Exploring the Promotion of Self-Direction in Learning through a Metacognitive Approach to Pair Programming

Betty Breed
North-West University
Potchefstroom Campus
Telephone: (+27) 18 299 1475
Email: betty.breed@nwu.ac.za
Abstract

One of the assumptions associated with the promotion of self-direction in learning is that it can be increased through certain teaching-learning methods (Francom, 2009). This theoretical article argues the potential contribution of a metacognitive teaching-learning strategy for pair programming to increase programming learners’ self-direction in learning. Self-direction in learning refers to the “degree to which an individual possesses the skills and subject matter knowledge necessary to self-direct learning within a subject domain” (Francom, 2009). Since metacognition is viewed as one of the primary dimensions of the psychological processes involved in self-direction in learning (Long, 2000: 16), it is proposed that explicit development of metacognitive skills in order to enhance the metacognitive awareness of learners programming in pairs will simultaneously contribute to an improvement in their level of self-direction in learning.

Based on a review of literature on self-directed learning, the article first discusses Thornton’s (2010) framework for a self-directed learning cycle, which includes the four phases of planning, implementing, monitoring, and evaluating. This is followed by a discussion of a metacognitive teaching-learning strategy for pair programmers, aimed at enhancing the metacognitive awareness of learners when programming in pairs. The teaching-learning strategy under discussion focuses on the development of metacognitive skills related to executive control of one’s thinking processes. Finally, the way in which the elements of the metacognitive teaching-learning strategy for pair programmers tie in with the above-mentioned framework for a self-directed learning cycle is highlighted. Theoretically, the metacognitive teaching-learning strategy for pair programmers fits well into the framework for a self-directed learning cycle, and it can be concluded that this strategy could possibly be adapted to enhance learner self-direction in learning in other subject areas.

Key words: self-directed learning; metacognition; reflection; metacognitive teaching-learning strategy; self-direction in learning

Introduction

Learner self-direction in learning refers to the “degree to which an individual possesses the skills and subject matter knowledge necessary to self-direct learning within a subject domain” (Francom, 2009). A basic element of self-directed learning is that learners take responsibility for their own learning. Bolhuis (2003: 329) refers to this as learners taking “ownership” of their own learning process. This ownership can be fostered and reinforced by equipping learners with the learning skills, attitudes, and knowledge needed to carry out self-directed learning activities (Bolhuis, 2003: 329). According to Francom (2010: 29), learners’ self-direction in learning, and the skills related to this, can be fostered through the implementation of purposeful teaching and learning activities. The skills needed to carry out self-directed learning activities include, among other things, the metacognitive skills of planning, monitoring, and evaluating (Van Merriënboer & Sluijsmans, 2009: 58), as well as reflection (Francom, 2010: 32). Shannon (2008: 18) opines that “metacognition is the engine that drives self-directed learning”. For self-
directed learning to be successful it is essential to stimulate learners’ metacognitive development (Cotterall & Murray, 2009: 34). Shannon (2008: 18) recommends that teachers should explicitly develop learners’ metacognitive skills, by incorporating active reflective activities throughout the learning process, and by teaching learners metacognitive strategies as a valuable skill to become more self-directed learners. In the light of Shannon’s (2008) recommendation the purpose of this article is to investigate the potential contribution of a metacognitive teaching-learning strategy to increase learners’ self-direction in learning during pair programming.

To emphasise the usefulness of this investigation a pragmatic approach was followed. Pragmatism places the research problem in the centre (Creswell, 2003: 11) and uses methods that can help to understand the problem best, without advocating the philosophical views underlying the methods used (Tashakkori & Teddlie, 2003: 679).

This article first briefly discusses some viewpoints on the promotion of self-directed learning among learners, followed by a summary of Thornton’s (2010) framework for supporting self-directed learning. It then briefly discusses the promotion of metacognition in the classroom and explains a metacognitive teaching-learning strategy that has been developed for pair programmers. Thereafter, the way the elements of the above-mentioned teaching-learning strategy tie in with Thornton’s framework for a self-directed learning cycle is highlighted. Finally, the potential contribution of this particular teaching-learning strategy to increase learners’ self-direction in learning during pair programming is investigated.

