World Applied Sciences Journal 20 (1): 34-39, 2012
ISSN 1818-4952
© IDOSI Publications, 2012
DOI: 10.5829/idosi.wasj.2012.20.01.1778

A Case Study on Improving Problem Solving Skills of Undergraduate Computer Science Students

Nadeem Ghafoor Chaudhry and Ghulam Rasool

Department of Computer Science,
COMSATS Institution of Information Technology, Lahore Campus, Pakistan

Abstract: Every human being should possess problem solving skills in order to reach one's desired goals in life. Problem solving skills become even more important when sustainable solutions are desperately needed to cope with twin problems of growing scarcity of resources and increasing populations. Practical problem solving skills, with clear economically viable, timely and implementable solutions are highly valued. Unfortunately the traditional curricula of most Universities, does not teach these skills up to the level which is desired by employers. The immediate and short term purpose of this research is to study and improve the problem solving skills of undergrad Computer Science (CS) students. In this paper, the long term objective is to learn the tools and techniques by which it can be achieved and then employ those in other disciplines also. The paper highlights three main folds of the research areas, firstly problem solving skills play a significant role in technical capabilities of people in general but more so for students studying in applied science and engineering areas, such as CS, secondly, problem solving skills can be "measured" and thirdly, problem solving skills can be improved with exercising and training.

Key words: Problem Solving Skills · DECSAR · Concept Mapping · Programming · Computer Science

INTRODUCTION

The importance of problem solving skills cannot be underestimated and compromised. A lot has been written about them [1-8]. It is beyond the scope of this paper to present a comprehensive review of all the literature regarding problem solving skills. Furthermore, this paper focuses on evaluating and improving problem solving skills of undergraduate CS students, which narrows the scope. For this reason a lot of problem solving skills literature related to school children and other disciplines like Chemistry, Physics, Biology etc in not in the scope of this paper.

According to Michalewicz *et al* [9], "A problem exists when there is a recognized disparity between the present and desired state. Solutions, in turn, are ways of allocating the available resources so as to reduce the disparity between the present and desired state".

In [10], Martinez defines problem solving as a "process of moving towards a goal when the path to the goal is uncertain". The paper differentiates between heuristics and algorithms. Algorithms are sequence of steps that are guaranteed to achieve the desired result every time we execute them in the proper order. For example taking a shower involves going to bathroom, taking off clothes, turning on the shower, rubbing soap and etc. Whereas heuristics are strategies to achieve a goal and are not guaranteed to work. For example reaching a friend's home when exact route is not known. Once a problem is solved by employing some heuristic and the steps of the solution are mastered then it is no longer a problem but an algorithm. Once reached at friend's home and memorize the route then going there next time is an algorithm not a problem. When original problem require solution again in future, "novel elements or new circumstances must be introduced or the level of challenge must be raised."

We present a controlled case study with major focus on evaluating and improving problem solving skills of undergraduate Computer Science Students in this paper. The rest of paper is organized into following six sections:

Corresponding Author: Ghulam Rasool, Department of Computer Science ,
COMSATS Institution of Information Technology, Lahore Campus, Pakistan.

A) Importance of Problem Solving Skills

Importance of Teaching Problem Solving Skills at Undergraduate Level: Michalewicz *et al* in [9] pointed out that technology has made it possible for us, rather forced us, to interact with an ever-increasing number of people. It means that the ramifications of our solutions to our problems affect more and more people. Our solutions might have irrevocable consequences in future, both in environmental and personal terms. Environmental consequences of bad solutions are quite obvious these days. People face the problems of air pollution; contaminated water etc. Candy *et al* [11] talks about transition of students from university environment to the real world working environment. In this paper, we point out that in addition to other shortcomings of fresh graduates, the employers frequently complained about their lack of problem solving skills. Authors attribute employer's complaints to four main differences between university and workplace learning. Out of these four, the third difference is of relevance in this section. They term the third difference as the difference between "manipulation of symbols" and "contextualized reasoning". According to them, in university reasoning is at a very abstract level, i.e students are taught the "manipulation of symbols" with little or no regard for the end purpose or its practical value. In real world the situation is completely inverted. The reason for this difference is because in real world only those problems are taken up and discussed which have a business value. So practical problem solving skills, with clear economically viable, timely and implementable solutions are highly valued. Unfortunately the traditional university curricula impose artificial and arbitrary limits. This is not what happens in the real world where problems have no limits and require unique solutions.

