



Un Enfoque Metacognitivo a la Programación de Pares: su Influencia sobre Conciencia Metacognitiva

Betty Breed¹, Elsa Mentz¹, y Gert van der Westhuizen³

¹ Faculty of Education Sciences, North-West University, Potchefstroom

² Department of Educational Psychology, University of Johannesburg,

Sudáfrica

Correspondencia: Betty Breed. Faculty of Education Sciences, North-West University, Private Bag X6001, Potchefstroom, 2520. South Africa. E-mail: betty.breed@nwu.ac.za

© Education & Psychology I+D+i and Ilustre Colegio Oficial de Psicólogos de Andalucía Oriental (Spain)

Resumen

Introducción. La investigación se centró en la metacognición en un entorno de aprendizaje colaborativo. Basado en un estudio exhaustivo de la bibliografía, los investigadores diseñaron una estrategia de enseñanza- aprendizaje metacognitivo para parejas de programadores. Nuestro objetivo fue investigar la influencia de esta estrategia de enseñanza - aprendizaje metacognitivo durante la programación en parejas en un contexto educativo, en la conciencia metacognitiva.

Método. Se utilizó un diseño de investigación mixto, incluyendo dos cuestionarios y entrevistas individuales. Los participantes fueron 67 estudiantes de onceavo grado que habían estado aplicando la programación en parejas en el aprendizaje de las Tecnologías de la Información en la enseñanza secundaria. La estrategia de enseñanza- aprendizaje metacognitivo fue implementado por el grupo experimental.

Resultados. Los resultados derivados del estudio empírico muestran que los alumnos del grupo experimental , que se había implementado la estrategia de enseñanza- aprendizaje metacognitivo mientras programaban en parejas, reportaron una mejora en las áreas de planificación, información de gestión, seguimiento y evaluación, con respecto a la conciencia metacognitiva .

Discusión y conclusiones. Esta investigación pone de relieve la importancia de una estrategia de enseñanza-aprendizaje metacognitivo cuando los alumnos están programando en parejas. Los hallazgos podrían hacer mucho para promover la enseñanza y el aprendizaje en la programación de computadoras, y para informar a la investigación futura y la práctica. La estrategia de enseñanza-aprendizaje metacognitivo aplicado durante la investigación se ha adaptado para que sea adecuado para el uso de profesores y alumnos durante la programación en parejas.

Palabras clave: metacognición, aprendizaje colaborativo, programación en parejas , estrategia enseñanza - aprendizaje metacognitivo .

Recibido: 24/09/13

Aceptación inicial: 23/10/13

Aceptación final: 20/03/14

A Metacognitive Approach to Pair Programming: Influence on Metacognitive Awareness

Abstract

Introduction. The research focused on metacognition in a collaborative learning setting. Based on a comprehensive literature study the researchers designed a metacognitive teaching-learning strategy for pair programmers. Our purpose was to investigate the influence of this metacognitive teaching-learning strategy during pair programming in an educational context on metacognitive awareness.

Method. A mixed method research design was used, including two questionnaires, individual interviews, and individual journals. The participants were 67 grade 11 learners who had been implementing pair programming in the learning of Information Technology at secondary school level. The metacognitive teaching-learning strategy was implemented by the experimental group.

Results. The results derived from the empirical study showed that the learners in the experimental group, who had implemented the metacognitive teaching-learning strategy while programming in pairs, reported an enhancement in the areas of planning, information management, monitoring and evaluation, with regard to metacognitive awareness.

Discussion and conclusion. This research highlighted the importance of a metacognitive teaching-learning strategy when learners are programming in pairs. The findings could do much to advance teaching and learning in computer programming, and to inform future research and practice. The metacognitive teaching-learning strategy implemented during the research was adapted to make it suitable for use by teachers and learners during pair programming.

Keywords: metacognition, collaborative learning, pair programming, metacognitive teaching-learning strategy.

Received: 09/24/13

Initial acceptance: 10/23/13

Final acceptance: 03/20/14

Introduction

In industry the emphasis has moved from the work and the abilities of individuals to the contribution that could be made in an organisation if individuals collaborated to produce and apply new knowledge (Vaicaityte, 2006). Unsurprisingly, this shift in focus from individual work to collaboration in groups also brought about changes in the field of industrial software development. The traditional approach in terms of which a programmer worked alone has changed to one where professional programmers now work in pairs at the same computer, on the same design, algorithm, code and test (Williams, Kessler, Cunningham, & Jeffries, 2000; Williams & Upchurch, 2001a). Industry's need for programmers who are able to work in pairs, as well as the success that has been achieved with this approach, has led to the implementation of pair programming in the teaching and learning of programming skills. Pair programming is resultantly currently widely used in educational contexts for learning computer programming (Preston, 2006) and at the same time it prepares learners to work in teams with other individuals (Cliburn, 2003).

The importance in education of implementing metacognitive skills to render learning more meaningful and effective has been researched for quite some time now. Little research has been done however, regarding the role of metacognition in the teaching and learning of computer programming skills, whether individually or collaboratively. According to Williams and Upchurch (2001b), metacognition, as is the case with other forms of knowledge and skills, should explicitly be attended to in teaching and learning. Based on the important role of metacognitive skills in learning, the research reported in this article aimed at determining whether explicit development of metacognitive skills during pair programming could enhance the metacognitive awareness of the members of the programming pair.

Pair programming

Pair programming is a collaborative programming technique where all the programming of a programming task is done by two programmers working at the same computer (Berenson, Slaten, Williams, & Ho, 2004). In the educational context, when two learners work as a pair on a programming task, each fulfils a specific role. The *driver* is actively busy typing on the computer or coding a design (Nicolescu & Plummer, 2003), whereas the *navigator* observes the driver's work, identifies problems, asks questions and makes proposals to improve the program design or code (Berenson et al., 2004). The navigator is responsible for

guiding the problem-solving process by suggesting alternative strategies and using applicable resources (Bipp, Lepper, & Schmedding, 2008). All of these tasks of the navigator require metacognitive thinking. The pair switches roles on a regular basis to ensure that both learners get the opportunity to develop the different skills. In educational context, the emphasis in pair programming is on interaction between the two members of the pair, with the outcome of improved learning (McDowell, Hanks, & Werner, 2003). The extent of the enhancement of learning brought about by the implementation of pair programming in the teaching and learning of programming skills, is reflected in aspects such as better quality of independent tasks (McDowell, Werner, Bullock, & Fernald, 2006; Mendes, Al-Fakhri & Luxton-Reilly, 2005; Williams, Wiebe, Yang, Ferzli, & Miller, 2002), higher examination marks and pass percentages (Mendes et al., 2005; Williams et al., 2002), higher course pass rates (McDowell et al., 2006; Williams et al., 2002) and an increase in learners' confidence in their work (McDowell et al., 2006; Mendes et al., 2005).