Promoting Self-Direction in Learning in the Classroom

From the literature review, it would appear that there is no consensus among authors and researchers in the definition of self-directed learning. The differences may emanate from different viewpoints regarding the meaning and value of self-directed learning, and whether it is related to lifelong learning in educational settings, knowledge productivity in economic contexts, or the broader context of society being a multicultural global village (Bolhuis, 2003: 238). However, self-direction in learning seems to be generally viewed as taking control of one’s own learning and managing processes to achieve one’s goals, which is the viewpoint recognised in this article.

According to Bolhuis (2003: 335), it would appear that self-direction in learning is a characteristic or quest of the individual, with no external factors influencing it. Yet Brockett and Hiemstra (as cited by Williams, 2003: 89) stated that “self-direction in learning is a product of both the external characteristics of an instructional process and the internal characteristics of the learner, where the autonomous individual assumes primary responsibility for the learning experience”. Research has shown that in spite of the fact that learners’ ability to self-direct their learning within educational settings can be influenced by contextual factors, self-directed learning skills can be fostered by implementing appropriate teaching-learning strategies (Francom, 2010: 32).
Teachers and other educators can thus play a significant part in encouraging and developing their learners’ skills in self-directed learning. The skills needed for independent learning can be developed in the classroom by incorporating relevant activities and strategies in teaching and learning events (Thornton, 2010). Teachers need to provide learners with learning experiences that foster self-directed learning, get learners actively involved in their own learning process, and explicitly teach learners how to learn, while guiding their learning process (Francom, 2010: 29; Bolhuis & Voeten, 2001: 837, 852). Development of these skills will equip learners for their current educational endeavours, as well as for opportunities beyond their formal education (Dynan et al., 2008: 100).

Research conducted on the promotion of self-direction in learning in the classroom has produced a number of models, tools, and frameworks to assist teachers to foster self-directed learning. Grow’s (1991) Staged Self-Directed Learning Model is an example of one of the earlier models. This model serves as a tool for teachers to match the current stage of self-direction in learning of the learner, and to prepare learners to advance to higher stages, with guidance regarding the role of the teacher and the types of activities that are appropriate for each stage. Another more recent development in this field of research is the four general principles that Francom (2010: 33) identified for teachers to follow to foster learner self-direction in learning in the classroom. Based on a review of empirical and theoretical literature on self-direction in learning, self-regulation, teaching, and learning models, Francom (2010: 33) suggests that teachers should:

- match the level of self-directed learning required in learning activities to learner readiness;
- progress from teacher-directed learning to learner-directed learning over time;
- support the joint acquisition of subject matter knowledge and self-directed learning skills; and
- have learners practise self-directed learning in the context of learning tasks.

For the purpose of the investigation reported by this article another tool for supporting self-directed learning, namely Thornton’s (2010) framework for supporting self-directed learning, is discussed in more detail below.


Considering the methodological aspects of self-directed learning, Thornton (2010) developed a framework for language teachers that intend to foster the skills their learners need to become more self-directed in their learning. The framework was developed with language teaching in mind, but it may be just as applicable for developing self-directed learning in other subject areas. The framework comprises a cycle of four phases: planning, implementing, monitoring, and evaluating. Each phase includes activities that can be used to develop self-directing skills. The fact that reflection is engaged in in each phase indicates that the four phases are cyclic, and not a linear process of steps to be followed. Reflection in each phase requires of learners to critically reflect on their own learning, and to act accordingly.
Planning of learning

Thornton (2010) highlights five aspects that need to be attended to when learners plan their learning. The first of these aspects concerns analysing learners’ needs and deciding on learning priorities, taking into account what learners are interested in, and what they would like to learn. Being involved in the decision-making process may motivate learners to study independently, even outside of the classroom. The needs analysis can be done by means of a questionnaire administered by the teacher, or a written report about the proceedings of a group discussion. The next aspect is to analyse learners’ current skills in terms of the level thereof. Once learners have determined their shortcomings, they can focus on activities to address the shortcomings. This can, for example, be done by obtaining feedback and help from teachers or peers on subject-specific tasks. A record should be kept of activities in this regard.

