SUPPORTING PROGRAMMING TEACHING BY ADAPTING LEARNING MATERIAL TO LEARNING STYLE

Mohammad T. Alshammari
College of Computer Science and Engineering, University of Hail, Saudi Arabia
md.alshammari@uoh.edu.sa

ABSTRACT— *Teaching programming is a challenging task because of the need to successfully incorporate both the theory and the application of programming concepts. In addition, learners differ in their characteristics such as knowledge level and learning style. As a result, there is a necessity in supporting different characteristics of learners. This paper proposes a specific approach based on learning style to enhance the learning of programming, and a virtual learning tool is designed based on the approach. The approach takes into account a widely used model of learning style called Kolb's Experiential Learning Cycle. An initial investigation of perceived usability of the tool is also provided. The tool is highly usable, and the approach could be particularly useful in lab-based learning of programming. Furthermore, the approach has a potential to be applied in different contexts for other courses such as cybersecurity and artificial intelligence.*

Keywords—learning; programming; learning style; teaching

I. INTRODUCTION

Computer programming is often considered a core module in computer science programs. However, teaching computer programming is a challenging task. In a traditional setting, the complexity of teaching computer science courses in general, and computer programming courses in particular, arises from the requirement of combining theoretical concepts with applications and examples [1], [2]. Another source of difficulty stems from the requirement to meet different requirements of all learners in both classroom and lab-based learning.

Traditional approaches in teaching programming may not well support different needs of learners. They usually provide the same material and the same presentation of learning material irrespective of the characteristics of learners - such as their knowledge, abilities and learning style. Furthermore, an independent approach of studying taken by learners may lead to poor decisions on what and how to study. In addition, pedagogical aspects when teaching programming need to be carefully considered; teaching programming should follow well-defined instructional design models [1].

Several instructional approaches have been proposed in computer science in order to make the educational process more effective and to meet the needs of learners. Adaptation of learning material based on knowledge and learning style has been the subject of intensive research in online learning [3]. For example, SQL-Tutor is an intelligent e-learning system that customizes the sequence of SQL lessons based on the knowledge level of learners [4]. An approach that takes into account the learning style in order to provide instructional recommendations to learners has also been represented by the eTeacher system for teaching artificial intelligence [5]. The Protus system combines knowledge level and learning style to personalize learning material for teaching programming [6]. Similarly, the AdaptLearn system integrates learning style to teach computer security; an experiment was also conducted and produced significant results about learning gain and satisfaction of learners [7].

The deployment of these systems, among others, has produced promising results in enhancing learning and the satisfaction of learners for different computer science topics [8].

However, few attempts have been made in incorporating learning style of learners in traditional learning in both lecture-based and lab-based learning. Furthermore, adapting learning material according to learning style is still a controversial issue; it is not always evident how to provide adaptation based on learning style [9]. These issues need to be addressed in order to make computer-programming education more effective.

This paper presents a specific approach based on learning style to enhance learning of programming. The approach takes into account a widely used model of learning style [10]. The model takes into account different phases that support different abilities of learners when teaching programming. An experimental evaluation of perceived usability of a learning tool implemented based on the proposed approach is also given.

The next section provides the theoretical foundations of this research. Section III gives an outline of the proposed approach. Section IV outlines the evaluation method. Section V presents the results of the experiment. Section VI offers a critical discussion of the work and draws some conclusions.

II. LEARNING STYLE

Learning style is defined as "characteristic cognitive, affective, and psychological behaviors that serve as relatively stable indicators of how learners perceive, interact with, and respond to the learning environment" [11]. Several studies have emphasized the importance of learning style in order to improve learning [11, 12]. More importantly, it is argued that computer science education should support many different learning styles [13].

Kolb has pointed out that knowledge results from the interaction between theory and experience, and proposed a learning model called Kolb's Experiential Learning Cycle [10]. The model suggests that learning is conceived as a cyclical process of four steps as shown in Figure 1. According to the model, learners must go through and be able to effectively recall, understand and apply the learning concepts. Moreover, by applying this model, different learning styles of learners can be accommodated.

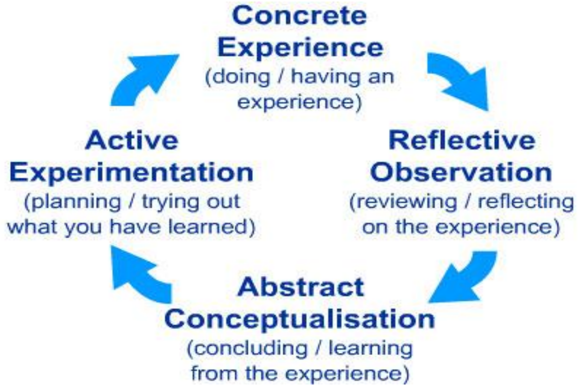

Figure 1. Kolb's Experencial Learning Cycle.