As a teaching-learning strategy, pair programming is grounded in the principles of collaborative learning methods, aimed at interaction between learners in the discussion and solving of problems (Ferzli, Wiebe, & Williams, 2002). A social-constructivist context is created (Berenson et al., 2004) in which learning is seen as a process sparked off by the social interaction between the two learners (Chaparro, Yuksel, Romero, & Bryant, 2005). Learners in collaborative learning environments have to provide explanations to each other, which is one of the best ways to extend one's knowledge and make new knowledge connections (Kramarski, 2004). 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.

Metacognitive awareness and the explicit development of metacognitive skills

Metacognitive awareness involves the individual's awareness of the way in which he/she learns, constructs knowledge or develops understanding, as well as his/her knowledge of learning strategies, the demands of the learning task and his/her progress (Anderson & Nashon, 2007; Wilson & Johnson, 2000). Metacognitive awareness enables the learner to learn how, when and where to use cognitive strategies. Learners who are metacognitively aware, are able to apply their metacognitive knowledge and to monitor and manage their own learning, including active reflection on their own thoughts and making decisions about what to do and how to act (Wilson & Johnson, 2000).

The importance of learners explicitly learning metacognitive knowledge and skills is emphasised by Williams and Upchurch (2001a). According to Williams and Upchurch (2001b), learners should, in addition to declarative knowledge (knowledge of “what”) and procedural knowledge (knowledge of “how”), also be able to implement metacognitive skills such as planning of a strategy, monitoring a process and its progress, making adjustments when necessary, and reflecting on the whole process to be able to improve. Most learners need to be taught and encouraged to develop these skills (Joseph, 2009).

The effect of purposeful teaching of metacognitive skills, especially in problem-solving contexts, is confirmed by the research of Kramarski (2004), Kramarski, Mevarech and Arami (2002), Kramarski, Mevarech and Lieberman (2001), Kramarski and Mizrachi (2004), Mevarech (1999), and Mevarech and Fridkin (2006), regarding the effect of metacognitive instruction on different aspects of mathematics in different learning contexts. The positive outcomes reached with metacognitive instruction in mathematics while learners are working in groups (Kramarski, 2004) emphasise the importance of this research on the effect of purposeful metacognitive instruction on the metacognitive awareness of the two members of a pair during pair programming.

According to Chalmers and Nason (2003), the development of the metacognitive skills of learners working in groups not only influences their knowledge, but also the contribution that each member of the group can make towards reaching the learning outcomes. Group members who have knowledge of metacognitive strategies such as planning, monitoring and evaluating, are furthermore able to make better contributions to complete the group task (Chalmers & Nason, 2003). Explicit instruction of metacognitive skills enables learners in a group to consider different applicable strategies to solve a problem, suggest and compare different strategies, and analyse the advantages and disadvantages of each strategy (Kramarski et al., 2002), which are the activities expected from the navigator in a programming pair. The role of metacognition in meaningful learning while working in groups necessitate the development of the metacognitive skills of learners programming in pairs, albeit that the group only has two members in the case of pair programming.

In this project, the metacognitive activities embedded in the problem-solving activities during the execution of a pair programming task were specifically addressed. Programming

tasks are generally complicated to solve, with probably more than one way of solving it, thus affording opportunity for interaction and metacognitive activities. Pair programming served as the teaching-learning context in which the learners collaborated to solve a problem through their analysis of it, the generation of a possible solution, and the application and evaluation of the solution.

Based on an intensive literature study regarding the development and improvement of metacognition in the classroom (e.g., Cardelle-Elawar, 1995; Gama, 2004; Mevarech & Kramarski, 1997; Palincsar & Brown, 1984; Scardamalia, Bereiter & Steinbach, 1984; Schoenfeld, 1991; Schraw, 2001), and keeping the unique nature of pair programming in mind, a metacognitive teaching-learning strategy (MTLS) for pair programmers was developed for purposes of this project. The MTLS for pair programmers consisted of four components.

The first component of the MTLS (component 1) was a diagram representing the elements of metacognition as viewed in this research. In general, researchers agree that metacognitive knowledge and metacognitive control are the two fundamental components of metacognition (e.g. Hofer, Yu, & Pintrich, 1998; Schraw, 2001; Zeidner, Boekaerts, & Pintrich, 2005). The close relationship and interaction between metacognitive knowledge and metacognitive control is brought about by the individual's ability to engage in reflection. Figure 1 represents the elements of metacognition diagrammatically.