From the learners’ analyses of their skills and shortcomings follows the third aspect of goal setting. Teachers can help learners in this regard by ensuring that the goals that they set for themselves are realistic and feasible. Being able to set their own goals according to the level of their skills, whether the goals be general goals or more manageable sub-goals, may lead to learners wanting to keep on studying in order to achieve their goals, thus contributing to their development as self-directed learners. The next aspect is for learners to choose resources that will be appropriate for achieving their goals. Once again, the teacher can guide learners in the selection and use of resources, in such a way that learners will stay focused on achieving their goals. The fifth and last aspect that learners must attend to when planning their learning is to formally make a plan. This could include their short-term and long-term goals, and an explanation of how they intend to achieve these goals. Learners are expected to give details about activities and resources they will use to achieve each goal within a certain time frame. Thornton (2010) recommends that a learning contract be set up between each learner and the teacher, in which learners commit themselves to specific study activities for a set period of time.

Implementing the learning plan

When implementing their learning plans, learners may prefer either more or less involvement by the teacher, depending on their current level of self-directedness. The more learners are able to independently use the resources they have chosen and implement the activities they have decided on, the less they need teacher support and guidance in implementing their learning plans. Thornton (2010) recommends that in such cases a learning record should be kept that can help learners, firstly, to monitor and evaluate their progress, and, secondly, to build up a portfolio of their work. If learners perceive that they are making good progress, and if they are satisfied with the quality of their portfolio, they may be motivated to continue learning. Such a learning record can be in the form of a log of the various different learning activities engaged in, or reflective diaries.
Exploring the promotion of self-direction in learning through a metacognitive approach to pair programming

Monitoring of process

Self-directed learning requires that learners have skills to effectively monitor their work and progress while executing learning tasks. The process of monitoring leads to the learning cycle being recursive and keeps learners focused on the goals they have set for themselves. Thornton (2010) suggests two types of monitoring to enhance self-directed learning when studying a language. The first type, mid-task monitoring, occurs while learners are busy doing the learning task. Learners should continually ensure that what they are doing and using contributes to the achievement of their goals. If not, it may be necessary to find new resources or change the learning activity. Continuing to do a task while using ineffective resources or engaging in ineffective activities may result in learners wasting time and not achieving their goals.

The second type of monitoring that Thornton (2010) suggests, namely study-balance monitoring, refers to a regular checking of how learners’ learning is progressing. In Thornton’s (2010) framework, this could constitute calculating the percentage of work that has been completed, using the knowledge that has been studied, or reviewing what has been studied within a certain period of time. Regular monitoring of this kind informs learners about their progress.

Being aware of their progress, or the lack thereof, may motivate learners to continue with the monitoring activities during their development of independent, self-directed learning. The teacher can be instrumental in the process, by supporting monitoring activities during the execution of learning tasks in class.

Evaluating the learning product and the learning process

After a certain period of time, in which learners have had the opportunity to practise self-direction in learning, evaluation has to be done. This fourth phase of successful self-directed learning concerns evaluation of what has been learned and the extent to which progress has been made. It also concerns the process that has been followed in learners’ endeavours to reach their goals. This can be done by comparing learners’ current level of skills in the goals that they set for themselves with the skill levels that they set for themselves during the planning process. The learning records that they kept during the implementation of their learning plans will most probably reveal whether they have made progress in achieving their learning goals. The outcome of this evaluation will determine whether it is necessary for learners to adjust their learning plans. Once again, the teacher can contribute by providing opportunities for evaluation activities in the classroom. This may develop the ability in learners to evaluate their learning when they are engaged in independent, self-directed learning.

