Scaffolding Techniques provided to learners in Tools for Learning and Teaching

Ali Raza
Department of Computer Science
University of Management and Technology
Lahore, Pakistan
aliirazii@gmail.com

Dr.Muhammad Shoaib Farooq
Department of Computer Science
University of Management and Technology
Lahore, Pakistan
Shoaib.farooq@umt.edu.pk

Dr.Adnan Abid
Department of Computer Science
University of Management and Technology
Lahore, Pakistan
Adnan.abid@umt.edu.pk

Ashfaq Ahmad
Department of Computer Science
University of Management and Technology
Lahore, Pakistan
Ashfaq.ahmed024@gmail.com

Abstract*—*Learning and teaching computer programming requires considerable effort. For promoting quality teaching and learning, such tools are being used which help in both the aspects and increase the learnability of learners. Scaffolding (Support) is provided in these tools to facilitate learning and teaching. With this required support apart from traditional methodology of teaching and learning, learners ability of grasping concepts is enhanced significantly. In this paper, scaffolding techniques are discussed which are provided in tools and it aids in learning. Lastly, guideline are given based on which scaffolding tools should be implemented for learners.

Keywords—Scaffolding; Tools; Learners; Teaching; Support

I. INTRODUCTION

Computer programming is a tough subject for the novice programmers [1] [2]. Scaffolding (Support) is needed by learner while they learn to program. Self-learning in the field of programming is sometimes impossible for novices. The scaffolding techniques help learners of programming and other subjects. The implication of the scaffolding varies in the form of guidance required from the students. The scaffolding techniques aim to make the novice learners independent problem solvers where learning by doing is implemented under guidance [3].Scaffolding also refers to the discovery based learning which provides guidance to students while attempting programming tasks. Guided tasks with scaffolding placed in them are given to novice learners to help them [4].The term scaffolding refers back to the work done by the L. Vygotsky in the 1930s [5].

Every teacher has different teaching style and methodology adopted for teaching, similarly, teachers provide scaffolding techniques to learners in different styles. These scaffolding techniques provided to learner have their own pros and cons [6].

In this paper, we are going to discuss the scaffolding techniques which are provided to learners with the help of tools of the learning. A framework is used based on which scaffolding techniques are discussed is defined below in Figure 1.

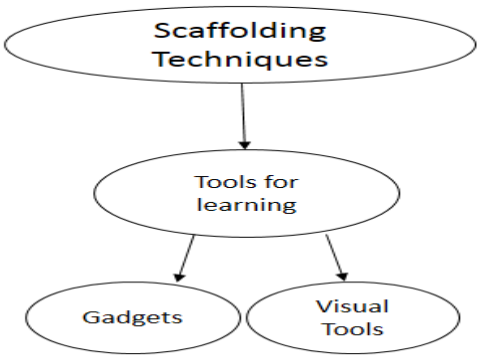

Figure 1: Framework used for explaining scaffolding techniques

II. TOOLS PROVIDING SCAFFOLDING

In this section, tools are discussed in which scaffolding techniques are provided to learners.

Quintana et al. [7] elaborated the use of a tool for scaffolding environment 'Symphony' which is a mix of different individual tools helping high school students in science inquiry using a learners centered approach, the study also evaluated the use of 'symphony' and final set of guidelines for scaffolding are also proposed. Hansen et al. [8] presents 'Ideogramic' which is a tool for gesture based modelling of object oriented concept in UML, it help teachers and learners in removing the deficiencies with the extent(gesture based tool help teachers in limiting functionality to student), complexity and usability, results have shown that 'Ideogramic' supports active learning scaffolding. . Louca [9] compared between Microworlds Logo and Stagecast Creator programming environment for fifth grade modeling.

Table 1: Comparison between Microworlds Logo and Stagecast Programming Environment

Tools	Focus	Usability	Mode
Microworlds Logo	Program structure	Tangible	Technical
Stagecast	Simulation	Easy	Story based