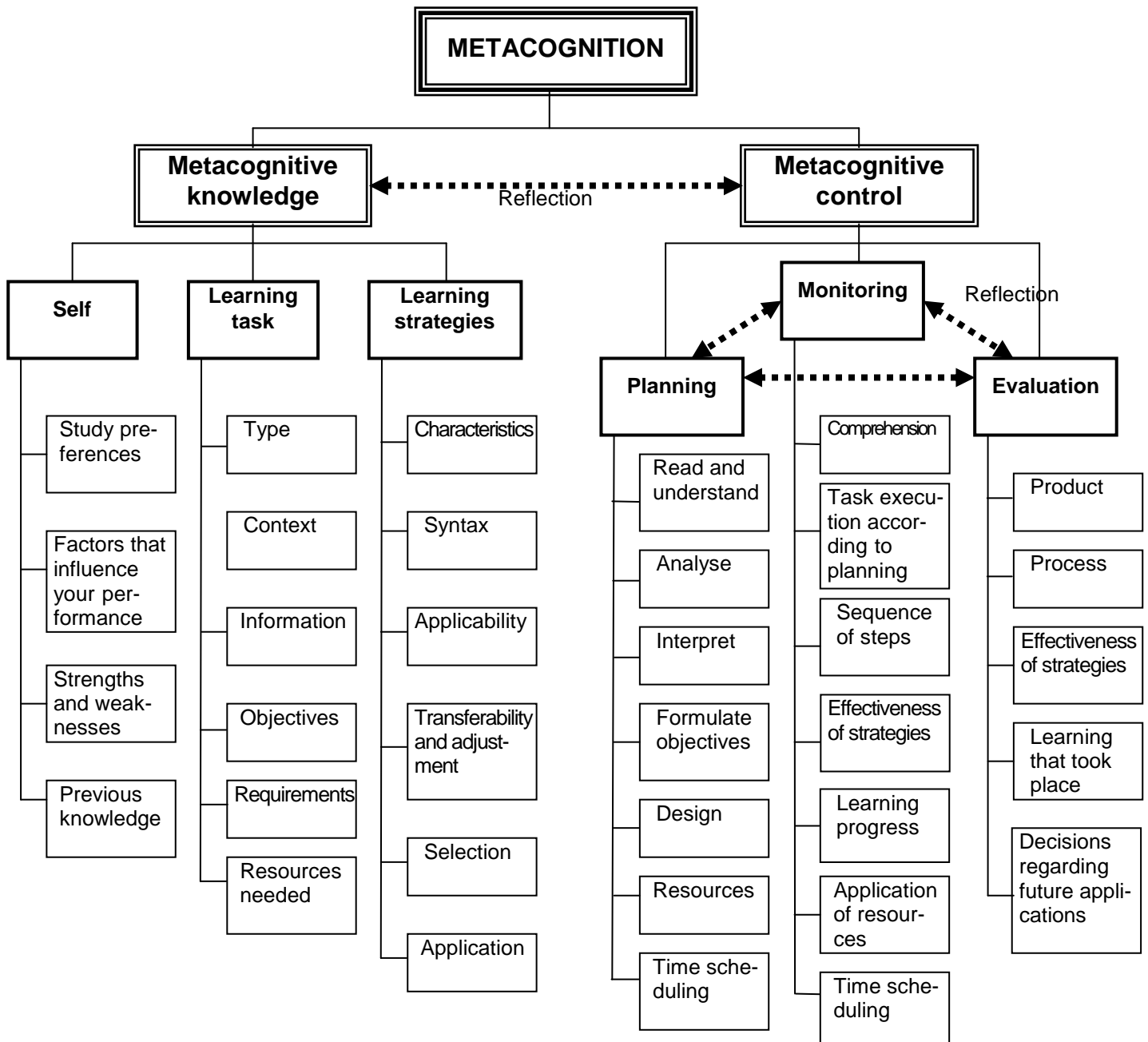


Figure 1. Diagrammatical representation of the elements of metacognition

A pair programming task falls into three phases, namely the pre-coding, intra-coding and post-coding phases. The second component of the MTL (component 2), a list of the metacognitive activities that should be attended to *while collaborating* in each of the phases of the programming task, was compiled, based on the literature study concerning metacognition during problem solving and existing examples of metacognitive teaching-learning strategies. Metacognitive activities that had to be attended to in the pre-coding phase regarding analysis of the problem and design of a solution included, for example, “Reflection on previ-

ous knowledge - determine relation to own experiences; make connections with aspects of the problem that are already known” and “Formulating the goals to be reached with the solution of the problem”. In the intra-coding phase the learners had to attend to metacognitive activities with regard to monitoring their own progress, for example “Regular re-consideration of whether what is being done is still understood” and “Anticipation of what has to be done next”. Finally, in the post-coding phase, the learners had to attend to metacognitive activities relating to the set goals, the strategies used and future applications. These activities included aspects such as, for example, “Evaluation of the extent to which the set aims were reached” and “Reflecting on what could have been done differently”.

The third component of the MTLs (component 3) was a set of self-directed metacognitive questions (henceforth referred to as SDM questions) that the programming pair had to answer collectively during the execution of a programming task. This set of questions was compiled based on the list of the metacognitive activities (see previous paragraph) that should be attended to *while collaborating* in each of the three phases of the programming task. While analysing the problem and designing a solution to it in the pre-coding phase, the pair had to answer questions that were intended to direct their metacognitive activities in this phase, for example, “Have we thought about aspects of the problem that are already known to us?” and “Do we understand exactly what outcomes we have to reach with the solution of this problem?” Self-directed metacognitive questions that the pair had to answer during the intra-coding phase included questions like “Do we continuously ensure that we understand what we are doing?”, “Do we think about how every step fits into what has already been done and what needs to be done next?” and “Do we monitor the progress made with the coding?” Examples of the self-directed metacognitive questions that the pair had to answer in the post-coding phase are “Did we check if the expected outcomes had been reached?” and “Have we thought about whether the best possible strategies/procedures were used to solve the problem?”

The fourth component of the MTLs (component 4) was the journals that the learners had to complete individually after completing a pair programming task. The purpose of the journals was to provide them with the opportunity to individually reflect on their own thinking processes and activities during the execution of the programming task.

Objective

The objective of this research was to investigate the influence of the implementation of the MTLs during pair programming in an educational context on the metacognitive awareness of the learners. We specifically wished to determine if implementation of the MTLs developed the learners' metacognitive awareness with regard to metacognitive control when solving problems during pair programming.

Method

Participants

The teachers and learners who were involved were required to be already skilful in the use of pair programming as teaching-learning strategy. Since there were only six schools in this particular South African province that had already implemented pair programming as a teaching-learning strategy, it was not possible to use a random sample, and thus the research was conducted in these schools. Five of the six schools were willing to participate in the research. Because of the low numbers of learners taking Information Technology as a school subject in South African secondary schools, all the grade 11 learners in Information Technology at these schools were involved. The decision to involve grade 11 learners was based on the fact that they had already used pair programming in grade 10 and that they were familiar with pair programming as a teaching-learning strategy. In order to have approximately the same number of learners in the experimental and control groups, the five schools were divided in a group of two (n=36) and a group of three schools (n=35). We randomly assigned the status of experimental group and of control group to the groups. The two groups were comparable in terms of prior programming skills and knowledge, socio-economic status, gender and average age.

All of the participants in both groups were involved in the quantitative part of the research. During the research, all the participants worked in pairs while programming. Pairs were randomly assigned by the teacher on a weekly basis. Weekly reconstruction of pairs ensured that learners worked with partners of different proficiency levels and personality types. In this way the issue of dyad effects was minimised.

For the qualitative part of the project, only the participants of the experimental group were expected to complete the SDM questions and to keep the personal journals. For the individual interviews, 20% of the participants at each school were identified, in consultation with the teacher, in such a way that the group included high, medium and low performers, based on their grade 10 marks for programming. The same group of learners at each school was interviewed at the beginning of the research, before the intervention and again after the intervention. Any participants who were absent during the second or third round of interviews were removed from the group of interviewees. A total number of 14 learners were interviewed on each of the three occasions.

Instruments

Questionnaire regarding metacognitive awareness: The purpose of this questionnaire was to determine the participants' self-reported metacognitive awareness and was based on the Metacognitive Awareness Inventory of Schraw and Dennison (1994). With the consent of the authors, the questionnaire was adapted to make it relevant for learners programming in pairs in the Information Technology classes in South African secondary schools. In accordance with the Metacognitive Awareness Inventory, the questionnaire used in this research also consisted of 52 items which were divided into eight constructs (categories). The constructs included declarative knowledge, procedural knowledge, conditional knowledge, planning, information management strategies, monitoring, debugging strategies and evaluation. The same set of questions was used on the three occasions, but with each test the questions were set in a different order to minimise the contamination effect. The Cronbach alpha values for these constructs were all higher than 0.7 which proves the reliability of the constructs for the population in which the questionnaire was used.

