govender, I., … Dignum, V. (2012). The diverse educational needs and challenges of Information Technology teachers in two black rural school. *Perspectives in Education*, *30*(1), 70–78.

Mevarech, Z. R. (1999). Effects of metacognitive training embedded in cooperative settings on mathematical problem solving. *The Journal of Educational Research*, *92*(4), 195–205.

Mevarech, Z., & Fridkin, S. (2006). The effects of IMPROVE on mathematical knowledge, mathematical reasoning and meta-cognition. *Metacognition and Learning*, *1*(1), 85–97.

Nicolescu, R., & Plummer, R. (2003). A pair programming experiment in a large computing course. *Romanian Journal of Information Science and Technology*, *6*(1–2), 199–216.

Nieuwenhuis, J. (2007). Qualitative research designs and data gathering techniques. In K. Maree (Ed.), *First steps in research* (pp. 69–97). Pretoria, RSA: Van Schaik.

Nonyane, J., & Mlitwa, N. (2008). ICT access and use in rural schools in South Africa: A case study in Mpumalanga Province. *Section 1: IS, ICT & E-commerce, E-health, E-learning*, *1*, 94–104.

Paris, S. G., & Paris, A. H. (2001). Classroom applications of research on self-regulated learning. *Educational Psychologist*, *36*(2), 89–101.

Sternberg, R. J. (2001). Metacognition, abilities, and developing expertise: What makes an expert student? In H. J. Hartman (Ed.), *Metacognition in learning and instruction: Theory, research and practice* (pp. 247–260). Dordrecht: Kluwer.

Veenman, M. V. J., & Spaans, M. A. (2005). Relation between intellectual and metacognitive skills: Age and task differences. *Learning and individual differences*, *15*(2), 159–176.

Williams, L., & Upchurch, R. (2001a). Extreme programming for software engineering education? Paper presented at the 31st ASEE/IEEE Frontiers in Education Conference, October 10–13, Reno, NV. doi: 10/1109/FIE.2001.963882

Williams, L., & Upchurch, R. (2001b). In support of student pair-programming. *ACM SIGCSE Bulletin*, *33*(1), 327–331.

APPENDIX

SELF-DIRECTED METACOGNITIVE QUESTIONS

Date: ______________________________

Use the following check-list with self-directed questions to direct your activities and thoughts during the execution of the pair programming task. Use a tick (√) in the applicable block to indicate if attention was given to the specific aspect.

Pre-coding phase

• Have we identified the most important information that was given?	
• Do we understand exactly what outcomes we have to reach with the solution of this problem?	
• Have we thought about aspects of the problem that are already known to us?	
• Have we thought about resources that can help us with the solution of the problem?	
• Do we know strategies/procedures that are appropriate to solve the problem?	
• Have we decided on the sequence in which the strategies/procedures must be applied?	

Intra-coding phase

• Do we continuously ensure that we understand what we are doing?	
• Do we continuously ensure that we are still on the right track?	
• Do we think about how every step fits into what has already been done and what needs to be done next?	
• Do we identify and rectify errors as the coding progresses?	
• Do we change the planning if necessary?	
• Do we monitor the progress made with the coding?	
• Do we consult the available resources if we need help?	

Post-coding phase

• Did we check if the expected outcomes had been reached?	
• Have we thought about whether the best possible strategies/procedures were used to solve the problem?	
• Have we reflected on what could possibly have been done in a different way?	
• Will we be able to apply the same thought processes in similar problems in future?	
• Have we reflected on what we have learned from solving this specific problem?	