Using Critical Social Research and Project-based Learning to Emancipate Students in Computer Programming Classes with Different Skill Levels

Roelien Goede
North-West University, South Africa

Abstract

In computer programming classes in higher education institutions one often finds students of different abilities, competencies and motivation in a single class. Often this leads to a feeling of frustration experienced by students and lecturers. This paper demonstrates how problem based learning can be used to provide a stimulating environment for students with different competencies. Action research is used from a critical social research perspective to improve the programming skills of the entire group of students. The aim is to emancipate students and the lecturer by providing a stimulating learning environment for all students while accommodating their differences in experience and skill levels.

1. Introduction

Programmes in Information Technology at universities comprise of diverse modules covering different aspects of the industry. Denning provides a summary of competencies required by information technology professionals covering different aspects of the industry [1]. Some modules in a degree programme focus on the more technical aspects such as computer networks, operating systems and computer programming, while others focus on the more human aspects such as systems analysis and design. This diversity is also present in the student population: some students are very good at programming and other students prefer the human-oriented modules in their course. However these students are thrown together into one classroom and the lecturer is confronted with the task of facilitating a learning environment that stimulates learning to all.

This diversity in students' ability to programme can be linked to many factors such as previous exposure to computer programming, aptitude, and interest. Most information technology programmes focus on computer programming in the first and second year of study. Diversity of student competencies is present in both years. Often the group of students who are not interested in programming has more problems in the second year of programming studies, as the gap between them (the "non-programmers") and the "programmers" widened after their first year of study. The research project reported in this study involves second year students in a Bachelor of Information Technology degree programme at a South African university.

It is important to view three parties that need to be emancipated: "non-programmers," "programmers" and the lecturer. The "non-programmers" should be guided in order for them to develop into better programmers. Persons who understand both the technical and the human aspects of Information Systems are in great demand as they are able to form an interface between the technical IT department and the work force in the organization.

"Programmers" should be emancipated as often they do not receive added value in programming modules. Many of them taught themselves to program and they deserve value for the tuition fees in terms of a stimulating learning environment.

The lecturer in a class with students of diverse skills and interest are always torn between helping the "non-programmers" to pass and simply ignoring them. They also seldom have the opportunity to create a challenge for the "programmers".

The aim of this study is to provide a learning environment that facilitates improvement of programming skills of both "programmers" and "non-programmers". The study is guided by action research from a critical social research perspective.

The paper commences with a discussion of critical social research in Section 2. Intervention from a critical social perspective is based on existing theory, and project-based learning is used as such and discussed in Section 3. The empirical research is discussed in Section 4 according to the phases of action research. Section 5 presents concluding remarks and future work.

2. Research Methodology

In this section critical social research is presented as research paradigm for this study and action research is used as research methodology.

2.1 Critical Social Research

Klein and Myers discuss three ontological distinct research paradigms in Information Systems, namely positivism, interpretive research and critical social theory [2]. Their paper focuses on interpretive case studies where the aim is the understanding of the problem environment. Positivistic research aims at measuring variables in a problem environment by an objective researcher [2]. In this research project the aim is to change the environment to facilitate learning for students of diverse levels of programming competencies. This aim corresponds to a critical social theory perspective where the ontological assumptions focus on oppression and disharmony and the epistemologically knowledge is obtain through intervention aimed at identifying and

relieving oppressive structures in the problem environment.

In this problem situation the oppressed parties are viewed as the different groups of students in the class as well as the lecturer, acting as facilitator of learning. As discussed in the introduction, the group of students who enjoy programming often is neglected and not stimulated as the lecturer focuses his or her attention on the other group. The students who are not interested in programming are frustrated since they feel of lesser value than the others. The lecturer is also affected negatively since he or she is often criticized by one or both these groups.

In another more recent paper by the Myers and Klein, they present principles for critical social theory research in the field of information systems [3]. Although the research project reported here started before this publication, the principles provided by Myers and Klein are supported by this study. Table 1 provides a reflection on these principles from the position of this research project.