Individual interviews: The qualitative, semi-structured interviews had an informal conversational character. The participants were expected to answer a set of predetermined questions regarding their metacognitive awareness during the pre-coding, intra-coding and post-coding phases of a pair programming task, especially with regard to metacognitive control. When necessary, follow-up questions were used to clarify the participants' answers (Nieuwenhuis, 2007), without revealing the researcher's own perspectives. The set of questions used on the three occasions of interviewing were differently formulated, but required the same type of information from the learners. This was done so that they had to think about

their answers again and not simply respond, based on what they remembered from the previous interview.

The set of questions used in the third interview included an additional question aimed at getting information about the participants' experiences of using the SDM questions during pair programming.

Individual journals: At the beginning of the intervention period, the participants in the experimental group each received a file that served as their individual journal. The files contained a number of documents and forms that they had to use and/or complete during the execution of pair programming tasks that were done during the intervention period. The first document in each file was a list of the metacognitive activities that had to be attended to in the pre-coding, intra-coding and post-coding phases of a pair programming task (component 2 of the MTLs). This list served as a reference document for the participants to review each time before they started on a new pair programming task. The files also contained a number of lists with the SDM questions (component 3 of the MTLs) and the same number of forms that the learners had to complete to describe their thought processes during the three phases of a pair programming task (component 4 of the MTLs). During the execution of each pair programming task the pair had to collaboratively read and tick off the SDM questions to guide their activities and thought processes. After completion of the task each participant had to complete the form (component 4) individually, in which they had to describe their thought processes during the three phases of a pair programming task.

Procedure

Data was collected by using questionnaires, individual interviews and individual journals. The questionnaires were completed and the individual interviews conducted right at the beginning of the research to make an initial evaluation (test 1) of the difference between the experimental group and the control group regarding their self-reported metacognitive awareness. These evaluations were repeated three months later (test 2), without any intervention, as a control measure to verify that pair programming *per se* had not influenced the participants' self-reported metacognitive awareness.

At the beginning of the intervention period, the concept of metacognition was discussed with the teachers and learners who were involved in the experimental part of the re-

search, by using component 1 of the MTLs. The elements of metacognitive control (planning, monitoring and evaluation) were emphasised to show the relationship with the phases of a pair programming task (pre-coding, intra-coding and post-coding). Using components 2 and 3 of the MTLs the use of metacognitive activities and the completion of the SDM questions were then modelled, giving the participants the opportunity to clarify any uncertainties. At the end of the experimental phase (after three months), during which the participants in the experimental group used the SDM questions while doing pair programming two to three times a week, solving problems that required interaction and the application of metacognitive skills, all participants again completed the questionnaires and the selected participants were again interviewed (test 3). In the period between the test 2 and test 3 the experimental group also kept their individual journals (component 4 of the MTLs) during pair programming.

Design and data analysis

A mixed method research design was used that included a repeated-measures control group experiment for the quantitative part of the project. Although a repeated-measures experiment usually does not include a control group, it was decided to include a control group so that comparison with the experimental group could show whether the self-reported metacognitive awareness of participants who were not part of the intervention, stayed unchanged. After calculating the descriptive statistics, a confirmatory factor analysis was done with the objective, according to Nicol and Pexman (1999), to determine whether the items of the questionnaires were correctly grouped in constructs. The principal component factor analysis was done using the Oblimin rotation method and Kaiser normalisation (Field, 2005). The factor analysis confirmed the validity of the constructs for the particular population. Because of the small number of participants ($n=71$) an exploratory factor analysis was then done only within each of the constructs to see if one or more factors existed in these constructs. If so, it was ascertained that these factors made sense theoretically and could be explained in terms of the literature study. At all stages of analyses the quantitative data was analysed by doing independent t-tests to test for differences within groups regarding their levels of metacognitive awareness. Because of the anonymity of respondents the pre- and post-test data could not be paired to do a dependent t-test. The p-values were calculated but not interpreted, because the sampling had not been done randomly. Only effect sizes were used to determine the practical significance of the differences (Cohen, 1988). Multivariate statistics (Wilk's lambda) were also used to determine the difference between the experimental and control groups at the different stages.

The qualitative part of the project entailed the personal interviews with participants in both of the control and experimental groups, as well as the individual journals that were kept by the participants in the experimental group. Since it was the aim of the qualitative part of this research to supplement the quantitative results and contribute to understanding and explaining the results, a basic qualitative analysis approach was followed. Transcription of the interviews and typing of the entries of the journals were followed by utilising the computer software Atlas.ti to organise the relevant information into segments of text and link these segments to codes. A combination of *a priori* and emerging codes was used. The *a priori* codes used were in accordance with the constructs of the questionnaires to ease the comparison of the qualitative and quantitative data. Other emerging codes allowed for further understanding and explanation of the results. Reliability of the processes was ascertained by having transcripts checked for correctness and codes cross-checked by another expert in the field of study.

Results

Construct scores

Table 1 shows the construct scores for the participant group in total (N=71) at the beginning of the research. These construct scores were calculated to get an idea of the initial metacognitive awareness of the group as a whole, keeping in mind that each participant's response on an item was their own opinion of their level of metacognitive. The abbreviations M1 to M8 are used for the eight constructs regarding metacognitive awareness (see table 1). Henceforth M1 to M8 will be used when referring to these constructs.

Table 1. Construct scores for total group at beginning of research

Construct	Description	N	Average	Standard deviation
M1	Declarative knowledge	71	4.66	(0.89)
M2	Procedural knowledge	71	4.71	(0.87)
M3	Conditional knowledge	71	4.76	(0.88)
M4	Planning	71	4.23	(1.03)
M5	Information management strategies	71	4.21	(0.85)
M6	Monitoring	71	4.15	(0.98)
M7	Debugging strategies	71	4.73	(0.87)
M8	Evaluation	71	4.19	(1.00)

The questionnaire on metacognitive awareness was answered on a 6-point Likert-type scale which means the arbitrary scale midpoint is 3.5. Comparing the averages of M1 to M8 in table 1 with the arbitrary scale midpoint, it seems that the entire group of participants already possessed some degree of metacognitive awareness, since the averages of all the constructs were higher than 3.5.

Metacognitive awareness

The effect sizes that were calculated with the independent t-test in test 1 (initial evaluation) did not yield any significant difference between the experimental and control groups regarding the constructs of metacognitive awareness (see table 2).