Promoting Metacognition in the Classroom

Among the several different definitions of metacognition provided in the literature, Long (2000: 18) views metacognition as “being conscious or aware of how and what one is thinking”. This definition forms the basis of the discussion in this article on the promotion of metacognition in the classroom. Long
(2000: 16) considers metacognition to be one of the primary dimensions of the psychological processes involved in self-directed learning. This view is based on the hypothesis that self-direction in learning requires the individual to be constantly aware of “important aspects of the cognitive processes employed in learning” (Long, 2000: 18). All the activities involved in self-directed learning (for example, goal setting, planning to achieve the goals, and evaluating the learning process) require metacognitive awareness (Loyens et al., 2008: 418). While executing a learning task, learners also need to continually monitor their understanding of what they are doing (Hmelo-Silver, 2004: 241).

Promotion of learners’ ability to think about how and why they work in specific ways can be done by explicitly developing learners’ metacognitive skills (Schraw, 2001: 14). Learners should be taught how to implement metacognitive skills such as planning, monitoring, making adjustments when necessary, and evaluating (Schraw, 2001: 11). This is just as applicable when learners are working in groups. Group members who have well-developed metacognitive skills are able to make better contributions to the group, thereby assisting to complete the task of the group (Chalmers & Nason, 2003). By explicitly teaching learners metacognitive skills, learners working in a group are enabled to consider different strategies to solve a problem, to compare and evaluate strategies, and to select the most appropriate strategy (Kramarski et al., 2002: 241).

From the literature on effective metacognitive instruction, the following three principles have been identified (Veenman et al., 2006: 9):

- Metacognitive instruction must be embedded in the content matter, to ensure connectivity. This means that metacognitive activities should form an integrated, natural part of the learning process (Lin, 2001: 37);
- Learners must be informed about the usefulness of metacognitive activities, to make them exert initial extra effort. If this is not explained, it may happen that learners will never use the metacognitive activities that they have learned (Winne, 2005: 561); and
- Training in metacognitive skills should occur over a prolonged period, to guarantee smooth and sustained application of metacognitive activity.

Since the 1980s a number of metacognitive teaching-learning strategies have been proposed by different authors. The examples below, which illustrate these strategies, relate to the solution of mathematical problems. These examples are cited because of their resemblance to the solution of programming problems since the metacognitive teaching-learning strategy discussed in the following section was developed to be implemented by learners programming in pairs.

Schoenfeld’s (1991) metacognitive strategy involves heuristic methods for the solution of mathematical problems. Schoenfeld focuses on processes such as generation of alternative solutions, evaluation of alternatives to determine whether the problem can be solved in the available time and with the available
Exploring the promotion of self-direction in learning through a metacognitive approach to pair programming resources, and assessment and monitoring of progress made. He uses modelling, guidance, support, and collaborative problem solving in the class context, as well as in small group discussions. At the end of a session, the teacher and the learners collaboratively analyse what they have done, and why they did what they did. This approach focuses more on the process than the product of the problem solving (Bransford et al., 2000: 67).

Another example of a metacognitive teaching-learning strategy involves the research of Mevarech and Kramarski (1997). Their research set out to develop teaching strategies that use metacognitive processes to improve reasoning, particularly mathematical reasoning and problem-solving abilities (Kramarski et al., 2001: 292). Mevarech and Kramarski’s (1997) IMPROVE method combines metacognitive training with cooperative learning (Mevarech, 1999: 197) and emphasises reflective discourse in the groups by creating opportunities for each learner to be involved in reasoning through the use of metacognitive questions (Kramarski et al., 2001: 292). The IMPROVE method addresses all the teaching-learning phases of metacognitive training, namely Introducing the new topic to the class; Metacognitive questioning in small groups; Practicing; Reviewing; Obtaining mastery on higher and lower cognitive skills; Verifying; and Enriching. In the IMPROVE method, four types of metacognitive questions have to be answered by the learners, focusing on understanding the problem, finding relationships between previous and new knowledge, using appropriate strategies to solve the problem, and reflection on the process and the product (Kramarski & Mizrachi, 2004: 170-172).