Table 1. Reflection on the Principles of Myers and Klein [3]

Principle	Reflection
1. The principle of using core concepts from critical social theorists	The critical thinker Ulrich [4] highlights the roles of the involved and the affected. In this study the students are affected with very little power and the lecturer – although also affected – has the power to change the teaching environment in order to emancipate the students and him- or herself.
2. The principle of taking a value position	The position taken here is that students who are interest in programming are not "better" or more important than others but they should also not be neglected or left to their own devices in a programming class.
3. The principle of revealing and challenging prevailing beliefs and social practices	Many of the "non-programmers" have very low expectations of improving their skills, while the "programmers" often have the same feeling but for a different reason. They feel that they will be neglected in the class room.
4. The principle of individual emancipation	Every student should improve his/ her programming ability and should experience a sense of belonging in the classroom.
5. The principle of improvements in society	Improved programming skills of information systems graduates should lead to improved information systems in industry that should in turn lead to improvements in society.
6. The principle of improvements in social theories	This project demonstrates the validity of using work from Ulrich [4] in educational action research projects.

2.2 Action Research

The origin of action research can be linked to the work of Lewin [5]. It is used as a research method to guide a researcher to improve a problem situation by affecting participative change. Although it is not always done explicitly from a critical social research perspective as in this paper, it lends itself to be used as such. Baskerville describes the two main part of action research as diagnosis and intervention [6].

Figure 1 from Baskerville indicates the phases of action research used in this study [6]. Action research is a cyclic process which only is completed once the aimed intervention is successful. This pragmatic approach facilitates the inclusion of research methods often associated with the other research paradigms.

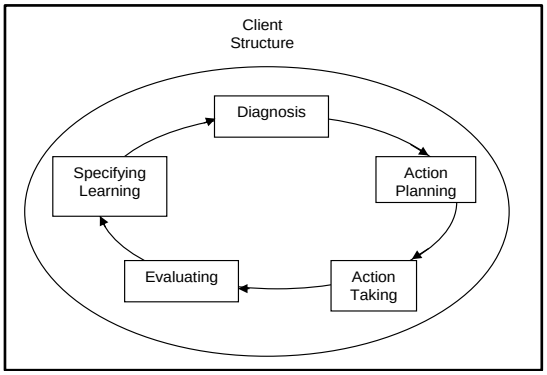

Figure 1. Action Research [6]

The cyclic process starts with a diagnosis of the problem situation. From a critical social perspective this is aimed at identifying oppressive structures in the environment. Part of action planning is to apply existing theory in planning and designing an intervention to relieve the oppression. In this study project-based learning is used. The action taking phase is used to affect the desired change. The success of the intervention should be evaluated. In this study it was done using interpretive interviews with students. Specifying learning is aimed at identification of the contribution of the empirical project. This is often done on three levels: the area of application, the methodology used, and the framework of understanding [7].

Before the detail of the empirical research project is discussed a brief literature review is provided on problem-based learning since it guided the planning of the intervention.

3. Problem-Based Learning

The aim of this section is to provide a brief introduction to problem-based learning (PBL).

Thomas provides a detailed discussion on PBL and a review of PBL literature [8]. For the purpose of this discussion the six distinguishing characteristics of PBL as defined by Helle will be used to introduce the reader to PBL [9]:

1. Learning should be project orientated implying that the project drives or motivates the learning activity of the students.
2. The students have to create an artifact. This is often done according to a set of steps where learners discover gaps in their own knowledge and do research to solve their own problems.
3. The learners take control of their learning process by searching for solutions to their problems.
4. Learning is contextualized in terms of the project and therefor students are confronted with the actual implementation of the artifact.
5. Students are exposed to multiple forms of knowledge representation in terms of sources and they develop skills to integrate knowledge from various forms of sources such as graphs, videos, and text.
6. Students are intrinsically motivated to achieve success in their project, since they take ownership of the learning experience.

Often authors accentuate the "learning by doing" aspect of PBL but Barron also emphasizes the "doing with understanding" aspect of PBL [10]. The solution to each step of the problem should equip the students with skills to be more successful in the next step. PBL is often linked to self-directed learning (SDL) as described by Knowles [11]. It is evident in SDL literature that different students have different levels of SDL-skills. Grow provides a model of this phenomenon. The level of control taken in principle 3 above is linked to the SDL-level of the student [12].