Table 2. Effect sizes for constructs of metacognitive awareness in tests 1, 2 and 3

Test	Construct	Experimental group		Control group		t-value	df	p	Effect size
		Average	Standard deviation	Average	Standard deviation				
Test 1	M1	4.61	0.91	4.71	0.87	-0.51	69	0.61	0.12
	M2	4.63	0.84	4.79	0.91	-0.74	69	0.46	0.17
	M3	4.64	0.89	4.87	0.87	-1.12	69	0.27	0.26
	M4	4.34	0.91	4.12	1.15	0.89	69	0.38	0.19
	M5	4.30	0.87	4.11	0.84	0.93	69	0.35	0.22
	M6	4.13	0.90	4.18	1.06	-0.21	69	0.84	0.05
	M7	4.79	0.85	4.67	0.90	0.58	69	0.56	0.13
	M8	4.31	0.99	4.06	1.01	1.03	69	0.31	0.24
Test 2	M1	4.47	0.92	4.76	0.77	-1.41	68	0.16	0.31
	M2	4.26	1.01	4.80	0.86	-2.43	68	0.02	**0.54
	M3	4.38	0.93	4.83	0.73	-2.26	68	0.03	**0.48
	M4	4.17	1.07	4.27	1.05	-0.39	68	0.70	0.09
	M5	4.21	1.01	4.37	0.83	-0.71	68	0.48	0.15
	M6	4.04	1.01	4.38	0.80	-1.55	68	0.13	0.33
	M7	4.45	0.84	4.67	0.86	-1.07	68	0.29	0.25
	M8	3.85	1.34	4.06	0.97	-0.74	68	0.46	0.15
Test 3	M1	4.48	0.86	4.60	0.88	0.61	67	0.55	*0.14
	M2	4.51	0.77	4.67	0.80	-0.82	67	0.41	*0.19
	M3	4.51	0.72	4.54	0.80	-0.16	67	0.88	0.04
	M4	4.42	0.91	4.08	0.96	1.52	67	0.13	**0.36
	M5	4.44	0.72	4.25	0.81	1.00	67	0.32	*0.23
	M6	4.34	0.88	4.13	0.87	1.01	67	0.32	*0.24

M7	4.55	0.69	4.76	0.63	-1.31	67	0.20	*0.30
M8	4.34	0.91	3.83	1.07	2.14	67	0.04	**0.48

* Small effect size

** Medium effect size

*** Large effect size - practical significant difference

The averages of the constructs in test 1, as in figure 2, show that for some of the constructs the averages for the experimental group were higher than those of the control group (M4, M5, M7 and M8). This tendency was confirmed by the results of the individual interviews with the participants in the experimental and control groups at the same stage. For the other constructs (M1, M2, M3 and M6) the control group had higher averages. It is noticeable that for both groups the averages of the constructs regarding planning (M4), information management strategies (M5), monitoring (M6) and evaluation (M8) were lower than the rest of the averages.

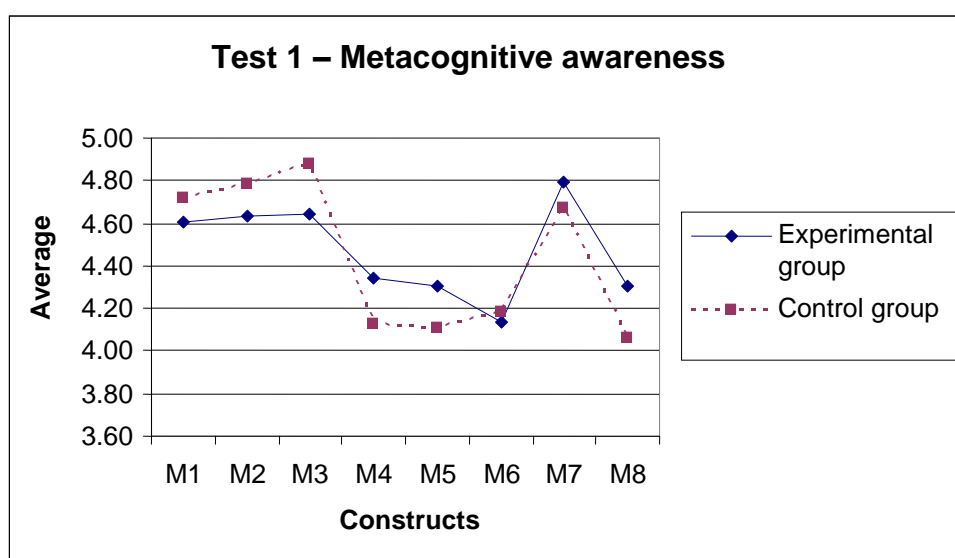


Figure 2. Averages of constructs regarding metacognitive awareness in test 1

In test 2 (at the beginning of the intervention period) the effect sizes that were calculated with the independent t-test again yielded insignificant or small differences between the experimental and control groups regarding most of the constructs of metacognitive awareness. The exceptions were M2 ($d=0.54$) and M3 ($d=0.48$) (see table 2). Albeit that these are not practical significant differences, figure 3 shows a drop in the averages of the experimental group, especially regarding M2 and M3, while the averages of the control group in test 2 were approximately the same as in test 1. As in test 1, the averages of M4, M5, M6 and M8 for both groups were still lower than the rest. The results of the individual interviews at this stage

did not help to clarify the drop in the averages of the experimental group in test 2. From the interviews it became clear that both groups, after completion of a task, did very little evaluation of the process, the degree to which the goals were reached, or the possibility of other options to solve the problem, as expressed by one participant: “We ran it (the program), so it was working. So we did nothing further.”