The particular metacognitive teaching-learning strategy for pair programmers which is discussed in this article is discussed in the following section.

**A Metacognitive Teaching-Learning Strategy (MTLS) for Pair Programmers**

The metacognitive teaching-learning strategy (MTLS) for pair programmers developed by Breed (2010) is aimed at providing teachers with a teaching-learning strategy that enhances learners’ skills in metacognitive control when programming in pairs. Metacognitive control refers to the regulation of learning activities, including planning, monitoring, and evaluation strategies (Veenman et al., 2006, as cited by DiDonato, 2013: 27). Development of the strategy was based on an intensive literature study of explicit development and improvement of metacognition in the classroom (e.g. Palincsar & Brown, 1984; Scardamalia et al., 1984; Schoenfeld, 1991; Cardelle-Elawar, 1995; Mevarech & Kramarski, 1997; Schraw, 2001; Gama, 2004), keeping Veenman et al.’s (2006) principles for successful metacognitive instruction and the unique nature of pair programming in mind. In the context of education, pair programming is a cooperative teaching-learning strategy (Mentz et al., 2008: 259), where a programming task is engaged in by two learners working at the same computer, performing the roles of “driver” and “navigator”, respectively. The “driver” is responsible for typing or coding a design (Nicolescu & Plummer, 2003: 200), while the “navigator” is responsible for the strategic thinking required to solve the problem (Williams et al., 2002: 198).
The MTLS for pair programmers consists of four components to be used in the implementation of the MTLS. The first of these components is a *diagram representing the elements of metacognition* (Appendix A). This diagram is used by teachers to initially discuss the concept of metacognition with learners, to make them aware of the importance of metacognitive activities, and the advantages of such activities. Teachers are expected to emphasise the elements of metacognitive control, that is, planning, monitoring, and evaluation, in order to focus learners’ attention on the relationship between these elements and the phases generally associated with a programming task, namely pre-coding, intra-coding, and post-coding.

The second component of the MTLS for pair programmers is a *list of metacognitive activities* that should be attended to while collaborating in each of the phases of the programming task. Learners are required to read through the list each time before they start with a new pair programming task. The third component of the MTLS is a *set of self-directed metacognitive questions* (SDM questions) that the programming pair has to answer collectively during execution of a programming task. These SDM questions prompt learners to consider whether they have applied the metacognitive activities in the list mentioned above.

The use of the list of metacognitive activities and the set of SDM questions will be discussed in more detail in the following sections. The final component of the MTLS is an *individual journal* for each learner, to be completed individually after a pair programming task has been completed. The journal provides learners with the opportunity to individually reflect on their own thinking processes and activities during execution of the pair programming task.

The process of solving a problem through pair programming can be divided into three phases: the phase before any coding is done (pre-coding); the phase during which coding is done (intra-coding); and the phase after coding has been completed (post-coding). The MTLS for pair programmers aims to foster specific aspects of metacognitive control during each of these phases in any pair programming task.

**Pre-coding**

In the pre-coding phase, the programming pair collaboratively analyses the problem and designs an algorithm to solve it. The two programmers’ abilities and skills are combined, and jointly they have a larger pool of self-knowledge, knowledge of learning tasks, and knowledge of appropriate learning strategies (Williams & Kessler, 2003: 9). In this phase, the two programmers have to identify which of the information provided is relevant and important, they have to consider the requirements of the problem, and they have to formulate goals. They jointly reflect on their previous knowledge, select from their pool of appropriate strategies, and negotiate about the order in which the strategies will be implemented. Jointly they have a greater chance of effectively deciding on resources and planning a time schedule. Below are some examples of SDM questions that learners need to answer collectively during this phase, to guide them in implementing metacognitive activities:

“Have we thought about aspects of the problem that are already known to us?”

“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?”
“Have we thought about resources that can help us to solve the problem?”