Kolb's model emphasizes four stages of learning to support different abilities of learners for effective learning. The four stages of the model are listed as follows:

- Concrete experience.
- Reflective observation.
- Abstract conceptualization.
- Active participation.

An individual should be able to carry out a specific action and then observe the results of the action in a particular situation or context (concrete experience).

The ability to review and reflect upon what has been completed and experienced without task involvement is important (reflective observation).

The learner should also be able to interpret the learning events that have been observed and to understand the relationships between the events' factors (abstract conceptualization).

The fourth ability is concerned with the application of what is known in a new context or situation (active participation).

The next section provides the proposed approach based on this model and offers a specific scenario to illustrate how the model can be used.

III. THE PROPOSED APPROACH

The proposed approach takes into account the four stages in Kolb's model (concrete experience, reflective observation, abstract conceptualization and active experimentation) in order to enhance learners' learning outcomes and interest. The application domain of this approach is related to computer programming. Learners in this approach follow a specific sequence of learning activities to grasp and understand the concept of programming being introduced.

The initial activities are based on the *concrete experience* stage. Learners perform a specific task with detailed instructions on writing a simple program related to the programming concept. This can give the learners direct experience while performing the task and writing the program.

In the next stage, *reflective observation*, activities are designed to support thinking and analysis abilities of learners. For example, learners are asked to answer how the program works after execution, what is the output of the program and what would happen if a modification to the program were performed.

More activities are also designed following the *abstract conceptualization* stage. Learners at this stage should formulate a hypothesis and understand the theoretical background of the programming concept that was performed and reflected upon in earlier stages.

Then, in the *active experimentation* stage, learners should apply the concept by solving different problems and try out another concrete experience without giving them procedural steps on how to solve or perform the task. In other word, learners should complete a new task or solve a programming problem without detailed instructions.

To illustrate, assume that the programming concept being taught is IF-statement in a programming language. The stages of delivering and teaching this concept are shown in Figure 2.

In the first stage, a programming task with detailed instructions on writing a specific program using IF-statement. Learners do not need to think about how to solve the problem but rather they should obtain the essential experience when writing such a program that uses IF-statements.

Once learners complete the task and have initial concrete experience, learners are asked to think about and analyze the components of the program, how IF-statements are written and how they work besides interpreting the output of the program. This is to support the reflection and observation abilities of learners. Then, learners need to draw a flow-chart diagram of the program, analyze the syntax of the IF-statement and then propose the generic syntax that can be used in different scenarios and problems.

In the final stage, learners should be given specific tasks to solve other problems that require the use of IF-statements. These tasks should not contain detailed instructions. This is to allow learners to map the experience they encountered with the theory they grasped at earlier stages.

A virtual learning tool was then designed based on the proposed approach. It is a Web-based tool implemented using PHP, JavaScript and MySQL, and can be accessed through any Web browser. Each student has a profile, and a number of Java programming lessons were created. The student studies the lessons based on that proposed teaching approach through the learning stages.

IV. METHOD

Usability is an issue that demands investigation since an adaptive learning system may be effective in enhancing learning but can also be difficult to use [14, 15]. Zaharias and Poylymenakou state that "very little has been done to critically examine the usability of e-learning applications" [16]. Therefore, the perceived usability is an important factor and is used to evaluate the virtual learning tool. The factor of perceived usability relates to the ease of use and learnability of the virtual learning tool reflecting the extent to which learners are satisfied with the interaction experience. It is expected that a high level of perceived usability when interacting with the tool leads to more satisfied, engaged and motivated learners which could reflect on their learning achievement [16, 17].

Perceived usability is measured by using the system usability scale (SUS) questionnaire [18], a quick, reliable and widely used test of system usability in both academia and industry [19]. SUS has 10 questions, each offering five responses with anchors ranging from "strongly disagree" to "strongly agree". SUS provides a single score on a scale that is easy to understand to measure overall usability. The score ranges between 0 and 100; the higher the score, the better the usability. Satisfactory systems should have a score between