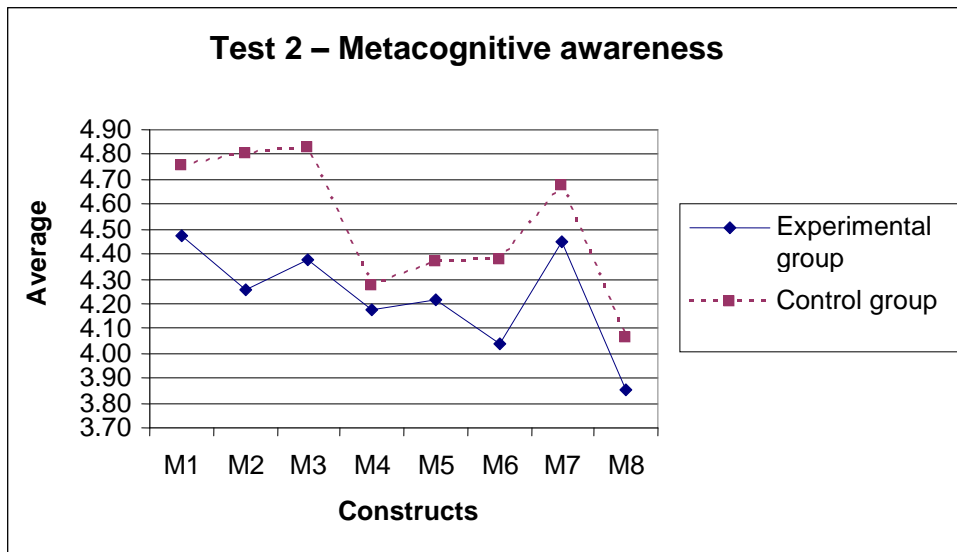


Figure 3. Averages of constructs regarding metacognitive awareness in test 2

The results of the independent t-test in test 3 (at the end of the intervention period) (see table 2) yielded differences of medium effect in constructs M4 ($d=0.36$) and M8 ($d=0.48$) between the experimental and control groups. The averages of the constructs in test 3, as in figure 4, show that the averages of constructs M4, M5, M6 and M8 were now higher than the corresponding averages of the control group. It is important to note that these constructs are the ones that bear relation to the use of the SDM questions that were used during the intervention period. Although the effect sizes did not show any practical significant differences between the two groups regarding the constructs of metacognitive awareness, figure 4 shows an increase in the averages of the constructs of metacognitive awareness for the experimental group, which may be attributed to the use of the SDM questions during pair programming.

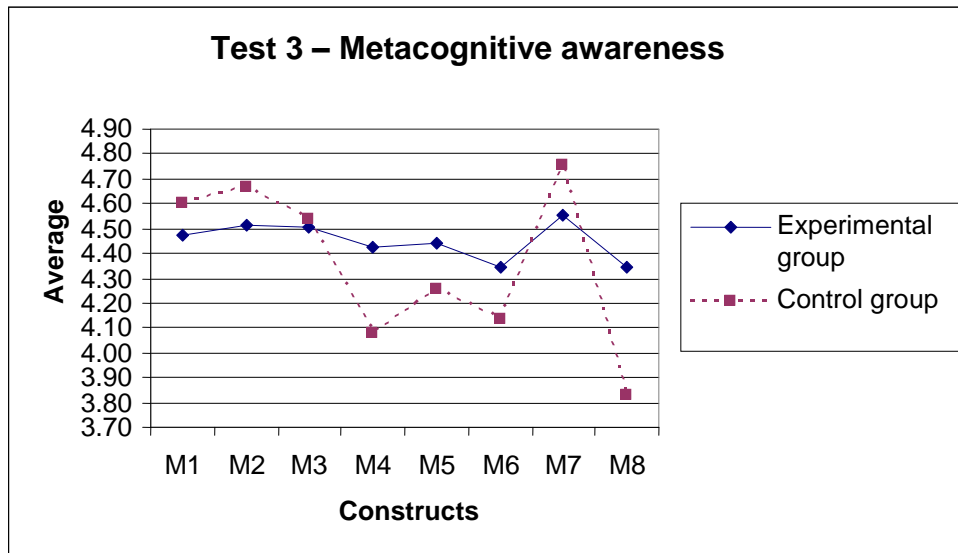


Figure 4. Averages of constructs regarding metacognitive awareness in test 3

From the results of the individual interviews it is also clear that the use of the SDM questions by the participants in the experimental group contributed to some extent to their metacognitive awareness. The following example illustrates the difference in response to a question regarding the participants' activities in the pre-coding phase. When asked about this during the individual interview in test 2, a participant said: "We discussed the problem." This same participant responded as follows during the individual interview in test 3 (after implementation of the SDM questions): "We discussed the problem to see what it was about. We went through the problem and then decided what strategies we are going to use and how we will apply them." Regarding their activities in the intra-coding phase, the participants in the experimental group responded during the individual interviews in test 3 with answers illustrating their monitoring of the process, for example: "If we got stuck we rethought our solution, read the problem again, and used the text book to help us solve the problem" and: "While my partner is coding I try to think about the coding and procedures and what still has to be done." When asked about their activities in the post-coding phase some of the participants in the experimental group indicated that they had done some evaluation of their solutions afterwards, for example: "We went through the program to see what we had done. We looked for another solution to see if we could get a better solution."

Although the results of the interviews could not fully explain or endorse the tendencies that were noted in the quantitative research, it was clear from the interviews that the use of the

SDM questions did influence the participants' awareness of planning (M4), information management strategies (M5), monitoring (M6) and evaluation (M8).

The results of the individual journals kept by participants in the experimental group during the intervention period showed that the participants had attended to more of the metacognitive activities in the course of the intervention period.

- 1) In the *pre-coding phase* they had especially given more attention to their decision making about the applicable strategies/procedures to be applied, as illustrated by the following entry: "We spent more time on deciding which procedures we were going to use in the program." This correlates with the increase in the average of the construct regarding planning (M4) that increased from 4.17 to 4.42 between test 2 and test 3.
- 2) During the *intra-coding phase*, the participants had given more attention to aspects like considering whether they were still on the right track and understood what they were doing, changed the planning of the solution if necessary, monitored their progress with the coding, and consulted applicable sources if they needed help. This change in approach is reflected in later entries such as "We made sure that what we do is what the problem requires from us", "If necessary, we changed our planning during the coding of the program", and "While we were programming we continually checked our progress". The increase in the application of these activities could explain the increases in the averages of the constructs regarding information management strategies (M5) and monitoring (M6). The averages of M5 and M6 changed from 4.21 to 4.44 and from 4.04 to 4.34, respectively, between test 2 and test 3.
- 3) The participants also indicated that in the *post-coding phase* they especially gave more attention to reflecting on the quality of their solution and what they could have done differently during the programming task that could possibly have led to a better solution, e.g. "At the end of the programming task we tested everything to see if it was the best way to write the program". This also is in accordance with the increase in the average of the construct regarding evaluation (M8) the increased from 3.85 to 4.34 between test 2 and test 3.

Multivariate statistics

Multivariate statistics were used to investigate the differences between the experimental and control groups regarding metacognitive awareness. Wilk's lambda tests were done and the effect sizes calculated for test 1, test 2 and test 3. The only practical significant difference

between the experimental and control groups were found in test 3 ($d=0.34$) (Steyn & Ellis, 2009).

Discussion

The purpose of this research was to investigate the influence of a metacognitive teaching-learning strategy during pair programming on the metacognitive awareness of the learners involved. Based on analysis of the data that was collected with the questionnaires, triangulated by analysis of the individual interviews and journals, a number of trends relating to specific constructs of metacognitive awareness, as well as the three phases of the pair programming task, were observed.