**Intra-coding**
During the intra-coding phase, the driver’s and the navigator’s involvement in metacognitive activities differs to some extent. Both the driver and the navigator continually monitor whether they still understand what they are doing, whether they are still on the right track, and whether they are making progress. If the driver detects a problem with any of these aspects, it is communicated to the navigator requesting explanation of, for example, what needs to happen next, or what changes need to be made. Since it is the task of the navigator to determine the sequence in which strategies must be employed, identify errors, think about alternatives, make changes if necessary, and consult resources, the navigator continually has to reflect on the process and the product. The navigator thus monitors whether the process is running in line with the original planning. The navigator also monitors if the tempo of work is in line with the original time planning. Some examples of SDM questions that the pair needs to answer during the intra-coding phase, to guide them in implementing the relevant metacognitive activities, are:

“Do we continually make sure that we understand what we are doing?”
“Do we continually make sure that we are still on the right track?”
“Do we think about how each step fits into what has already been done, and what needs to be done next?”
“Do we monitor the progress being made with the coding?”

**Post-coding**
In this final phase of the pair programming task, the driver and the navigator are again involved in the same metacognitive activities. Jointly they determine whether their goals have been achieved, whether the strategies used were effective, and what could have been done differently, and they decide on possible adjustments which should be made when engaging in similar tasks in the future. They also have to reflect on what they individually and collectively have learned from the pair programming process. Examples of SDM questions that the pair needs to answer in this post-coding phase to guide them in implementing metacognitive activities relevant to this phase, are:

“Did we check whether 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 whether anything could perhaps have been done using a different method?”

In the following section, the MTLS for pair programmers and Thornton’s (2010) framework for supporting self-directed learning are compared, in order to investigate the extent to which the MTLS ties in with Thornton’s framework.
Comparing the MTLS for Pair Programmers and Thornton’s (2010) Framework

An important similarity between Thornton’s (2010) framework for self-directed learning and the MTLS for pair programmers is the fact that both of these tools have been designed to be used by teachers who wish to foster their learners’ learning skills, to make them more successful learners. In both of these tools, teachers have an important role to play. According to Thornton (2010: 29), teachers have the responsibility to support and guide learners to develop into self-directed learners and take control of their own learning. With regard to the development of metacognitive skills, teachers also have the responsibility to help learners to develop metacognitive skills, and to apply these skills in appropriate learning situations (Peters, 2000: 169).

Another similarity between Thornton’s (2010) framework and the MTLS for pair programmers is the role of reflection in all the phases of both tools. With both tools, learners are expected to continually reflect on their own learning and actions. Learners are required to act accordingly, and, if necessary, to make appropriate changes.

An examination of the metacognitive activities addressed in the three phases of the MTLS for pair programmers shows that these activities cover to a great extent the aspects that Thornton (2010) recommends in the four phases of her framework. The activities in the pre-coding phase of the MTLS for pair programmers guide learners’ thinking processes while they analyse the problem, reflect on previous knowledge, and design an algorithm. In Table 1, the suggested activities in the pre-coding phase of the MTLS for pair programmers and the planning of learning phase of Thornton’s (2010) framework are compared.
Table 1:
A comparison of the planning of learning phase of Thornton’s (2010) framework for self-directed learning and the pre-coding phase of the MTLS for pair programmers

<table>
<thead>
<tr>
<th>Thornton’s framework for self-directed learning</th>
<th>MTLS for pair programmers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning of learning</td>
<td>Pre-coding phase</td>
</tr>
<tr>
<td>☐ Analyse own learning needs</td>
<td>☐ Reflection on previous knowledge in relation to own experiences; aspects of the problem that are known already; similar problems solved previously</td>
</tr>
<tr>
<td>☐ Analyse current knowledge and skills</td>
<td>☐ Formulating goals to be reached with the solution of the problem</td>
</tr>
<tr>
<td>☐ Set goals</td>
<td>☐ Assigning appropriate resources</td>
</tr>
<tr>
<td>☐ Choose resources</td>
<td>☐ Selecting appropriate strategies/procedures</td>
</tr>
<tr>
<td>☐ Make a plan</td>
<td>☐ Designing an algorithm to solve the problem</td>
</tr>
<tr>
<td></td>
<td>☐ Deciding on a sequence in which the strategies/procedures will be applied</td>
</tr>
</tbody>
</table>

