A Self Review and External Review Model for Teaching and Assessing Novice Programmers

Sohail Iqbal and Om Kumar Harsh

Abstract—**Teaching and assessing programming skills in novice programmers is a challenging and difficult task. Most of the introductory programming courses emphasized in teaching programming knowledge (syntax and semantics of programming language to the novice programmers) instead of giving equal attention and time to problem solving strategies, analysis of output(results) and improvement in programming practices . Hence this paper makes an attempt to introduce ADRI model (well known quality assurance model for self review and external review) approach in preparing teaching and assessment materials in programming courses to novice programmers so that it ensures that programming knowledge and problem solving strategies are properly incorporated and practiced in teaching and assessing materials for novice programmers.**

Index Terms—**Novice programmers, ADRI model, programming knowledge, problem solving strategies.**

I. Introduction

Coping with high failing rate or dropout from the introductory programming course is a big challenge in computer science discipline [1], [2]. Different approaches have been proposed to improve the programming skills of novice programmers [3]-[6]. It includes creating visualizations building block from multimedia tools [7], goal/plan analysis programming strategies [8], teaching machine [9] and natural language tutor –ProPL [10]. Winslow [11] pointed out "*that novice programmers know the syntax and semantics of individual statements, but they do not know how to combine these features into valid programs*". Present work addresses the key issue that how can we prepare our teaching and assessment materials which ensures that novice programmers are practicing programming knowledge (syntax and semantics) [8] and problem solving strategies by paying identical attention and time.

For realizing above mentioned tasks in greater details, we propose ADRI (Approach, Deployment, Result, and Improvement) model which is well known within quality assurance and enhancement discipline for self review and external review [12]. It is worth to mention here that ADRI model is a general tool for assessment and improvement [13] in many ways. The Present authors have already been worked on ADRI model in explaining the Knowledge Management [14]-[15].

Jantti [16] explained four approaches of ADRI (Fig. 1) as follows:

1) "Approach - Thinking and planning"
2) "Deployment - Implementing and doing"
3) "Results - Monitoring and evaluating"
4) "Improvement - Learning and adapting"

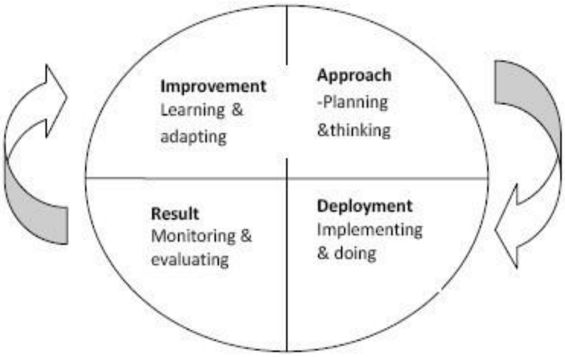

Fig. 1. ADRI model.

The first stage is Approach which generally consists of thinking and planning about tasks. It normally furnishes development of goals, strategies, objectives, outcomes, plan and targets. Planning should clearly indentify qualitative and quantitative goals and should indicate steps to achieve these goals [13].

The second stage is Deployment which provides a platform to execute or implement tasks. It is important that clear understanding including steps to achieve goals should be in place. Therefore it is evident that proper planning should be ensured at the first stage to achieve targets or goals [13].

The third stage is the Result, which refers to the output or findings as consequences of the first and second stages. In other words, Result stage refers to what is actually achieved. Here the most important point is that there should be link between output (Result stage) and goal (approach stage). Result stage provides comparison between intended output and achieved output. Therefore, the result stage also gives opportunity to analyze output and draw conclusions.

The fourth stage is Improvement which refers to the conclusions drawn from the results and analysis phases. This stage identifies what needs to be improved [17]? As mentioned above, ADRI is a continuous cycle of improvements; therefore how to improve is left to the next implementation of the ADRI cycle.

II. Different Aspects of Novice Programmers and Programming Languages

Most of the text books used for teaching programming

Manuscript received November 4, 2012; revised January 8, 2013.

S. Iqbal is with the Department of IT of Buraimi University College, Muscat, Oman (e-mail: Sohail@buc.edu.om).

O. K. Harsh is with Hindustan College of Science & Technology/ Director, Mathura, India (e-mail: Director.hcst@gmail.com).

DOI: 10.7763/IJIET.2013.V3.247

languages mainly emphasize the syntax of programming languages rather than paying equal attention to problem solving skills. In some cases, problem solving topics used to be generally discussed only during first Chapter. Normally, there are only small numbers of text books where problem solving topics are discussed and integrated throughout the book in different topics [8].