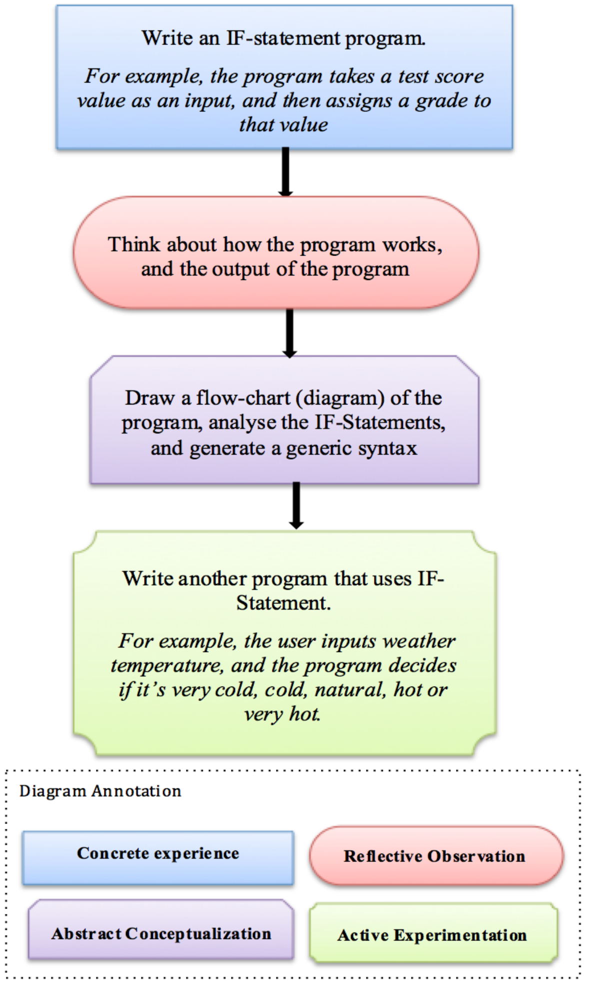

Figure 2. A scenario to teach the concept of IF-statement based on the proposed approach.

70 and 80, while a score higher than 90 indicates an exceptionally usable system [20]. The 10 questions/statements of the SUS tool are as follows [18]:

- *I think that I would like to use this system frequently.*
- *I found the system unnecessarily complex.*
- *I thought the system was easy to use.*
- *I think that I would need the support of a technical person to be able to use this system.*
- *I found the various functions in this system were well integrated.*
- *I thought there was too much inconsistency in this system.*
- *I would imagine that most people would learn to use this system very quickly.*
- *I found the system very cumbersome to use.*
- *I felt very confident using the system.*
- *I needed to learn a lot of things before I could get going with this system.*

An experiment was conducted through three learning sessions of about 90–120 minutes each at the University of the Hail, Saudi Arabia. Participants were informed of the experimental procedure. Then, they randomly assigned to the controlled group and the experimental group. In the control group, learners were asked to learn through a traditional learning tool; participants in the experimental group interacted with the adaptive version of the tool based on learning style according to the proposed approach. Both groups studied the same learning material, and the difference is the material delivery approach. The learning material relate to Java Programming, the application domain of the tool. The material were mainly designed for beginners who have no prior experience in Java. They consisted of four main units: overview, basic syntax, variables and control flow. At the end of the learning session participants completed the usability questionnaire using SUS.

V. RESULTS

The experiment was conducted with 40 participants, 20 males (50%) and 20 females (50%). All participants completed the three learning sessions. The control group consisted of 20 participants, and the same number of participants were assigned to the experimental group. The two groups were also balanced in terms of the number of males and females.

The perceived usability results indicated that the average score of the experimental group is 78 while the average score for the controlled group is 69.25. This may imply that both tools (the traditional version and the adaptive version) are useful and valuable in teaching programming, and the participants found them easy to use.

As to related work, Shi et al. recently developed an adaptive social e-learning system called Topolor and examined its usability using the same tool as employed in this experiment; the overall average score of usability was 75.75 [21]. Cristea et al. also compared three different adaptive e-learning systems using the same tool called MOT, WHURLE and MOT2WHURLE; their usability scores were 75, 66.6 and 60.70, respectively [22].

The usability of the designed adaptive version based on the proposed approach to teach programming is better than the usability of those systems as it scored 78 and even better than the traditional learning version used in this experiment which was designed primarily to investigate perceived usability.

VI. CONCLUSION

This paper presented a specific approach in teaching programming based on a widely used learning style model called Kolb's Experiential Learning Cycle. The paper also offered a specific scenario in how to apply the model for teaching programming. By taking into account the model, learners can have different abilities that are required to understand, recall and apply programming concepts. Although the application area of the model is related to computer programming, the model can be beneficial in other computer science courses such as databases, computer security and artificial intelligence. An experimental evaluation of perceived usability of a virtual learning tool implementing the proposed approach has also been conducted. The results indicated that the adaptive version of the tool based on learning style is more usable than the traditional version of the learning tool.

Regarding the learning model used, it may be criticized for assuming that learning always occurs in linear and ordered steps and for failing to explicitly integrate social and cultural aspects of learning [10]. However, social and collaborative learning can be incorporated in the *reflective observation* stage. Learners can collaborate and work as groups to think about and reflect upon the programming problems. Also, the model has been successfully integrated in an adaptive learning system for teaching computer security [1]. The findings of the study revealed significant improvement to learning outcomes and learning satisfaction of learners. This works differs in applying the Kolb's model in for another domain which is computer programming.