For the experimental group, the outcomes of the data analysis provided evidence of a trend that the use of the MTLs for pair programmers enhanced their metacognitive awareness, specifically regarding the problem-solving activities and strategies related to planning, information management, monitoring and evaluation. Little has been published regarding metacognition in pair programming specifically, but previous research have highlighted the improvement of metacognitive awareness and skills through a metacognitive teaching-learning approach to problem solving in mathematics (e.g. Cardelle-Elawar, 1995; Gama, 2004; Kramarski, 2008; Mevarech & Fridkin, 2006; Mevarech & Kramarski, 1997; Schoenfeld, 1991). Another finding of the project was that the learners evaluated their declarative, procedural and conditional knowledge, as well as their application of debugging strategies, higher than the constructs regarding planning, information management, monitoring and evaluation. This could indicate that the learners perceived that they lacked the skills related to metacognitive control. This tendency changed after the implementation of the MTLs during pair programming. The learners who had used the MTLs reported an increase in the application of activities and strategies regarding planning, information management, monitoring and evaluation. This increase may be explained as being an outcome of the implementation of the MTLs for pair programmers, which was specifically directed at the development of skills related to metacognitive control.

The SDM questions that were used in the MTLs for pair programmers were intended to direct the learners' focus to the metacognitive aspects of problem solving. Shraw (2001)

recommended the use of a checklist of self-directed questions to assist learners in implementing systematic control of their task execution. The SDM questions for the pre-coding phase of the pair programming task (e.g., “Do we understand exactly what outcomes we have to reach with the solution of this problem?” and “Do we know strategies/procedures that are appropriate for solving the problem?”) were intended to lead learners to consider clear goal setting, identification of important information, reflection on relevant aspects of prior knowledge, decisions on how the problem could be solved, and what strategies and resources to use. Focusing on these aspects during the pre-coding phase may have led to the self-reported increase in activities regarding the constructs on planning and information management. In the intra-coding phase of the pair programming task the SDM questions (e.g., “Do we continuously ensure that we understand what we are doing?” and “Do we think about how every step fits into what has already been done and what needs to be done next?”) were directed at regular re-consideration of whether what was being done was still understood, thinking about how each step fitted in the coding process, anticipation of what has to be done next, monitoring the progress that had been made, identification and correcting of errors as coding progressed, changing the planning, if necessary, and consultation of resources, also if necessary. Answering these questions during the intra-coding phase may have led to the increase in the learners’ perception of their monitoring abilities. The SDM questions that the learners had to answer during the post-coding phase of the pair programming task focused on metacognitive evaluation (e.g., “Did we check if the expected outcomes had been reached?” and “Have we thought about whether the best possible strategies/procedures were used to solve the problem?”). Metacognitive evaluation required learners, on completion of a pair programming task, to evaluate to what extent the set goals had been reached, evaluate the effectiveness of the strategies/procedures that had been used, think about what could have been done differently and if they had considered all their options, reflect on what they should have learned from the task, and reflect on what they had learned from the doing the task. The self-reported increase in activities related to the construct on evaluation after the intervention may be as result of the use of these SDM questions. These results are in line with earlier findings (Kramarski, 2004; Sandi-Urena, Cooper, & Stevens, 2011) that meaningful, purposeful collaboration between learners working in a group and simultaneous use of self-directed metacognitive questioning promotes metacognitive development.

Conclusion and final remarks

The literature study emphasised the importance of a metacognitive approach to problem solving during pair programming in order to enhance metacognitive awareness. Based on the literature study, the MTLs for pair programmers was suggested and then applied in the empirical study. The results of the empirical study fell into line with the findings of the literature review and indicated that the metacognitive awareness of the participating learners programming in pairs could be enhanced by implementing a metacognitive teaching-learning strategy. The fact that no statistically significant findings emerged from the project can probably be explained by the short duration of the project, since the development of metacognitive skills is a long-term activity, as emphasised by Veenman, Van Hout-Wolters and Afflerbach (2006). This aspect will have to be addressed in future research on the application of a MTLs during pair programming. Furthermore, the findings of this project cannot be generalised to learning situations other than pair programming.

The results were based on the self-reported metacognitive awareness of the participating learners. A question may be raised about the relation between the learners' self-reported metacognitive awareness and their real or actual metacognitive activities. Future research based on real-time observation of these activities (Veenman & Spaans, 2005) and the think-aloud pair problem solving method (Pate & Miller, 2011; Pate, Wardlow, & Johnson, 2004) might shed more light on this aspect, and bring about more insight regarding the influence of the MTLs for pair programming on these aspects.

The findings must be viewed against the background of the complex nature of metacognition. For the purpose of this research, metacognition was distinguished in a number of constructs, based on the literature review. The discrete measurement and discussion of the constructs entailed a rather analytical view of metacognition. It should, however, be kept in mind that metacognition is a set of complicated and integrated abilities that an individual possesses. The constructs, as employed in this research, should not be seen as discrete entities, but rather as collectively contributing to the individual's metacognitive abilities. In practice, individuals exercise these abilities in different ways and to different degrees. The implementation of a MTLs may thus bring about the development of different metacognitive abilities in different individuals.

This project, however, highlighted the importance of a metacognitive teaching-learning strategy when learners are programming in pairs. The findings could do much to advance teaching and learning in computer programming, and to inform future research and practice.

Acknowledgement

This research was based on work financially supported by the National Research Foundation (NRF) in South Africa. The NRF had no further involvement in the research. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF 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. doi: 10.1002/sce.20176
- Berenson, S., Slaten, K.M., 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. doi: 10.1145/1060071.1060074
- 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. doi: 10.1016/j.bbr.2011.03.031
- Cardelle-Elawar, M. (1995). Effects of metacognitive instruction on low achievers in mathematics problems. *Teaching and Teacher Education*, 11(1), 81-95. doi: 10.1016/j.bbr.2011.03.031
- Chalmers, C., & Nason, R.A. (2003, December). *Developing primary students' group metacognitive processes in a computer supported collaborative learning environment*. Paper presented at the AARE - NZARE Joint Conference, Auckland, New Zealand. Retrieved from <http://www.aare.edu.au/03pap/nas03501.pdf>
- Chaparro, E.A., Yuksel, A., Romero, P., & Bryant, S. (2005). Factors affecting the perceived effectiveness of pair programming in higher education. In P. Romero, J. Good, E. A. Chaparro, & S. Bryant (Eds.), *Proceedings of the 17th Workshop of the Psychology of Programming Interest Group, June 28-July 1, 2005*, 5-18. Brighton, UK: University of Sussex.