Rosson et al. [10] describe the workshop on project wConnect (scaffolded environment for the female high school learners for the dynamic website building using the Bridgetools components) in which scaffolding is provided in the form of tool and materials e.g. instantiated a database. Lasserre et al. [11] drafted a proposal that games can be used to learn to program, it proposes a database type gaming system. A gaming strategy is defined which help programmers to practice programming on the daily basis. Hmelo [12] advocated learning by doing approach to learning by providing two types of scaffolding, Black Box (allow learners to complete the task, but don't provide details on how it's been done) and Glass Box (provide inside details of scaffolding provided, what and why support is needed) in two different software's for learning CLINSIM (Clinical Simulation) and EMILE (Physics Simulation).Guzdial et al.[13]defined the integration of Computer aided design tool(CAD) with the scaffolding to make it a benchmark called GPCeditor (Goal-Plan-Code editor), which proved to help novice programmers in programming, GPCeditor covered all the learning process of programming and scaffolding is provided in the form of encouraging articulation and coaching. Quintana et al. [14] designed Symphony in Java, a scaffolded integrated tool environment (SITE) for the science project of high school students built on the learner centered design (LCD) and the needs of learners are known with the help of process-space model and these shortcomings identified are then provided with scaffolding in the Symphony which help in the complex work processes. Chang et al. [15] compared two modules of computer based concept mapping and construct with pencil and paper in biology. The modules of computer based mapping were construct-by-self and construct-by-scaffold and both had evaluation and hint function in them, results proved that construct-by-scaffold was more effective way of concept mapping in biology than the rest of the techniques. Garner [16] discuss the resources and tools to help novice programmers in problem solving under the four phases of the software development life cycle.

Table 2: Learning Environments falling under Software Development Life Cycle

Phases	Analyze the Problem	Design and Develop a Solution	Implement the Algorithm	Test and Revise the Algorithm
Tools	SOLVEIT	Karel the Robot	FLINT	Visual BASIC,
	Video Clips	FLINT	Visual BASIC editor	BRADMAN
	Microworlds	The Sort Algorithm Animator	BlueJ	Karel the Robot and Animation

Carr et al. [17] address a tool that supports explanation system for helping teacher's present code to students step by step and providing scaffolding for students by providing detail of the design process of each part of the program and it also provides aid for cognitive apprenticeship. Reiser [18] unfold how mechanism of structuring the task (decomposing complex tasks, focusing efforts, monitoring) and problematizing the aspects of subject matters (Elicit articulation, surface gaps) in the tools help students in their learning process and further how these two mechanisms complement each other. Beale [19] described the scaffolded intelligent environment for the reading of special children in which scaffolding principles are designed with the help of behavioural concepts, young learners are also integrated assessed in the environment which also capture the response of the learner with the environment and then further these scaffolding principles are incorporated in the intelligent learning environment. Cagiltay [20] untangle the scaffolding strategies implemented in the electronic performance support system (EPSS) that can provide support to students and the challenges faced during the implementation of the scaffolding strategies in the EPSS which are fading of scaffolding, database and expert system aspects, interface design and human related factors, users learning style and scaffold coupling (Sharing performance between scaffolds can direct user to perform efficiently).

Table 3: Scaffolding techniques in the EPSS

Scaffolding Techniques	Conceptual	Meta-Cognitive	Procedural	Strategic-Intrinsic
Explanation	What to consider and association between ideas	What I have done and what to do next?	Utilization of tools and resources given in the environment	Guidance in analyzing and approaching learning tasks

Razzaq [21] conducted an experiment, between scaffold and hints in a web based assessment system for class 8 math students, that resulted in the favor of scaffolds and survey showed students in the scaffolds were able to learn more rather than hints. Quintana & Fishman [22] designed the scaffolding framework for describing scaffold on students and teachers in science software in inquiry based science activities and review the framework for where students need scaffolding.

Table 4: Tools used in framework that provide scaffolding

Framework	Sense Making	Process Management	Reflection	Articulation
Tools	Model-It, eChem, KNOW	KNOW, Symphony, Digital IdeaKeeper, PERT	PERT	Symphony process wheel with plan grid

Lee et al. [23] explain the effectiveness of screencasts (Screenshots merged to make a video) and Cognitive tools (BlueJ) for the object oriented programming of the novices. The authors stress that BlueJ as a cognitive tool should be used so learners can represent their schemas. The study found no significant effect of screencast on learning of the learners and BlueJ on testing. But authors did not dismiss the value of screencasts and BlueJ as a scaffolding provider for learning and cognitive tool. Kölling [24] designed Greenfoot, an educational integrated development environment for programming in JAVA that helps learners in learning JAVA rather than other ALICE and SCRATCH.

Table 5: Strength and Weakness of Greenfoot Environment

Strength	Better illustration of Object Oriented concepts	Good Scaling up	Simple start in Greenfoot
Weakness	Need of learning syntax	Error handling	Designed in 2D system

Vihavainen et al. [25] briefed how Test My Code (TMC) automated assessment and data gathering tool provides students scaffolding in programming exercises, bi-directional feedback is supported between students and instructor. TMC allows better use of resources and time management for students during the course CS1, CS2, Web Programming and others.