4. Improving the Programming Skills of the Diverse Group

The aim of this section is to present the empirical part of the research project aimed at improving the programming skills of all students in a diverse programming class. After a short discussion of the context of the research, it is organized according to the phases of an action research project as demonstrated in figure 1.

4.1 Background

The research was conducted over a six year period where a module in data structures was presented to second year students in a Bachelor of Information Systems programme at a university in South Africa. The author of the paper is the lecturer responsible for the module. The second semester data structures module is designed to round off the programming knowledge of the students after they have completed four previous programming modules focusing on technical programming skills and graphical user-interface programming.

Students are confronted for the first time with abstract data types and methods and they should incorporate all their existing knowledge of object-orientated programming to master the skill to develop of programs of higher usability and economic use of resources. The module also covers the asymptotic analysis of algorithms.

4.2 Diagnosis of the Problem

Often a formal diagnosis phase in action research uses methods such as semi-structured interpretive interviews to understand the perception of various stakeholders in a problem-environment. In this instance no formal interviews took place. An understanding of the nature of the problem was developed by the researcher as a result of many personal discussions, initiated by unsatisfied students. These discussions took place in lecture rooms after lectures and in the office of the lecture with students who were either bored from a lack of stimulation or students who felt overwhelmed by the course material and believed that "they will never be programmers".

When the situation is reflected on using the work of the critical systems thinker Werner Ulrich, one has to reflect on the assumptions made by various parties in the problem environment [4]. As a critical social scholar the researcher reflected on the discussions with students and formulated the following characteristics of the problem from a critical social perspective:

- Some of the students do not expect to excel but rather expect to struggle with the module based on their perceived lack of programming interest and poor achievement in previous modules.
- Some of the students do not expect to be stimulated by the classes. They expect the level of lecturing to be below their level of knowledge focused on struggling students.
- The lecturer finds it difficult to accommodate both groups in the class and since the focus of the university is on throughput of students, the group of struggling students is prioritized.

According to Ulrich there are involved parties, who are able to change the situation and affected parties, who are not part of the process but are affected in the process [4]. The lecturer has the power to change the situation in terms of instructional design but needs to function in the environment of different skill levels. In systems thinking the environment of a system are those factors that influences the system but cannot be

controlled by the system. In this case the lecturer cannot control the initial interest and skills of the students in programming. Another environmental factor is the structure of the programme in which the module is presented. Contact time and assessment structures are to some extend planned and limited by faculty rules.

Ulrich stresses the role of a guarantor when planning an intervention [4]. There must be some proof that the planned intervention will succeed. In other words the experts in the problem situation must be sure to a certain extend that the planned intervention will lead to successful results. In this instance, the advantages of PBL are viewed as contributing to the success of the intervention.

4.3 Action Planning

Since the information technology industry is project-driven, it was decided that PBL would be a natural choice as teaching strategy to emancipate both group of students. The idea that different students have different levels of SDL skills was also taken into account in the design of the project.

The choice of project is crucial in this situation. It should provide a challenge to the students with good programming skills without demotivating the struggling students. The technical outcomes of the course should also be achievable. After many projects were analyzed and found to be unsuitable the researcher decided on the implementation of the intuitive method of solving a Sudoku puzzle.

Background of the programming strategy to solve the Sudoku puzzle is now provided. Sudoku is a puzzle game where a grid of 9 by 9 cells is partially completed. The player has to complete the missing numbers by applying simple rules:

Use the digits 1-9 to fill the spaces provided that no digit may appear twice in any row, column or predetermined 3x3 block.

Figure 2 represents a Sudoku puzzle and its solution [14].

The students have to develop a computer program (the PBL artifact) to solve a given Sudoku puzzle, not using brute force computer methods, but programming algorithms they use when they do it manually.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Figure 2. A Sudoku Puzzle and its Solution

For reference purposes, Figure 3 presents row, column, and block identification numbers.

