

A case study regarding teachers' problem-solving activities and approaches towards computer programming in diverse learning environments

Marietjie HAVENGA, Elsa MENTZ, Betty BREED
Faculty of Education Sciences, North-West University
Potchefstroom, South Africa

Desmond GOVENDER, Irene GOVENDER
University of KwaZulu-Natal
Durban, South Africa

Frank DIGNUM
Utrecht University
Utrecht, The Netherlands

Virginia DIGNUM
Delft University
Delft, The Netherlands

ABSTRACT

The main focus of this research was to explore six Computer Science teachers' use of problem-solving and programming activities and approaches in a case study. From the initial interviews, it is indicated that teachers have been using problem-solving skills in a rather random fashion, based on their intuition. The findings indicate that participating teachers' initial teaching/learning strategies were not as effective as one would expect. The results suggest that by applying problem-solving guidelines as part of an intervention, it directed teachers' teaching/learning activities in a more efficient way which, in turn, benefited the students.

Keywords: Computers, Diverse Learning Environments, Problem solving, Programming and Teachers.

1. INTRODUCTION

South Africa offers Computer Science/Information Technology (IT) as a selective school subject from grade 10 through grade 12. The main aim of this subject at school level is to enable students to use appropriate techniques and procedures to plan solutions and devise algorithms to solve problems [3]. A large number of students from diverse learning environments, specifically in schools with black students in South Africa, enroll annually for IT in grade 10. Despite the complex nature of this subject [13], it is a popular choice at many rural and suburban schools as students believe that working with computers will enhance their future career possibilities [10]. However, most of these students fail the national grade 12 IT practical programming examination.

This paper reports on research in progress which is aimed at training in-service IT teachers in rural and suburban schools to enable them to teach problem-solving and programming skills to their students as part of an intervention. This paper considers the following research questions:

1. Which problem-solving and programming activities and approaches are taught by IT teachers on how to solve programming problems? and
2. In what way does the teaching of detailed problem-solving steps and approaches enhance the solution of programming problems?

2. LITERATURE OVERVIEW

Various activities and approaches are involved in solving programming problems. An activity in this context refers to the ability to apply problem-solving thinking processes in order to achieve the aim of solving a programming problem. An approach refers to the intended actions on how to deal with the programming problem and indicate the way how to proceed in the process of solving the problem.

It is often assumed that teachers implicitly apply the required knowledge, skills and approaches to teach high-level problem-solving activities when teaching programming content. Merely giving enough exercises does not help if students are not taught how to follow proper problem-solving processes methodically [7]. According to Rist [12], the main reason why students find programming difficult is the lack of planning. Ismail, Ngah and Umar [9] identified the following aspects that should be addressed, namely lack of skills in analyzing problems, ineffective use of problem representation techniques, ineffective use of teaching strategies for problem solving and coding, and students' inability to apply programming constructs.

Since the nature of computer programming is a combination of science (reasoning, problem solving and critical thinking) and art (creative thinking, program design and development) [6], teachers need to teach a variety of thinking processes to scaffold students in their efforts. The teaching of comprehension, problem analysis, reasoning, program design, synthesis, evaluation and reflection are required [1, 4, 9]. These activities are part of procedural knowledge on 'how' to proceed

with the solution [14]. Ismail et al., [9] emphasize that students need to acquire reasoning and problem-solving skills before they learn how to apply and use various tools and programming languages. Explicit teaching of problem-solving and programming activities and approaches should therefore be directed towards ways of how to plan, represent, design and solve the problem at hand. The premise of this paper is that, regardless of the programming approach in use, detailed thinking processes, problem analysis and explicit teaching of problem-solving activities are required to support successful programming.

In the next section we described the empirical research to answer the research questions.

3. EMPIRICAL RESEARCH

Research design

Since this study focused on the teaching activities and approaches of IT teachers, a qualitative research approach was followed to develop a more detailed understanding thereof. A comparative case study research design [2] was employed to enable researchers to obtain data regarding six individual teachers' similar and different approaches and experiences on how they taught problem-solving activities to Information Technology students.

The purpose of a case study is to study a single or several individuals as part of an in-depth analysis of a bounded system (e.g. a group or an intervention) to gain insight, to discover and to interpret [2, 11] their experiences. In the context of this study the case selection criteria were the following:

- Information Technology teachers in economically deprived schools in two provinces in South Africa; and
- The teaching of grade 10 students.