Novice programmers offer less priority for "planning and testing code and try to solve programs" [18] in the current context rather than understanding the broader sense of programs.

Ala-Mutka [19] mentioned that learning programming languages include different activities like "learning the programming language features, program design and comprehension". Therefore novice programmers should not only concentrate on programming knowledge but to pay equal attention to problem solving and analytical skills.

Novice programmers generally understand programming language syntax but find it difficult to combine these features into valid program [20].

Novice programmers generally start writing their programs before making an attempt to plan them [10].

Soloway [21] mentioned that it takes 10 years for novices to become expert programmers. Therefore it is necessary for novices to practice programming at a broader sense instead of just getting surface knowledge.

Webster [20] discussed that in most of the cases programmers try to solve the "problem definition directly into coding the problem" as shown in Fig. 2.

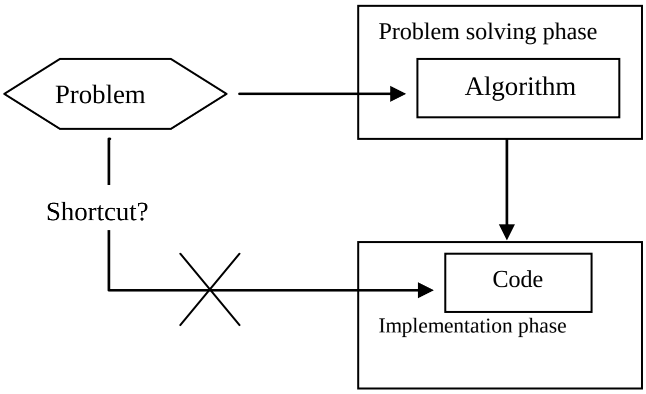

Fig. 2. Programming shortcut? Webster [20] –Page 7.

Programmers normally prefer shortcut (problem→ code) to save time but Webster [20] concluded that "shortcut actually takes more time and effort". Developing an algorithm for a problem definition avoids mistakes and "keeps your thought straight". If programmers don't spare time at the beginning to develop an algorithm then later on it needs more time to debug and revise program [20]. Therefore it is very important for novice programmers to follow proper steps to address problem definition. Hence this paper makes an attempt to introduce ADRI model approach in preparing teaching and assessment materials in programming courses to novice programmers so that it ensures that programming knowledge and problem solving strategies are properly incorporated and practiced in teaching and assessing materials for novice programmers.

III. ADRI Model to Handle Programming Knowledge and Problem Solving Strategies

De Raadt [8] mentioned that one of the reasons for poor performance of novice programmers (in standardized program generation tests) is the method of preparing and teaching traditional introductory courses which fail to adequately educate the majority of students about the programming acquaintance and problem solving approaches.

In the present work we suggest the following tasks and sub tasks in four approaches as a part of ADRI model in the context of novice programmers.

1) Approach: Problem solving strategies
2) Deployment: Programming knowledge (Syntax and semantics)
3) Result: Analysis and comparison of intended and achieved output. Make sure that intended goals are achieved or not.
4) Improvement: Learning and recommendations for instructor and novice programmer

TABLE I: Proposed ADRI Model Four Approaches for Novice Programmers

ADRI model four approaches	Task and sub tasks
Stage 1: Approach	*Problem solving strategies:* -Understanding and specification (Analyze the problem and "what the solution must do".) -General solution and algorithm (List the steps to solve problem and specify data types) -verify the algorithm (check whether required solution is achieved by following steps) [20]
Stage 2: Deployment	*Programming knowledge (Implementation)* -"Concrete solution" (using particular programming language syntax and semantic to develop program from the algorithm) -Testing and debugging syntax errors (Compile the program, if find errors, locate the errors in source code and make corrections) [20]
Stage 3: Results	*Analysis* - Compare intended results with achieved results - Make sure that intended goals achieved or not - Discuss achieved results to incorporate any changing requirements - Figure out any program running errors and correct it
Stage 4: Improvement	*Learning and recommendations* For novice programmer: -Add more features or functionality in program -Try different programming constructs to solve same problem (loops, logical operators) For Instructor: -Update teaching materials or add more lab questions/sheets -Discuss topic again in class -Change teaching methodology

IV. Benefits and Execution Steps of ADRI Model

ADRI model [12] provides four steps (Approach – Deployment – Result – Improvement) which gives clear understanding to novice programmers that what is expected from them to develop their programming skills and knowledge. The ADRI model not only emphasizes problem solving strategies (Approach) and programming knowledge (Deployment), it also gives opportunity to analyze (Result) the Results, learn (Improvement) and adapt improvements.