From the table, it is evident that the activities in the pre-coding phase of the MTLS for pair programmers cover four of the five aspects of the planning phase of Thornton’s (2010) framework. Reflection on previous knowledge may be seen as a way of determining and analysing one’s current knowledge and skills. The other three aspects of Thornton’s (2010) framework (setting goals, choosing resources, and making a plan) are clearly addressed by the remainder of the activities of the MTLS for pair programmers. With both tools, learners have to set their goals and choose appropriate resources. Making a plan to reach set goals can in the case of computer programming be described as selecting appropriate strategies, designing an algorithm, and deciding on the sequence of the strategies. The only aspect in the planning phase of Thornton’s (2010) framework for self-directed learning that is not addressed in the pre-coding phase of the MTLS for pair programmers is the issue of analysing own learning needs and deciding what one would like to learn. This may be explained in terms of the fact that the MTLS for pair programmers was designed to be implemented as a teaching-learning strategy during the normal teaching and learning of programming topics, as prescribed by the curriculum, and consequently does not make allowance for learners to decide for themselves what they would like to learn.

The activities included in the intra-coding phase of the MTLS for pair programmers serve the purpose of making learners aware of metacognitive activities involved when implementing their algorithm and monitoring the process. The intra-coding phase of the MTLS for pair programmers encompasses both the implementation and monitoring phases of Thornton’s (2010) framework. The programming pair has to implement their planned solution collaboratively, adhering to the roles of the driver and the navigator. Thornton’s (2010) framework subdivides the monitoring phase into mid-task monitoring and study-
balance monitoring. Table 2 presents the suggested activities in the *intra-coding* phase of the MTLS for pair programmers and the *implementation* and *monitoring* phases of Thornton’s (2010) framework.

**Table 2:**

A comparison of the implementation and monitoring phases of Thornton’s (2010) framework for self-directed learning and the intra-coding phase of the MTLS for pair programmers

<table>
<thead>
<tr>
<th>Thornton's framework for self-directed learning</th>
<th>MTLS for pair programmers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Implementing the learning plan</strong></td>
<td><strong>Intra-coding phase</strong></td>
</tr>
<tr>
<td>Monitoring the process</td>
<td></td>
</tr>
<tr>
<td>□ Mid-task monitoring of</td>
<td>□ Checking whether what is being done is still understood</td>
</tr>
<tr>
<td>- work progress</td>
<td>□ Checking whether the process still corresponds with the planning</td>
</tr>
<tr>
<td>- goal-directedness</td>
<td>□ Thinking about how each step fits in</td>
</tr>
<tr>
<td>- applicability of resources</td>
<td>□ Anticipation of what has to be done next</td>
</tr>
<tr>
<td>- necessity to change planning</td>
<td>□ Monitoring the progress that has been made</td>
</tr>
<tr>
<td></td>
<td>□ Identification and correcting of errors</td>
</tr>
<tr>
<td></td>
<td>□ A change in planning, if necessary</td>
</tr>
<tr>
<td></td>
<td>□ Consulting resources when necessary</td>
</tr>
<tr>
<td>□ Study-balance monitoring</td>
<td>□ Checking on the amount of time spent</td>
</tr>
</tbody>
</table>

The activities in the *intra-coding* phase of the MTLS for pair programmers tie in well with the aspects under consideration in the *mid-task monitoring* sub-phase of Thornton’s (2010) framework. The metacognitive activities in the MTLS provide for monitoring of comprehension, the process, and progress made. The activities also address monitoring related to a possible change in planning, if necessary, the use of resources, and work tempo. In the MTLS for pair programmers, the *study-balance monitoring* of Thornton’s (2010) framework is not applicable, since learners are expected to complete the pair programming task within a prescribed period of time.