Participants

Since there are not many schools in the economically deprived areas that offer IT as a subject, only six teachers were selected as participants in this study. This selection included three teachers at schools in the North-West province and three teachers in the KwaZulu-Natal province respectively, who are teaching IT to grade 10 students. One of the participating teachers in the North-West Province has a BEd with specialisation in IT, one an MSc with no teaching qualifications and one has an IT Diploma and is enrolled for the PGCE (Post-Graduate Certificate of Education). In KwaZulu-Natal, two teachers have a BEd with specialisation in IT and one teacher has an IT Diploma and is enrolled for BTech. studies. All of these teachers have at least three years of IT teaching experience.

Participation was entirely voluntary and all teachers completed informed consent forms. Permission to conduct this research was obtained from the North-West Department of Education, KwaZulu-Natal Department of Education, the ethics committee of the university where the study was conducted, as well as from school principals.

Research plan

This research comprised an initial semi-structured interview with teachers to determine their teaching of problem-solving activities, an intervention, and a semi-structured interview following the intervention.

Initial interviews: The first interview investigated teachers' problem-solving activities and approaches used before the intervention. The interviews were recorded to ensure that everything mentioned during the discussion was preserved.

Intervention: This comprised the following: training of participating teachers regarding the teaching of specific problem-solving strategies and activities [8] by using a manual consisting of firstly, problem-solving knowledge and activities, secondly, detailed problem-solving guidelines (Table 1) and thirdly, problem-solving and programming examples to support the teaching. The problem-solving guidelines include the following:

Table 1: Problem-solving guidelines

<p>1. Write down the main ideas and requirements of the problem. -Read the problem and underline all the important concepts to clearly understand and interpret the question; -Determine what you do not understand.</p>
<p>2. Represent the problem by using a diagram, table, flow chart, description or any other method to indicate how you understand and represent the problem.</p>
<p>3. Plan the detailed steps and mention the purpose and processes of each section or method. -Determine the purpose of each method; -Plan the input, processing and output; -Go back to Number 1 and check the planning of the solution.</p>
<p>4. Code your planning in a programming language. -Determine which code and/or constructs you will use to input the data; -Which statements will you use to process or calculate the data? -Which statements will you use to display the output? -Compile the program and correct the programming errors.</p>
<p>5. Reflect on how well you have solved the problem. -Use test data and ensure that the extreme cases of test data are included; -Explain if you could correct any programming errors; -Did you use resources to support your programming process? -How did you choose the test data and extreme values? -Are you satisfied with your solution? Explain. -Did you solve the problem?</p>

Interviews after the intervention: Semi-structured interviews that followed the intervention explored teachers' experiences regarding application of the problem-solving activities and guidelines (Table 1).

In addition, teachers were required to reflect in a journal on their experiences regarding the use of the mentioned guidelines as well as additional problems or issues. The journal served as an additional source of information.

The interviews comprised the following questions to teachers:

1. How would you describe your response regarding the use of the problem-solving guidelines?

2. How would you describe students' performance regarding the use of the problem-solving guidelines?
3. In which way does the use of the guidelines support students in their problem-solving and programming activities?
4. In which way can you enhance the use of these guidelines to support the learning process?

4. QUALITATIVE DATA ANALYSIS

After preparing the interview transcripts for data analysis, the textual information was manually coded and grouped according to a list of *a priori* codes – also known as concept-driven coding [5], to indicate pre-defined main ideas. These codes were based on specific questions used during the interview schedule, namely the explicit teaching of problem-solving strategies, application of problem-solving activities, and teachers/students' experience of using the problem-solving guidelines (see Section 3). Inter-rater reliability was applied where a colleague checked the categorisation of main ideas from the interview texts.

5. RESULTS

Interview results before the intervention

Initially the participating teachers used the following problem-solving strategies when teaching computer programming: explain (mentioned by 5 teachers), breaking down a problem into parts (3), algorithms (3), question analysis (1), problem analysis (1), analyze examples (1), scenarios (1), IPO tables (input, processing and output tables) (2), and creative programming (1). They also used additional approaches such as problem-based learning (1), cooperative learning (1), and questions and answers (1). Despite these efforts, teachers experienced problems in teaching effective problem-solving and programming skills. Some teachers indicated that their students had problems in applying their knowledge in a computer program.

Results after the intervention

Results from both within-case and cross-case data analyses [11] are mentioned in this subsection to address details regarding each individual participant and all participants respectively.

Within-case analysis: According to the participating teachers, using the mentioned guidelines supported the problem-solving process of students. Teachers' experiences are mentioned by referring to participants individually.

Participant 1: 'I went through it [guidelines] and tried to implement it.' 'If they [students] plan [they] get all the marks ... It helps a lot.' 'The more you plan, the more you understand the question.' 'They are more interested in [step] 4 [programming]. However, sometimes they still ... 'take shortcuts' and 'want to rush.' 'Time is a factor ... I think that with time they will embrace it as we move on.'