- Cliburn, D.C. (2003). Experiences with pair programming at a small college. *Journal of Computing Sciences in Colleges*, 19(1), 20-29.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. (2nd ed.). Hillsdale, NJ: Erlbaum.
- Creswell, J.W., & Plano Clark, V.L. (2007). *Designing and conducting mixed methods research*. Thousand Oaks, CA: Sage.
- Ferzli, M., Wiebe, E., & Williams, L. (2002). *Paired programming project: Focus groups with teaching assistants and students*. Technical Report. Raleigh, NC: North Carolina State University.
- Field, A. (2005). *Discovering statistics using SPSS*. (2nd ed.). London: Sage.
- Gama, C. (2004). Metacognition in interactive learning environments: The Reflection Assistant model. In J. C. Lester, R. M. Vicari, & F. Paraguacu (Eds.), *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, August 30-September 3, 2004*, 668-677. Berlin: Springer.
- Hofer, B.K., Yu, S.L., & Pintrich, P.R. (1998). Teaching college students to be self-regulated learners. In D.H. Schunk, & B.J. Zimmerman (Eds.), *Self-regulated learning: From teaching to self-reflective practice*, 57-85. New York: Guilford Press.
- Joseph, N. (2009). Metacognition needed: Teaching middle and high school students to develop strategic thinking skills. *Preventing School Failure*, 54(2), 99–103. doi: 10.1080/10459880903217770
- 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. doi: 10.1016/j.bbr.2011.03.031
- Kramarski, B. (2008). Promoting teachers' algebraic reasoning and self-regulation with metacognitive guidance. *Metacognition and Learning*, 3(2), 83-99. doi: 10.1007/s11409-008-9020-6
- 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. doi: 10.1023/A:1016282811724
- 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. doi: 10.1080/00220670109598765
- Kramarski, B., & Mizrachi, N. (2004). Enhancing mathematical literacy with the use of metacognitive guidance in forum discussion. In *Proceedings of the 28th Conference of the*

- International Group for the Psychology of Mathematics Education, July 14-18, 2004*, 169-176. Retrieved from http://www.emis.de/proceedings/PME28/RR/RR306_Kramarski.pdf
- McDowell, C., Hanks, B., & Werner, L. (2003). Experimenting with pair programming in the classroom. *ACM SIGSCE Bulletin*, 35(3), 60-64. doi: 10.1145/961290.961531
- McDowell, C., Werner, L., Bullock, H.E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, 49(8), 90-95. doi: 10.1145/1145287.1145293
- Mendes, E., Al-Fakhri, L.B., & Luxton-Reilly, A. (2005). Investigating pair-programming in a 2nd-year software development and design computer science course. *ACM SIGSCE Bulletin*, 37(3), 296-300. doi: 10.1145/1151954.1067526
- Mevarech, Z., & Fridkin, S. (2006). The effects of IMPROVE on mathematical knowledge, mathematical reasoning and meta-cognition. *Metacognition and Learning*, 1(1), 85-97. doi: 10.1007/s11409-006-6584-x
- 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. doi: 10.1080/00220679909597597
- Mevarech, Z.R., & Kramarski, B. (1997). A multidimensional method for teaching mathematics in heterogeneous classrooms. *American Educational Research Journal*, 34(2), 365-394. doi: 10.3102/00028312034002365
- Nicol, A.A.M., & Pexman, P.M. (1999). *Presenting your findings: A practical guide for creating tables*. Washington, DC: American Psychological Association.
- Niculescu, 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*, 69-97. Pretoria: Van Schaik.
- Palincsar, A.S., & Brown, A.L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, 1, 117-175. doi: 10.1207/s1532690xci0102_1
- Pate, M.L., & Miller, G. (2011). A descriptive interpretive analysis of students' oral verbalization during the use of Think-Aloud Pair Problem Solving while troubleshooting. *Journal of Agricultural Education*, 52(1), 107-119. doi: 10.5032/jae.2011.01107

- Pate, M.L., Wardlow, G.W., & Johnson, D.M. (2004). Effects of thinking aloud pair problem solving on the troubleshooting performance of undergraduate agricultural students in a power technology course. *Journal of Agricultural Education*, 45(4), 1-11.
- Preston, D. (2006). Using collaborative learning research to enhance pair programming pedagogy. *ACM SIGITE Newsletter*, 3(1), 16-21. doi: 10.1145/1113378.1113381
- Sandi-Urena, S., Cooper, M.M., & Stevens, R.H. (2011). Enhancement of metacognition use and awareness by means of a collaborative intervention. *International Journal of Science Education*, 33(3), 323-340. doi: 10.1080/09500690903452922
- Scardamalia, M., Bereiter, C., & Steinbach, R. (1984). Teachability of reflective processes in written composition. *Cognitive Science*, 8(2), 173-190. doi: 10.1207/s15516709cog0802_4
- Schoenfeld, A.H. (1991). On mathematics as sense-making: an informal attack on the unfortunate divorce of formal and informal mathematics. In J. F. Voss, D. N. Perkins, & J. W. Segal (Eds.), *Informal Reasoning and Education*, 331-343. Hillsdale, NJ: Erlbaum.
- Schraw, G. (2001). Promoting general metacognitive awareness. In H. J. Hartman (Ed.), *Metacognition in Learning and Instruction: theory, research and practice*, 3-16. Dordrecht: Kluwer.
- Schraw, G., & Dennison, R.S. (1994). Assessing metacognitive awareness. *Contemporary Educational Psychology*, 19, 460-475. doi: 10.1016/j.bbr.2011.03.031
- Steyn, H.S., & Ellis, S.M. (2009). Estimating an effect size in one-way multivariate analysis of variance (MANOVA). *Multivariate Behavioral Research*, 44(1), 106-129. doi: 10.1080/00273170802620238
- Vaicaityte, L. (2006). *Personal stumble-stones in the process of knowledge productivity in group work* (Master's thesis). University of Twente, Enschede.
- 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. doi:10.1016/j.lindif.2004.12.001
- Veenman, M.V.J., Van Hout-Wolters, B.H.A.M., & Afflerbach, P. (2006). Metacognition and learning: Conceptual and methodological considerations. *Metacognition and Learning*, 1(1), 3-14. doi: 10.1007/s11409-006-6893-0
- Williams, L., Kessler, R.R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE Software*, 19-25. doi: 10.1109/52.854064

- Williams, L., & Upchurch, R. (2001a, October). *Extreme programming for software engineering education?* Paper presented at the 31st ASEE/IEEE Frontiers in Education Conference, Reno, NV, USA. doi: 10/1109/FIE.2001.963882
- Williams, L., & Upchurch, R. (2001b). In support of student pair-programming. *ACM SIGCSE Bulletin*, 33(1), 327-331. doi: 10.1145/366413.364614
- Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the Introductory Computer Science course. *Computer Science Education*, 12(3), 197-212. doi: 10.1076/csed.12.3.197.8618
- Wilson, J., & Johnson, P. (2000). Students thinking about their learning: assessment to improve learning. *Educational research quarterly*, 24(2), 10-20.
- Zeidner, M., Boekaerts, M., & Pintrich, P.R. (2005). Self-regulation: Directions and challenges for future research. In M. Boekaerts, P.R. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation*, 749-768. San Diego, CA: Academic Press.

[Página en blanco por razones de paginación]