Similarly O'Leary [12] examined the graduate employment statistics of Australia. Her findings of the survey revealed that the top three "characteristics considered by employers to be desirable for effective workforce participation" included communication skills, capacity to learn new skills and procedures and capacity to make decisions and solve problems. We found it surprising that professional knowledge was considered to be of less importance than these characteristics.

B) Importance of Teaching Problem Solving Skills to Undergraduate Cs Students: Most of the core Undergraduate CS courses such as programming, algorithms and software engineering require students to be good problem solvers [13]. Programming in particular requires good problem solving skills. Every time one writes a new program one is involved in a "process of moving towards a goal when the path to the goal is uncertain" [10]. This is true for novice programmers than experienced programmers. Experienced programmers have developed certain heuristics and problem solving skills [5].

[14,15] highlighted the problems that students face while learning computer programming languages. In these papers, authors presented a list of about 27 problems that students faced in the introductory programming course. The most frequent problem was basically mechanics in nature, i.e missing semicolon (;) and/or other typos. The second highest recurring problem that students faced was "they understand the task / solution but can't turn that understanding into an algorithm, or can't turn the algorithm into a program". Through practical experience of teaching course of Data Structures and Algorithms, we found that some students feel it easy to write directly programs using computer but they face problem in writing algorithm for same program on paper. This dilemma also occurs due to curriculum of some universities that teach programming course first and then they offer Data Structures and Algorithm.

Even though the authors do not mention it as a problem solving problem but it is very clear that in reality it is. Students understand what is the task they have been assigned and they know what is the expected solution but they cannot "turn that understanding into an algorithm, or can't turn the algorithm into a program". This is the core problem of novice programmers. They have difficulty in "moving towards a goal when the path to the goal is uncertain" [4].

Method

A) Measuring Level of Problem Solving Skills: Measuring level of problem solving skills in individuals is by itself, quite an involved problem. A lot of research work has been conducted on social problem solving inventories. Authors in [1] developed Social Problem Solving Inventory which they later enhanced to Social Problem Solving Inventory-revised (SPS-R). The inventory is not available publically so we purchased inventory for conducting our experiment. The same inventory has been used by several other researchers [2, 4] for evaluating problems solving skills of students. The inventory consists of 52 statements. These statements are grouped into 5 scales : Positive Problem Orientation (PPO), Negative Problem Orientation (NPO), Rational Problem Solving (RPS), Impulsive/Carelessness

Style(ICS), Avoidance Style (AS) and 4 subscales Problem Definition and Formulation (PDF), Generation of Alternative Solutions (GAS), Decision Making (DM), Solution Implementation and Verification (SIV). Each scale and subscale is meant to measure one specific aspect of problem solving skill.

B) Technique and Tool Used to Improve Problem Solving Skills: DECSAR [10] is a formal technique of problem solving. "The DECSAR Method is a six-step troubleshooting strategy that is designed to model effective troubleshooting". The six steps are:

- Define the problem
- Examine the situation
- Consider the Causes
- Consider the Solution
- Act and Test
- Review the troubleshooting

The starting point of any problem solving activity is the definition of the problem. It is a concise, one or two sentence long statement that states what is the problem. The next step is a situational analysis. It is basically a list of facts about what is working and what is not, in the context of the problem being solved. Both are important because "what is working" list will help us in limiting possible causes of problem so that we don't go on a wild goose chase. On the other hand the "what is not working" list will point us to possible causes of the problem, things that might be contributing to the problem.

Based upon facts listed in the above lists we can come up with a list of possible causes. For example if I turn on a light switch but the bulb does not light up, it is possible that the bulb is burned out or there is no power. But if the fan is running then we can eliminate the possibility of power outage.

Once we have listed possible causes in a priority sorted order, we can think about possible solutions based upon the listed causes. The important thing about solutions is that they should be viable or doable under the given circumstances. This constraint includes financial, social, ethical, environmental, technological and other aspects. For example many leather tanneries in Pakistan solve their problem of contaminated water by simply draining it out of their premises. For sure it solves their problem, but it is a solution with devastating consequences for people living around their tanneries. There- for it is not socially, ethically and environmentally good solution.

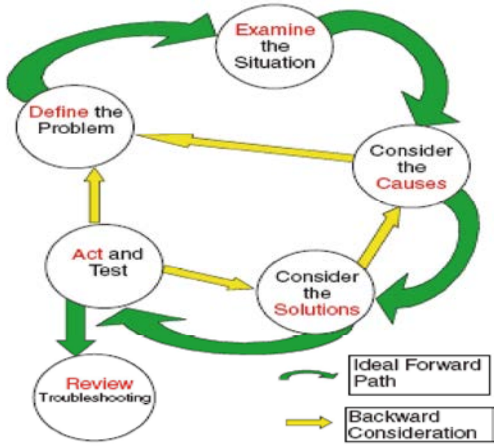

Fig. 1 : Six steps of DECSAR [3].