The last phase of the MTLS for pair programmers, the *post-coding* phase, concerns evaluation of the product and the process of the programming task. Table 3 lists the metacognitive activities to be attended to in this *post-coding* phase and the activities in the *evaluation* phase of Thornton’s (2010) framework for self-directed learning.
Exploring the promotion of self-direction in learning through a metacognitive approach to pair programming

Table 3:
A comparison of the evaluation phase of Thornton’s (2010) framework for self-directed learning and the post-coding phase of the MTLS for pair programmers

<table>
<thead>
<tr>
<th>Thornton’s framework for self-directed learning</th>
<th>MTLS for pair programmers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluating product and process</td>
<td>Post-coding phase</td>
</tr>
<tr>
<td>□ Evaluation of</td>
<td>□ Evaluating the product to determine whether</td>
</tr>
<tr>
<td>- what has been learned</td>
<td>the expected outcomes have been achieved</td>
</tr>
<tr>
<td>- the extent to which progress has</td>
<td>□ Evaluating the extent to which goals have been</td>
</tr>
<tr>
<td>been made</td>
<td>achieved</td>
</tr>
<tr>
<td>- the process</td>
<td>□ Evaluating the effectiveness of</td>
</tr>
<tr>
<td></td>
<td>strategies/procedures that were used</td>
</tr>
<tr>
<td></td>
<td>□ Reflecting on what could have been done</td>
</tr>
<tr>
<td></td>
<td>differently</td>
</tr>
<tr>
<td></td>
<td>□ Reflecting on possible application of the same</td>
</tr>
<tr>
<td></td>
<td>processes to future problems</td>
</tr>
<tr>
<td></td>
<td>□ Reflecting on what was learned from solving</td>
</tr>
<tr>
<td></td>
<td>the particular problem</td>
</tr>
</tbody>
</table>

The evaluation done in the post-coding phase of the MTLS for pair programmers differs from the recommendations pertaining to evaluation made in Thornton’s (2010) framework, in the sense that pair programmers are expected to do evaluation after the completion of each pair programming task. Thornton (2010: 164), on the other hand, recommends that when implementing the framework for self-directed learning, evaluation should take place after learners have been undertaking self-directed learning for a reasonable period of time, for example after a month.

Conclusion
This investigation has the limitation that the MTLS for pair programmers was compared to Thornton’s (2010) framework for supporting self-directed learning only. Within this limitation, from the discussion of Thornton’s (2010) framework for supporting self-directed learning and the MTLS for pair programmers, and the comparison of the two tools, it appears that the MTLS for pair programmers fits well theoretically into this framework for a self-directed learning cycle. It is therefore proposed that explicit development of metacognitive skills in order to enhance the metacognitive awareness of learners programming in pairs might simultaneously contribute to an improvement in learners’ level of self-direction in learning, and might be beneficial to the development of self-directed learning skills. Further research including empirical investigations will have to be done to test this proposal.

Although the MTLS under discussion is specifically aimed at developing the metacognitive awareness of pair programmers, it seems that the strategy could possibly be adapted to enhance learner self-direction in
learning in the field of programming in general, and maybe also in other subject areas involving problem solving. This will also have to be investigated further in future research.


FRANCOM, G.M. 2009. Teach me how to learn: exploring assumptions behind efforts to foster learner self-direction. Paper presented at 23rd international symposium on self-directed learning, February, Cocoa Beach, FL.


Appendix A

**METACOGNITION**

**Self**
- Study preferences
- Factors that influence your performance
- Strengths and weaknesses
- Previous knowledge

**Metacognitive knowledge**
- Type
- Context
- Information
- Objectives
- Requirements
- Resources needed
- Application

**Learning task**
- Characteristics
- Syntax
- Applicability
- Transferability and adjustment
- Selection
- Application

**Learning strategies**
- Planning
  - Read and understand
  - Analyse
  - Interpret
  - Formulate objectives
  - Design
  - Resources
  - Time scheduling
- Evaluation
  - Comprehension
  - Task execution according to planning
  - Sequence of steps
  - Effectiveness of strategies
  - Learning progress
  - Application of resources
  - Time scheduling

**Metacognitive control**
- Monitoring
- Reflection