WARFORD [26] explained BlackBox framework, which is an environment for an object oriented application development and shared its use in the CS1/CS2 course. Human/computer interaction, Formal methods and object oriented programming are introduced during the freshman year and these courses are taught with the help of BlackBox framework. Support is provided to novice learners in the form of simple design, a new language designed component Pascal, Fast compilation and lowest cost. The framework has proved to be ideal in helping beginners in CS1/CS2.

Dr.Scheme is a graphical programming environment for Scheme, which works as a graphical editor which is used in the Rice University for teaching introductory and advanced level courses. It is specially designed for students as it has a tower of syntactical variants of the scheme that help to catch student's mistakes and give their explanation. It comes with an algebraic expression, syntax checker and static debugger. The future versions of the Dr. Scheme provide support in a stepper for managing advance and full level scheme. That will include call and multi-threaded programs, high order procedures [27].

Sanders & Dorn introduced Jeroo, an effective integrated environment and simulator for teaching concepts of objects, methods and control structures to naïve programming students. Jeroo syntax provided an easy syntax so students can easily do transition to Java or C++. The Development environment consists of the single window in which all the components were visible. The tool proved to be effective for helping students of the university of Northwest Missouri State in learning difficult topics [28].

Moreno et al. presented a visualization tool named Jeliot 3 that helped novice learner in learning procedural and object oriented programming. When students develop their program, then a visualization of the execution of the program is made. The main new features of Jeliot 3 were better error information, a larger number of programs accepted, attractive design and objects support. The future work consists of developing new visualization models in Jeliot 3, which can adjust to different situations [29].

III. Guidelines for Creating a Scaffolded Tool

In this section, guidelines are provided for building a scaffolded tool that can help novice learners not only in programming relevant subject but aids in other subjects also.

i) All interfaces should be neat and clean

ii) Extra detail should be excluded from the main interface especially

iii) Early support should be provided to novice learners and as they make the transition from novice to expert that support should be faded away.

iv) Incorporation of visualizations in the tools

v) Simulations should be provided to learners with respect to different topics

vi) Execution of the tool should be fast

vii) Tool should incorporate automated features for novice learners

viii) Better explanation of errors specifically for programming related tools

ix) Embedding expert system aspects in the tool

x) Tools should decompose complex tasks, provide instruction, hints and guidance for the novice learners

xi) Tools should have pre-defined examples relevant to tasks from different levels.

xii) Tools should be build using Learner- centered methodology by understanding needs of learners.

xiii) Tools should have option for learner to work with Black Box and Glass Box methodology so learners can feel the difference working with and without details

xiv) Tools can incorporate gaming methodologies and can make task interesting for learners.

xv) Use of Computer aided devices should be incorporated in the tool

xvi) Finally, tools should be built on scaffolding frameworks found in the literature.

IV. Conclusion

Scaffolding techniques provide aid to teachers and learners in facilitating their learning. Tools play an important role in the teaching and learning of not only programming but other subjects also. If such scaffolding techniques should be used while building programming or any other tool then learning can be enhanced for both teacher and learners.

Acknowledgment

I am very thankful to the department of computer science at University of Management and Technology for providing me peaceful and ambient environment. I am also very thankful to all the reviewers who took time out of their busy schedule for reviewing this article.

References

[1] Reardon, S., & Tangney, B. (2014). Smartphones, Studio-Based Learning, and Scaffolding: Helping Novices Learn to Program. ACM Transactions on Computing Education (TOCE), 14(4), 23.

[2] Dehnadi, s., bornat, r., & adams, r. (2009, june). Meta-analysis of the effect of consistency on success in early learning of programming. Ppig

[3] Quintana, C., Krajcik, J., & Soloway, E. (2002). Scaffolding Design Guidelines for Learner-Centered Software Environments.

[4] Alfieri, L., Brooks, P. J., Aldrich, N. J., & Tenenbaum, H. R. (2011). Does discovery-based instruction enhance learning?. Journal of Educational Psychology, 103(1), 1.

[5] L Vygotsky. Mind in Society: Development of Higher Psychological Processes (14th edition).Harvard University Press, Cambridge, MA, 1978

[6] DUNCOMBE, R. (2010). Mobiles for development research: Quality and impact.M4D 2010, 12.

[7] Quintana, C., Krajcik, J., & Soloway, E. (2002, April). A case study to distill structural scaffolding guidelines for scaffolded software environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 81-88). ACM.