Participant 2: 'I have found that it is working, but there are [a] few learners who are still struggling with the syntax.' 'You see the problem is [that] we do not give marks for the planning phase, so they do the practical [programming] to get marks.' Some concerns were the time to complete the guidelines' steps in class and 'not all students implemented it.'

Participant 3: 'I was busy applying these [problem-solving guidelines] but we had to rush for the exams.' 'I think in the next term it will be more useful, we will have more time to practice all those things.' 'It 'enforces critical thinking,' 'makes a big difference [and] ensures that students refocus on the problem in hand.' However, 'some [students] were lazy.'

Participant 4: 'I am pleased with the way the strategy [guidelines] is working.' 'It gave them [students] a good understanding in terms of how to tackle a problem.' This participant also mentioned that students applied the planning when they started doing a new programming problem.

Participant 5: 'It is working ... they can learn some of the things from the problem part.' 'The method I was using was just difficult. Since I have introduced it, it just made life easier for them.'

Participant 6: 'It is working. I am used to group them [students], and if I see there is still a problem, I just intervene and help them.'

Cross-case analysis: With reference to the interview questions in Section 3, the following results are organized according to specific questions from the semi-structured interviews after the intervention, to indicate some similarities across the cases.

Question 1: How would you describe your response regarding the use of the problem-solving guidelines? 'It is ... no problem' [Participant1 (P1)]. '... here the planning is very important. Yes, it is working' [P2]. 'It is very good because it forces them [students] to think critically about issues' [P3]. 'I find it very interesting ... they have to know everything about the first step to move to the next step, until you [they] reach a solution...' [P4]. 'Yes I think it is working, because I have seen the difference' [P5]. '... the one [guidelines] I am using now, this is much easier' [P6].

Question 2: How would you describe students' performance regarding the use of the problem-solving guidelines? Some examples: '... normally they go straight to number four [the programming], now at least they ... can see that if I plan, ... I am going to do the rights things' [P1]. 'The planning is very important' [P2]. 'Once you break it down into simple steps, they start to ... like it, it is something that they can relate to' [P3]. 'Yes.' [The strategy is working] [P4]. '... they are able to understand what is expected from them' [P5]. 'It is much better for them to understand' [P6].

Question 3: In which way does the use of the guidelines support students in their problem-solving and programming activities? 'It helps them a lot ... now they are actively involved' [P1]. 'Yes, to some extent it supports them' [P2]. 'There are more chances for us to do revising of the work when doing those steps' [P3]. 'I think this is the strategy that I will use when I introduce problem solving' [P4]. 'Since I have introduced it, it just made life easier for them, they are enjoying it' [P5]. 'Yes, they are enjoying it' [P6].

Question 4: In which way can you enhance the use of these guidelines to support the learning process? Some participants indicate ways to enhance the use of these guidelines. 'I would suggest that we start with ... an algorithm and then we go to number 4 [programming]' [P1]. 'The activity should start by planning and then they have to do programming' [P2]. 'Use colorful pictures ... a poster and reduce it to less detail' [P3].

6. DISCUSSION

This section answers the research questions.

Question 1: Which problem-solving and programming activities and approaches are taught by IT teachers on how to solve programming problems?

Results from the initial interview indicate that five of the six teachers had some educational background and this was evident in their knowledge of skills and the use of some problem-solving strategies. Some examples being: breaking down a problem into parts or subproblems, applying IPO tables and using algorithms. However, it seems that the teaching of these activities and approaches were randomly applied as needed in class without their purposeful application. The lack in the teaching approaches of the participating teachers requires additional training to enable them to teach effectively. Overall, these teachers encountered problems to direct students' thinking processes, to analyse, present and apply explicit problem-solving processes as mentioned by Ismail et al. [9] and Hasni and Lodhi [7]. These findings indicate that they were unable to guide students during various steps of problem solving. It can be deduced from the results that these teachers needed additional training and support to enhance their teaching performance.

Question 2: In what way does the teaching of detailed problem-solving steps and approaches enhance the solution of programming problems?

Reflection on the problem-solving guidelines

The activities and guidelines were designed to primarily address the following problematic issues: to follow a methodical way (step-by-step) [7] and to include aspects such as planning [12], problem analyses, representation [9], program design, implementation into programming code [4] and reflection [1]. These guidelines are based on a strategic approach to direct the structural organisation of thinking processes and to address both the overarching program as well as detailed methods and events.