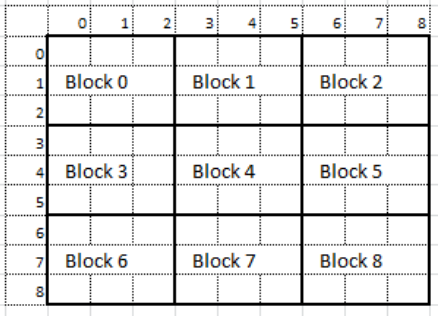

Figure 3. A Sudoku Grid with Identification Numbers

The basic strategy of this solution is to update the possible values array in each cell and to make an allocation when only one value is possible for that cell. After any allocation was made, the possible values of all the affected cells are updated again and the process is repeated. Affected cells are those in the same row, column or block as the one that was changed. The cell class contains methods to handle simple operations on cells such as setting a specific value in the possible values array equal to zero.

This process of updating the possible values of the cells after an allocation is a good exercise in

nested loop programming for the students. The given grid (left side of Figure 2) is given as input for to the program. Figure 4 presents a depiction of the cell in row 1, column 1 after the initial allocation was made. Note that since the array index starts with position 0, the value in position *i* represents the digit *i*+1.

possible_values:

index	0	1	2	3	4	5	6	7	8
possible_values: *values*	0	2	0	4	0	0	7	0	0

cell_row 1

cell_col 1

cell_value 0

Figure 4. Variable Values for Cell (1,1)

The fact that Java starts array indices at 0 rather than 1 complicates the implementation, but is a good learning experience for the students. From figure 4 is clear to see that the cell (1,1) (the centre cell in Block 0) may be a 2,4 or 7.

The main strategy is implemented by an algorithm that counts the entries in the array not equal to 0 (this method is in the cell class). If there is only one possible value not equal to 0, an allocation can be made to the specific cell. This method is called from the solution class for each cell in a nested loop with an exit condition if a cell is found with only one possible value. The allocation is made to the cell that only has one possible value and the affected cells need to be updated.

This project satisfies the content outcomes of the modules in that:

- It uses a variety of data structures for the two dimensional grid and the list of possible values in each cell and objects and classes for the cells.
- It is focused on developing algorithms. Students should stimulate their own intuitive problem solving by developing algorithms in the program.
- The solution should be able to handle different data types. Children often play Sudoku with colors instead of numbers. The program should be able to use colors or alphabetical characters as well as the traditional digits. In programming this feature is implemented using abstract data types and methods.
- The students should analyze the running time of their programme and are encouraged to prove the effective use of resources regarding running time and memory usage.

The instructional designed is done according to milestones of a typical project environment. Only the first few programming milestones of the project are given to the students initially. They are:

1. Create an object orientated class for each cell in the Sudoku grid and create the "solution manager" handling the 9x9 grid of cells. No solution should be done, only the initial setup of the game.
2. Display the possible values for each empty cell given the basic rules of Sudoku where no digit may be repeated in a row, column or pre-defined 3x3 block in the grid.
3. Provide a first solution by assigning a value to empty cells with only one possible value. The program segments developed in milestones 1 and 2 are used to show the state of the solution.
4. Create an iterative process to incorporate any changes made in milestone 3. The students are provided with Sudoku puzzles which can be solved in full by applying the rule represented in milestone 3.

Most students are able to achieve this milestone with relative little assistance as they have mastered the programming techniques required in the prerequisite modules of the module in data structures.

After evaluation of the efforts of the students, they are provided with a proposed solution for the problem up to this stage. The given code is well-document with comments linking the code to the technical aspects of the course content.

After milestone 4 the students can select the level of assistance they want to receive in the development of a solution for more challenging Sudoku problems. Students may choose to take full control of the project. They are required to provide documentation on their iterative approach to solving Sudoku puzzles. Students who prefer receive more guidance in identifying their own intuitive strategies for solving Sudoku puzzles. Milestones are jointly developed to implement each strategy.

A strict completion date for each milestone is given to the students. Students have the option to receive a proposed solution for each milestone after their efforts were evaluated. Formal contact sessions focus on theory of data structures and algorithm development and analysis. Additional contact sessions are scheduled after completion of milestones to discuss solutions to milestones and students are encouraged to reflect on and share their experiences.