After deciding upon the best solution under the given circumstances, it is time to implement and verify that it has produced the desired results or not. If it has not produced the desired results then we can always follow one of the backward consideration paths. We don't have to go back all the way to the first step. If the problem is solved, then we know that our solution was correct, but we still have to perform one more step. Figure 1 gives a graphical representation of how these steps are to be performed.

Concept Mapping [16] is a tool for teaching and learning new concepts. Novak [16] based his theory on Ausubel's Assimilation Learning Theory. It is beyond the scope of this document to describe concept mapping in detail. Readers can refer to [16] and/or [17] for details. However a very brief overview is presented next.

The basic building block of concept mapping is "concept". According to Novak [6] a concept is a perceived regularity in events or objects, or records of events or objects, designated by a label. For example the perceived regularities of the concept of solid are : 1) Does not flow 2) Retain its shape 3) May have a crystalline structure or irregular structure. Each of these regularities by itself is a concept, e.g shape is a concept, which has its own regularities like two-dimensional shape or three-dimensional shape etc. In a concept map, concepts are represented with rectangular boxes and are linked with each other by using linking words to form propositions as shown in the Figure 2.

Strategy, Population and Sampling: The purpose of this research was to investigate and evaluate the effect of teaching DECSAR and Concept Mapping in improving problem solving skills of students. It was an experimental

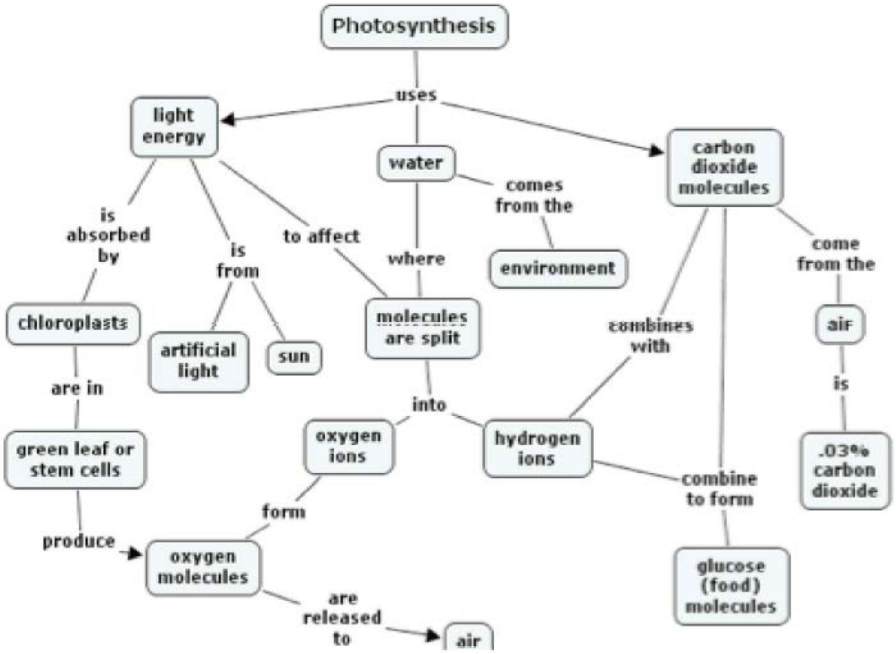

Fig. 2: An Example Concept Map for Photosynthesis [1].

study in which the researchers taught DECSAR and Concept Mapping to one group (experimental) but not the other (control). At the end, problem solving skills of the experimental and control groups were compared to see if the treatment made any difference in problem solving skills of both groups. For this purpose, a inventory with 52 questions/statements was distributed to students for evaluating and comparing their problem solving skills.

The population was the undergraduate students of CS department in Comsats Lahore campus. The sample was the group of students taking Object Oriented Programming from the first author of this paper. This course is offered in third semester of undergraduate CS studies. All the students taking this course were made part of the study. They were divided into control and experimental groups in a two-step process.