Teachers' experiences and students' performance

The teachers were asked to apply the problem-solving activities and approaches when teaching programming skills. By using these guidelines, they directed teachers' teaching/learning activities, supported the learning process, enforced critical thinking and clarified terms by means of underlining all the main concepts in the programming problem. Furthermore, it supported students to focus and address the main issues in the problem. Some teachers mentioned challenges and referred to the time to implement these activities.

Within-case and cross-case analyses

Participants mentioned different experiences regarding the problem-solving activities. Participant 1 mentioned a positive experience when using the guidelines, since previously some students wanted to take shortcuts and started with the programming immediately. Both Participants 2 and 3 referred to the time frame as a problem to apply the guidelines and steps in class. To solve this, teachers may require students to do the planning at home prior to the next lab session. By using the mentioned activities and approaches, they guide students, make it easier to solve problems [P5] and enhance understanding of the problem [P6].

Similarities between cases are also indicated when teachers mentioned that students who applied the guidelines, performed better than their planning and programming attempts prior to the use of the steps and guidelines. The problem-solving activities and approaches supported students to plan their programs in detail, think critically, apply various steps before implementing the problem in a program, and optimize the programming process.

7. CONCLUSION

We found that our premise, as stated in Section 2, has been vindicated. The findings indicate that by applying the problem-solving guidelines and detailed thinking processes as part of the intervention, participating teachers' approaches and teaching activities were directed in a more efficient way to enhance successful programming. Teachers indicated that students preferred the structured approach to problem solving, and overall they felt that this improved their problem-solving skills.

8. ACKNOWLEDGEMENT

This research is based on work financially supported by SANPAD – South African Netherlands Project for Alternative Development. Any opinion, findings and conclusions or recommendations expressed in this paper are those of the authors and therefore SANPAD does not accept any liability in this regard.

9. REFERENCES

- [1] S. Bergin, R. Reilly & D. Traynor, "Examining the Role of Self-Regulated Learning on Introductory Programming Performance", **International Computing Education Research (ICER)**, 2005, pp. 81-86.
- [2] J.W. Creswell, **Educational Research. Planning, Conducting, and Evaluating Quantitative and Qualitative Research**, (3rd Ed.), New Jersey: Pearson Merrill Prentice Hall, 2008.
- [3] Department: Basic Education. Republic of South Africa. Curriculum and Assessment Policy Statement (CAPS). Information Technology, 2011.
- [4] A. Ebrahimi & C. Schweikert, "Empirical Study of Novice Programming with Plans and Objects", **The SIGCSE Bulletin. Inroads**, Vol. 38, No. 4, 2006, pp. 52-54.
- [5] G. Gibbs, **Analyzing Qualitative Data**, Edited by U. Flick, Los Angeles: Sage, 2010.
- [6] R. Green & H. Ledgard, "Coding Guidelines: Finding the Art in the Science", **Communications of the ACM**, Vol. 54, No. 12, 2012, pp. 57-63.
- [7] T.F. Hasni & F. Lodhi, "Teaching Problem Solving Effectively", **ACM Inroads**, Vol. 2, No. 3, 2011, pp. 58-62.
- [8] M. Havenga, "Problem-solving processes in computer programming: a case study". *South African Computer Lecturers' Association (SACLA) Conference Proceedings*. ISBN: 978 1 86840 721 7, 2011, pp. 91-99.

- [9] M.N. Ismail, N.A. Ngah & I.N. Umar, "Instructional strategy in the teaching of computer programming: A need assessment analyses", **The Turkish Online Journal of Educational Technology**, Vol. 9, No. 2, 2010, pp. 125-131.
- [10] E. Mentz, R. Bailey, M. Havenga, B. Breed, D. Govender, I. Govender, F. Dignum & V. Dignum, "The diverse educational needs and challenges of Information Technology teachers in two black rural schools", **Perspectives in Education**, Vol. 30, No. 1, 2012, pp. 71-79.
- [11] S.B. Merriam, **Qualitative Research and Case Study Applications in Education. Revised and Expanded from Case Study Research in Education**, San Francisco: Jossey-Bass, 1998.
- [12] R. S. Rist, "Teaching Eiffel as a First Language", **Journal of Object-Oriented Programming**, Vol. 9, No. 1, 1996, pp. 30-41.
- [13] J. Sajaniemi & M. Kuittinen, "From procedures to objects: A research agenda for the psychology of object-oriented programming education", **Human Technology**, Vol. 4, No. 1, 2008, pp. 75-91.
- [14] R.J. Sternberg, **Cognitive Psychology**, (4th Ed.). United States: Thomson Wadsworth, 2006.