4.4 Action Taking

The project is implemented as described above. Care is taken to motivate the students at the beginning of the project.

The students who are good programmers normally take a lot of pride in their own Sudoku

solving skills and can't wait to showcase them in terms of a computer program.

The students who have low self-esteem in their own programming ability need more motivation. It is important to convince them that when one divide the problem into smaller pieces only basic programming skills, which they have mastered, are required.
Results of previous students are used to this effect. They are normally convinced after the completion of the first milestone that the milestone method will help them to achieve the full solution.

The feeling of achievement at the end of the project is immense in all cases.

4.5 Evaluation of Intervention

Each of the six years of the project can be viewed as a separate iteration of the action research project. Each year the researcher (and lecturer) takes great care to understand the attitude of the students towards the module. A semi-structured set of open-ended questions are given to the students at the beginning and end of the semester. Data are analyzed by means of interpretive content analysis. Students also have to quantitatively evaluate the module as part of the university's formal quality control measures. Student opinions are formed also by students who completed the module and this was evident from the third year. Students knew what to expect and all the students had higher level of confidence. Stronger students looked forward to the module.

After the third year a change in evaluation of student efforts was required. Some students obtained solutions to milestones provided to previous groups and did not put in enough effort of their own. They were then required to give an oral demonstration of their efforts to an assistant lecturer who asked them several technical questions on their solution. This problem remains an obstacle and it was decided to develop five such projects which can be rotated from year to year.

In the formal summative evaluation of the module, students are expected to apply their knowledge of the concepts covered by the module to other problems. Care is taken to always provide "fresh" problems in the examination. From the analysis of the students' interviews it is clear that some students are looking forward to the challenge of the examination and are more confident than others.

4.6 Specified Learning

Learning or contribution from the action research takes place on three levels according to the FMA model of Checkland and Holwell depicted in Figure 5 [7].

The first level is the area of application (A of FMA). Improvement is visible of the attitude and performance of all students in the data structures module in the specific Bachelor of Information Technology programme.

Some level of emancipation did occur since the "programmers" experience a challenge and freedom to "impress". The "non-programmers" were able to keep up and felt a sense of achievement from the milestone organisation of the module. The lecturer experienced a better balance in time allocation to the different groups of students.

The second level in the FMA framework is the methodology (M) used. This project demonstrates how milestones and PBL can be used to emancipate students of different levels of competence through

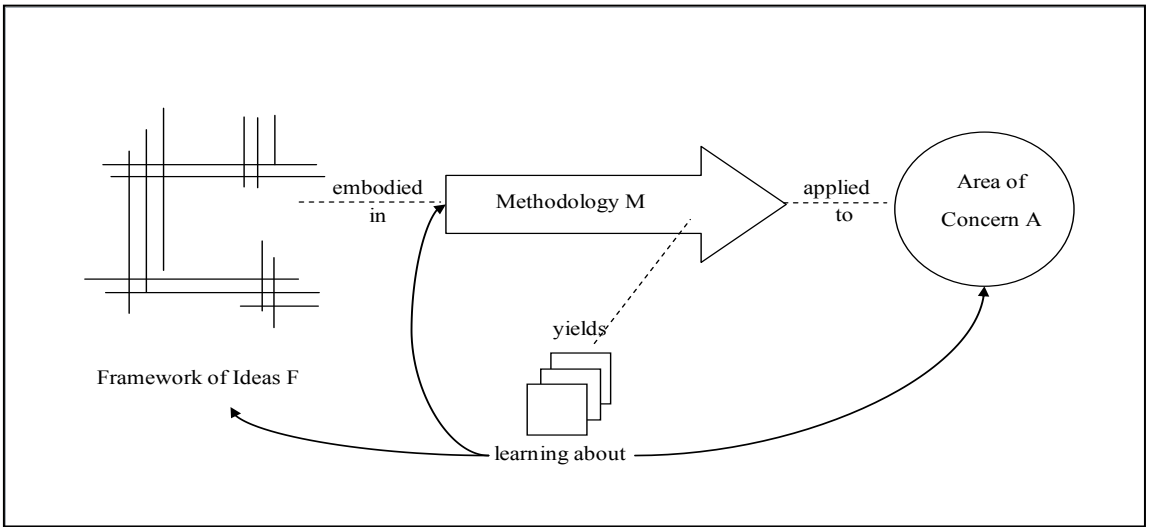

Figure 5. The FMA Model of Checkland and Holwell [7]