There were two sections of the class. In the first step the class rosters of both sections were combined and sorted in an ascending order, based upon student ID. In the second step this merged roster was split into two by putting every alternate student in the experimental group, with remaining students making up the control group. Thus there were approximately 30 students in each group. Unfortunately all of them cannot complete the course and a few did not fill the questioner. As a result, the final count of students in the control group was 19 where as 17 students in the experimental group stayed through the whole experiment. To provide equal motivation to both groups, authors had taken prior permission from the Head of CS Department that 5% of the final grade of every student will depend upon his/her participation in the experiment. The control group was given some extra work to cover for the 5% grade.

RESULTS

Even though the creation of the groups was based upon random selection, we compared the two groups for SPS-R scores. Table 1 below shows the result of t-test performed at the beginning of the semester between the control and experimental groups. The details on t-test are available in [18]. The entries in Table 1 are based on results collected through statements/questionnaire at the start of semester and t-test performed on control and experimental group. The last column shows the consolidated value that is combined effect of all factors mentioned in Table 1.

It is obvious from the Table 1 that apart from DM there is no significant difference between the two groups. This gave the confidence that the distribution of students between two groups was random and that the differences between the two groups at the end of semester were due to the treatment.

Similarly, Table 2 shows the results of T-test performed at the end of semester between the control and experimental groups.

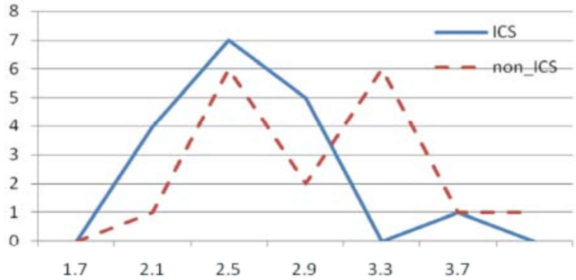

Fig. 3: Comparison of CGPA of students based on ICS score

Table 1: Result of t-test between control and experimental groups at start of semester

PPO	NPO	PDF	GAS	DM	SIV	RPS	ICS	AVS	Cons
1.27	-0.47	1.97	1.41	2.37	1.11	2.01	-0.24	0.08	1.23

Table 2: Result of t -Test between control and experimental groups at the end of semester

PPO	NPO	PDF	GAS	DM	SIV	RPS	ICS	AVS	Cons
2.669	-2.24	2.477	2.004	1.13	2.162	2.398	-0.38	-0.17	2.036

Table 3: CGPA based comparison of students below and above average score on ics

PPO	NPO	PDF	GAS	DM	SIV	RPS	ICS	AVS
-1.61	0.021	-1.01	-2.05	-0.69	-0.96	-1.45	-2.46	-1.73

One very interesting scale of inventory is ICS. Rodriguez-Fornells *et al* in [2] demonstrated that ICS is a very good predictor of subsequent academic achievement, better than some of the entrance exams used by universities in Spain. Even though the primary purpose of the present research was not ICS but the results obtained were significant enough that the author decided to include it. The table below presents t-test to compare CGPA based on the scales and subscales of inventory. As shown in Table 3, only ICS showed significant difference.

A simulation-based representation of the comparison based on ICS is shown in Figure 3. X-axis represents CGPA while Y-axis represents number of students in Figure 3. Authors calculated the average value of ICS (consolidated, i.e both experimental and control combined) to obtain the real time data. Then split the combined sample into two groups, one with below average score of ICS and the other with above average score of ICS. Even though the non-ICS curve (representing students with lower score of ICS) is bi-modal but one can clearly see that it is right shifted as compared to that of ICS group. This means that non-ICS students had significantly higher CGPAs.

A) Summary, Findings and Recommendations

Summary: Problem solving skills are very important and it is imperative that students should develop these skills before leaving their academic institutions and entering the real world. The presented work is an effort at improving the problem solving skills of undergrad students of CIIT studying Computer Science. Students opting OOP course from the first author were divided into control and experimental groups. The experimental group was exposed to the treatment while the control was not. The results were compared at the end of semester.

The outcome extracted from inventory data forms [1] filled by students produced mixed results. Even though the overall score, as measured by the inventory, did not show a significant difference between the two groups, but there were sub-areas such as PPO, PDF which showed improvement.

The most significant improvement occurred for PPO subscale, which meant that the treatment had a positive impact on their emotional response when they faced problems. Similarly improvement in PDF subscale meant that students who received treatment improved their abilities to understand the problems and identify the obstacles in their goal state.

The two subscales that showed least change were ICS and AVS. Out of these two, ICS had a very interesting relationship with CGPA. There was a clear relationship between score on ICS subscale and CGPA of students. Students with high score on ICS subscale had lower CGPA. The same relationship was also observed in [2].