Instructors should develop teaching materials particularly lab sheets according to the ADRI model's [12] four approaches. It helps novice programmers to practice all necessary skills of programming. As discussed above, most of the programming books emphasize only on programming knowledge (syntax) [8] but ADRI model four approaches not only cover syntax but also offer equal attention to problem solving, analysis and improvement of novice programmers' skills and knowledge.

As shown in Table I, stage 1: Approach is mainly concerned with problem solving strategies which helps novice programmers to understand the problem and develop algorithm or general solution.

Stage 2: deployment provides platform for novice programmers to translate algorithm developed in stage 1 into computer program by using any programming language. This stage helps novice programmers to become familiar with programming constructs and develops their skills to test and debug syntax errors.

ADRI model stage 3 (Result) helps novice programmers to analyze and compare intended output with achieved output. It also provides an opportunity to compare goals (Approach) with achieved output.

ADRI model stage 4 (Improvement) has two perspectives, one for instructor and other for novice programmer. In Improvement section of lab sheets, instructor can give directions to add more features or to achieve the same tasks by using different construct of programming language. It provides more exposure of programming constructs to novice programmers which ultimately helps to leverage their programming skills. Instructors will get feedback from novice programmers (or after analyzing their achieved output); it gives opportunity to them to improve their teaching and assessment materials.

While preparing assessment tasks for novice programmers, instructor should consider and incorporate all the four approaches of ADRI model in questions. In fact, instructor should inform novice programmers at the beginning of the semester or through course descriptions that they will be assessed against all the four key areas of ADRI model approaches. All the assignments, quizzes or other assessment tools should clearly specify the ADRI model's four approaches along with weight-age so that the novice programmers will understand the importance of programming knowledge (syntax and semantics) [8] and problem solving strategies. In most of the cases assessment tools include tasks related to problem definition and expected output but this paper emphasize to include tasks related to improvements which helps novice programmers to develop problem solving and analytical skills again and again by using different constructs of programming language.

V. Conclusions

Since available literature demonstrates that most of the introductory programming courses emphasize on programming knowledge (syntax and semantics) and use less effort on problem solving strategies as a result, novice programmers perform poorly in standardized program generation tests [8]. However, in the present discussion we can realize that ADRI model's four approaches provide satisfactory solution to above mentioned situation during teaching and assessment materials preparation which not only covers programming knowledge and problem solving strategies while it also enhances novice programmers' analytical and critical thinking skills. The ADRI model's four approaches suggest novice programmers to solve problem definition by going through four stages instead of using "programming shortcut" which apparently saves time but needs more efforts to debug and revise programs [20]. The ADRI model introduces Result (stage 3) and Improvement (stage 4) as separate tasks for novice programmers which helps them to pay equal attention to analyze the achieved output as well as provides opportunity to solve the same program with different programming constructs and building up solution by adding more functionalities or features. More work in this line is in progress and will be communicated shortly elsewhere.

References

[1] E. Lahtinen, K. Ala-Mutka, and H. M. J ärvinen, "A study of the difficulties of novice programmers," *ACM SIGCSE Bulletin*, vol. 37, no. 3, pp. 14-18, 2005.

[2] E. R. Sykes, "Determining the Effectiveness of the 3D Alice Programming Environment at the Computer Science I Level," *Journal of Educational Computing Research*, vol. 36, no. 2, pp. 223-244, 2007.

[3] J. Kay, M. Barg, A. Fekete, T. Greening, O. Hollands, J. H. Kingston, and K. Crawford, "Problem-based learning for foundation computer science courses," *Computer Science Education*, vol. 10, no. 2, pp. 109-128, 2000.

[4] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, and J. Paterson, "A survey of literature on the teaching of introductory programming," in *Proc. Working group reports on ITiCSE on Innovation and technology in computer science education, ACM*, 2007.

[5] A.Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and Discussion," *Computer Science Education*, vol. 13, no. 2, pp. 137-172, 2003.

[6] G. R ößling, M. Joy, A. Moreno, A. Radenski, L. Malmi, A. Kerren, T. Naps, R. J. Ross, M. Clancy, A. Korhonen, R. Oechsle, and J. Á. Iturbide, "Enhancing learning management systems to better support computer science education," *SIGCSE Bulletin*, vol. 40, no. 4, pp. 142-166, 2008.

[7] T. Linden and R. Lederman, "Creating visualizations from multimedia blocks: A simple approach to teaching programming concepts," *ISECON Proceedings*, vol. 28, no. 1619, 2011.