accommodation in an instructional design. The intuitive solution of Sudoku in an iterative model can be viewed as an example of the recent work on learning through developing intermediate cognitive models of Miwa [13]. It highlights the pitfalls of repeating an instructional design for consecutive academic years.

A third level of contribution is on the level of framework for understanding (F of FMA). The reader should take note that, as promoted by Myers and Klein, critical social researchers should guide their actions by using critical social theories. In this case the ideas of the critical systems thinker Werner Ulrich were used. By identifying the assumptions of the involved and affected the lecturer was able to focus on the specific needs of the different groups of students [4]. It also helped to identify which factors belongs to the environment of the system and which can be controlled by the system.

5. Conclusion and Future Work

Project-based learning (PBL) provides a teaching strategy to accommodate students of diverse competencies in a single instructional design. PBL is intrinsically linked to self-directed learning (SDL). Students should take control of their own learning objectives. The running of this project in the past six years has shown how students are proud of their achievements to provide unique strategies for the solution of Sudoku.

The author has decided that although the availability of solutions provided by previous students has been managed by the instructional design, it would be advantages to rotate the specific problem from year to year. Several new projects will be selected and developed to be used in future.

6. References

[1] P.J. Denning, "Great principles of computing." *Communications of the ACM*, ACM, London, 2003, 46(11), pp.15-20.

[2] H.K. Klein, and M.D. Myers, "A set of principles for conducting and evaluating interpretive field studies in information systems", *MIS Quarterly*, Management Information Systems Research Center, Minnesota, 1999, 23(1), pp. 67-94.

[3] M.D. Myers and H.K. Klein, "A set of principles for conducting Critical Research in Information Systems", *MIS Quarterly*, Management Information Systems Research Center, Minnesota, 2011, 35(1) pp. 17-36.

[4] W. Ulrich, *Critical heuristics of social planning: A new approach to practical philosophy*, Haupt, Bern. 1983.

[5] K. Lewin, "Frontiers in group dynamics". *Human Relations*, Sage, London, 1947, 1(1) pp. 5-41.

[6] R.L Baskerville, "Investigating information systems with action research" *Communications of the AIS* AIS. Atlanta, 1999, 2, article 19.

[7] P. Checkland, and S. Holwell, *Information, Systems and Information Systems making sense of the field*. Wiley, Chichester, 1998.

[8] J.W. Thomas, "A review of research on PBL". 2000 http://www.bobpearlman.org/BestPractices/ PBL_Research.pdf Date of access: 24 August 2014.

[9] L. Helle, P Tynjala and E Olkinuora, "Project-based learning in post-secondary education – theory, practice and rubber sling shots." *Higher Education*, Springer, New York, 2006, Vol. 51, pp. 287–314.

[10] B.J.S. Barron , D. L. Schwartz , N.J. Vye, A. Moore , A,Petrosino, L. Zech and J.D. Bransford, "Doing With Understanding: Lessons From Research on Problem- and Project-based Learning", *Journal of the Learning Sciences*, Routledge, London, 1998 7:3-4, pp. 271-311.

[11] M.S. Knowles,. *Self-directed learning*. Association Press, New York, 1975.

[12] G.O. Grow, "Teaching learners to be self directed" *Adult education quarterly*, Sage, London, 1991, 41(3) pp. 125-149.

[13] K. Miwa, J. Morita, R. Nakaike and H. Terai, "Learning through intermediate problems in creating cognitive models", *Interactive Learning Environments*, Routledge, London, 2014, 22:3, pp. 326-350.

[14] Anonymous, Sudoku, Wikipedia, accessed on 11/12/2014 at http://en.wikipedia.org/wiki/Sudoku.