B) Findings:

- The overall t-test result of TPSS inventory was 2.036 which is less than 2.04. Strictly speaking it does not qualify as significantly different value, but if we look at individual scales and subscales in Table 2 a significant difference can be seen.
- PPO showed the most significant improvement with a t-test value of 2.669.
- The biggest drags on the overall score of TPSS inventory were ICS and AVS. These two are more of styles than skills that one can learn by using some technique like DECSAR. As a result they show no improvement with or without treatment. But it does not mean that they are not important, especially ICS.
- As demonstrated by [9] and reaffirmed by the present research ICS has a very high level of predictability regarding academic performance of students. Being unable to make a modification in ICS score does not

mean that it is not important. On the contrary it is essential that it be investigated more thoroughly and appropriate remedies be investigated and tried.

C) Recommendations: Based upon the findings of the research study, following recommendations are proposed:

- DECSAR or any other formal technique of problem solving should be taught to students during the first year of undergraduate program.
- Students should be exposed to Concept Mapping and teachers should use it their courses for both teaching and evaluation purposes.
- Undergraduate courses should be structured in such a way that they compel the students to develop problem solving skills.
- Considering the low t-test values of ICS and AVS and the relationship of ICS with academic achievement it is imperative that it should be investigated more thoroughly and remedial measures be taken.

REFERENCES

1. D'Zurilla, T., A. Nezu and A. Maydeu-Olivares, 2002. Social problem-solving inventory-revised: Technical manual. Toronto, ON, Canada: MHS.
2. Rodriguez-Fornells, A. and A. Maydeu-Olivares, 2000. Impulsive/careless problem solving style as predictor of subsequent academc achievement. Personality and Individual Differences, pp: 639-645.
3. Ross, C. and R.R. Orr, 2007. Teaching Structured troubleshooting: Integrating a Standard Methodology into an Information Technology Program.
4. Yunus, A.S., R. Hamzah and R.A. Tarmizi, 2006. Problem solving abilities of Malayasian university students. International Journal of Teaching and Learning in Higher Education, pp: 86-96.
5. Robins, A., J. Rountree and N. Rountree, 2003. Learning and Teaching Programming: A Review and Discussion. Computer Science Education, pp: 137-172.
6. Muslim, N., J. Alias and A.Z. Mansor, 2012. "Viewpoint of Students of National University of Malaysia on Generic Skill Courses," World Applied Sciences Journal, pp: 754-761.
7. Alper, A., 2008. "Attitudes Toward Problem Based Learning in a New Turkish Medicine Curriculum," World Applied Sciences Journal, pp: 830-836.
8. Celik, S., 2008. "Fundamental Skill Levels of the Students in Vocational and Technical Education (Higher Education) Institutions," World Applied Sciences Journal, pp. 336-342.
9. Michalewicz, Z. and D.B. Fogel, 2000. How to Solve It: Modern Heuristics. Heidelberg: Springer.
10. Martinez, M.E., (1998, April 1). What is problem solving? Phi Delta Kappan.
11. Candy, P.C. and R.G. Crebert, 1991. Ivory Tower to Concrete Jungle: The Difficult Transition from the Academy to the Workplace as Learning Environments. The Journal of Higher Education, pp: 570-592.
12. O'Leary, J., 1995. What we can infer from Australian graduate employment statistics: The discrepancy between employers' expectations and graduate performance. Queensland Journal of Educational Research, pp: 15-27.
13. Kramer, J., 2007. Is abstraction the key to computing? Communications of ACM. pp: 37-42.
14. Garner, S., P. Haden and A. Robins, 2005. My Program is Correct But it Doesn’t Run: A Preliminary Investigation of Novice Programmers’ Problems. Australasian Computing Education Conference (pp: 173-180). Newcastle, Australia.: Australian Computer Society, Inc.
15. Robins, A., P. Haden and S. Garner, 2006. Problem Distributions in a CS1 Course. Eighth Australasian Computing Education Conference. Hobart, Tasmania, Australia: Australian Computer Society, Inc.
16. Novak, J.D., 1998. Learning, Creating and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations. Mahwah, New Jersey: Lawrence Erlbaum Associates.
17. Novak, J.D. and A.J. Cañas, 2005. Building on New Constructivist Ideas and CmapTools to Create a New Model for Education. Institute for Human and Machine Cognition.
18. John, W.B. and V.K. James, 2008. Research in Education, Pearson Prentice Hall, pp: 407.