[8] M. De Raadt, "Teaching programming strategies explicitly to novice programmers," PhD Thesis, School of Information Systems, University of Southern Queensland, 2008.

[9] M. Bruce-Lockhart and T. Norvell, "Lifting the hood of the computer: program animation with the Teaching Machine," in *Proc. the Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 831-835, 2000.

[10] H. C. Lane and K. VanLehn. (Feb, 2012). Teaching the tacit knowledge of programming to novices with natural language tutoring. [Online]. Available: http://www.citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124

[11] L. E. Winslow, "Programming pedagogy—a psychological overview," *ACM SIGCSE Bulletin*, vol. 28, no. 3, pp.17-22, 1996.

[12] M. Carroll. (Feb 2nd, 2010). ADRI. *A Quality Assurance model for self reviews and external reviews*. [Online]. Available: http://www.oac.gov.om/files/qe/training/slides/01v6.pps

[13] B. A. Abuid, "ADRI – Self assessment model for teaching and learning," presented at 2nd international conference on global trends

and challenges in higher education and quality assurance, 12-13 June, 2010, Oman.

[14] A. Alani, O. K. Harsh, and S. Iqbal, "Role of Information and Communication Technology on Knowledge Management in a higher educational environment," presented at International Conference on Information and Communication Technology, March 22-23, 2009, Middle East College of Technology, Muscat, Oman.

[15] A. Alani, O. K. Harsh, and S. Iqbal, "Qualitative Knowledge Management and Knowledge Reuse in Higher Educational Setting," presented at the second international conference on quality and higher education,Muscat, Oman, June 12 to June 13, 2010.

[16] M. H. Jantti, "Minding your own business: can a business excellence framework translate to the education sector?" in *Proc. Quality conversations on the Annual Higher Education Research and Development Society of Australasia Conference*, presented at 25th Annual International HERDSA Conference, Perth, 7-10 July, 2002.

[17] Quality Frameworks. (2006). *Reflections from Australian Universities*. edited by Jeanette Baird. Australian Universities Quality Agency. [Online]. Available: http://www.auqa.edu.au. Accessed on 10th Sep 2009.

[18] M. Kӧlling and J. Rosenberg, "BlueJ - A Language for Teaching Object-Oriented Programming," in *Proc. the 27th SIGCSE Technical Symposium on Computer Science Education*, pp. 190-194, 1996.

[19] K. Ala-Mutka, Problems in learning and teaching programming - a literature study for developing visualizations in the codewitz-minerva project – Codewitz needs analysis.

[20] M. Webster, "Overview of programming and problem solving," *Collegiate Dictionary*, Tenth Edition, 1994.

[21] E. Soloway and J. Spohrer, "Studying the Novice Programmer," *Lawrence Erlbaum Associates*, Hillsdale, New Jersey, pp. 497, 1989.

Sohail Iqbal Malik was born in Chakwal, Pakistan on April 05, 1976. He did his Masters' in Computer Science from Victoria University, Melbourne, Australia in 2003. He is head of IT Department at Buraimi University College, Oman. His research interest includes novice programmers, knowledge management and web learning.

Om K. Harsh was born in Kolkata, India on Sept 12, 1952. He possesses four research Degrees, namely Ph.D. (Physics, Agra University, Agra, India, 1979), D.Sc. (Physics, Kanpur University, Kanpur, India, 1990), Ph.D. (Information Systems, from University of New England, Armidale, Australia, 2011) and Masters' of Electronic Engineering (by research from University of New South Wales, Sydney, Australia, 1997).

Dr. Harsh has served at institute of technology, BHU, India as a post doctoral research fellow, Lucknow University, Lucknow, India as a post doctoral research fellow, Kanpur University, as a faculty member in physics. In addition, he also served at University of New South Wales in Sydney, as a research scholar, Monash University, Malaysia, University of South Australia, Australia, Champlain College, Dubai, Buraimi University College affiliated to California State University in Oman as a faculty member of computer science/information technology. Presently he has been working as a director of prestigious Engineering College of Western India which is situated in mathura, India. He has guided five Ph. D. students, 4 M. Tech students and several Masters' and Honours students in various areas and published around 50 research papers in the journals of International repute in the fields of Semiconductor Plasma Physics, X-Rays Plasma, Chemical Shift, Software Engineering, Distance Education, Information Systems and Knowledge Management.

Professor Harsh is a Member of Rio ́s International Scientific Council, Association of Engineering and International Forum of Educational Technology & Society. He was awarded most prestigious "Rashtriya Gaurav Award" by India's International Friendship Society for his excellence in academia.