[8] Hansen, K. M., & Ratzer, A. V. (2002, June). Tool support for collaborative teaching and learning of object-oriented modeling. In *ACM SIGCSE Bulletin* (Vol. 34, No. 3, pp. 146-150). ACM.

[9] Louca, L. (2004, June). Programming environments for young learners: a comparison of their characteristics and students' use. In *Proceedings of the 2004 conference on Interaction design and children: building a community* (pp. 129-130). ACM.

[10] Rosson, M. B., Ioujanina, A., Paone, T., Sheasley, G., Sinha, H., Ganoe, C., ... & Mahar, J. (2009, March). A scaffolded introduction to dynamic website development for female high school students. In *ACM SIGCSE Bulletin* (Vol. 41, No. 1, pp. 226-230). ACM.

[11] Lasserre, P., & Kotowick, K. (2010, May). Proposal for a new strategy to practice programming. In *Proceedings of the 15th Western Canadian Conference on Computing Education* (p. 9). ACM.

[12] Hmelo, C. E., & Guzdial, M. (1996, July). Of black and glass boxes: Scaffolding for doing and learning. In *Proceedings of the 1996 international conference on Learning sciences* (pp. 128-134). International Society of the Learning Sciences.

[13] Guzdial, M., Hohmann, L., Konneman, M., Walton, C., & Soloway, E. (1998). Supporting Programming and Learning-to-Program with an Integrated CAD and Scaffolding Workbench*. *Interactive learning environments*, *6*(1-2), 143-179.

[14] Quintana, C., Eng, J., Carra, A., Wu, H. K., & Soloway, E. (1999, May). Symphony: A case study in extending learner-centered design through process space analysis. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 473-480). ACM.

[15] Chang, K. E., Sung, Y. T., & Chen, S. F. (2001). Learning through computer-based concept mapping with scaffolding aid. *Journal of Computer Assisted Learning*, *17*(1), 21-33.

[16] Garner, S. (2003). Learning resources and tools to aid novices learn programming. In *Informing Science & Information Technology Education Joint Conference (INSITE)* (pp. 213-222).

[17] Carr, L. A.,Davis, H. C., & White, S. A.(2004).Annann-a tool to scaffold learning about programs.

[18] Reiser, B. J. (2004). Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *The Journal of the Learning Sciences*, *13*(3), 273-304.

[19] Beale, I. L. (2005). Scaffolding and integrated assessment in computer assisted learning (CAL) for children with learning disabilities. *Australasian Journal of Educational Technology*, *21*(2).

[20] Cagiltay, K. (2006). Scaffolding strategies in electronic performance support systems: Types and challenges. *Innovations in education and Teaching International*, *43*(1), 93-103.

[21] Razzaq, L., & Heffernan, N. T. (2006, January). Scaffolding vs. hints in the Assistment System. In *Intelligent Tutoring Systems* (pp. 635-644). Springer Berlin Heidelberg.

[22] Quintana, C., & Fishman, B. (2006). Supporting science learning and teaching with software-based scaffolding. *American Educational Research Association (AERA)*.

[23] Lee, M., Pradhan, S., & Dalgarno, B. (2008). The effectiveness of screencasts and cognitive tools as scaffolding for novice object-oriented programmers. *Journal of Information Technology Education: Research*, *7*(1), 61-80.

[24] Kölling, M. (2010). The greenfoot programming environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 14.

[25] Vihavainen, A., Vikberg, T., Luukkainen, M., & Pärtel, M. (2013, July). Scaffolding students' learning using test my code. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education* (pp. 117-122). ACM.

[26] Warford, J. S. (1999, March). BlackBox: a new object-oriented framework for CS1/CS2. In *ACM SIGCSE Bulletin* (Vol. 31, No. 1, pp. 271-275). ACM.

[27] Findler, R. B., Clements, J., Flanagan, C., Flatt, M., Krishnamurthi, S., Steckler, P., & Felleisen, M. (2002). DrScheme: A programming environment for Scheme. *Journal of functional programming*, *12*(02), 159-182.

[28] Sanders, D., & Dorn, B. (2003, February). Jeroo: a tool for introducing object-oriented programming. In *ACM SIGCSE Bulletin* (Vol. 35, No. 1, pp. 201-204). ACM.

[29] Moreno, A., Myller, N., Sutinen, E., & Ben-Ari, M. (2004, May). Visualizing programs with Jeliot 3. In *Proceedings of the working conference on Advanced visual interfaces* (pp. 373-376). ACM.