Future work will involve the creation and development of learning materials that support a complete computer-programming course. The learning material will then be validated by domain experts to ensure their quality and their suitability to be mapped to each stage of Kolb's model. When that phase is completed, an experimental evaluation of the approach will be conducted. The experiment will take into account a number of variables including learning outcome, learner motivation and satisfaction to confirm that the proposed teaching approach enhances learning.

VII. REFERENCES

[1] A. Konak, T. K. Clark, and M. Nasereddin, "Using Kolb's Experiential Learning Cycle to improve student learning in virtual computer laboratories," *Computers & Education*, vol. 72, no. 0, pp. 11–22, 2014.

[2] I. Marsa-Maestre, E. De La Hoz, J. M. Gimenez-Guzman, and M. A. Lopez-Carmona, "Design and evaluation of a learning environment to effectively provide network security skills," *Computers & Education*, vol. 69, pp. 225–236, 2013.

[3] P. Brusilovsky, "Adaptive Hypermedia for Education and Training," *Adaptive Technologies for Training and Education*, p. 46, 2012.

[4] A. Mitrovic, "An intelligent SQL tutor on the web," *International Journal of Artificial Intelligence in Education*, vol. 13, no. 2, pp. 173–197, 2003.

[5] S. Schiaffino, P. Garcia, and A. Amandi, "eTeacher: Providing personalized assistance to e-learning students," *Computers & Education*, vol. 51, no. 4, pp. 1744–1754, 2008.

[6] A. Klasnja-Milicevic, B. Vesin, M. Ivanovic, and Z. Budimac, "E-Learning personalization based on hybrid recommendation strategy and learning style identification," *Computers & Education*, vol. 56, no. 3, pp. 885–899, 2011.

[7] M. Alshammari, R. Anane, and R. Hendley, "An E-Learning Investigation into Learning Style Adaptivity," in *The 48th Hawaii International Conference on System Sciences (HICSS-48)*, 2015, pp. pp11-20.

[8] Y. Akbulut and C. S. Cardak, "Adaptive educational hypermedia accommodating learning styles: A content analysis of publications from 2000 to 2011," *Computers & Education*, vol. 58, no. 2, pp. 835–842, 2012.

[9] P. Brusilovsky and E. Millán, "User models for adaptive hypermedia and adaptive educational systems," *The adaptive web*, pp. 3–53, 2007.

[10] D. A. Kolb, *Experiential learning: Experience as the source of learning and development*, vol. 1. Prentice Hall, 1984.

[11] J. W. Keefe, "Learning style: An overview," *Student learning styles: Diagnosing and prescribing programs*, pp. 1–17, 1979.

[12] R. M. Felder and L. K. Silverman, "Learning and teaching styles in engineering education," *Engineering education*, vol. 78, no. 7, pp. 674–681, 1988.

[13] G. Rößling, M. Joy, A. Moreno, A. Radenski, L. Malmi, A. Kerren, T. Naps, R. J. Ross, M. Clancy, A. Korhonen, R. Oechsle, and J. Á. V. Iturbide, "Enhancing Learning Management Systems to Better Support Computer Science Education," *SIGCSE Bull.*, vol. 40, no. 4, pp. 142–166, Nov. 2008.

[14] C. Gena, "Methods and techniques for the evaluation of user-adaptive systems," *The knowledge engineering review*, vol. 20, no. 01, pp. 1–37, 2005.

[15] K. Höök, "Steps to take before intelligent user interfaces become real," *Interacting with computers*, vol. 12, no. 4, pp. 409–426, 2000.

[16] P. Zaharias and A. Poylymenakou, "Developing a usability evaluation method for e-learning applications: Beyond functional usability," *Intl. Journal of Human--Computer Interaction*, vol. 25, no. 1, pp. 75–98, 2009.

[17] C. Ardito, M. F. Costabile, M. De Marsico, R. Lanzilotti, S. Levialdi, T. Roselli, and V. Rossano, "An approach to usability evaluation of e-learning applications," *Universal access in the information society*, vol. 4, no. 3, pp. 270–283, 2006.

[18] J. Brooke, "SUS-A quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, p. 194, 1996.

[19] T. S. Tullis and J. N. Stetson, "A comparison of questionnaires for assessing website usability," in *Usability Professional Association Conference*, 2004, pp. 1–12.

[20] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Intl. Journal of Human--Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

[21] L. Shi, A. I. Cristea, J. G. K. Foss, D. Al Qudah, and A. Qaffas, "A social personalized adaptive e-learning environment: a case study in Topolor," *IADIS International Journal on WWW/Internet*, vol. 11, no. 3, pp. 1–17, 2013.

[22] A. Cristea, H. Ashman, C. Stewart, and P. Cristea, "Evaluation of adaptive hypermedia systems' conversion," in *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, 2005, pp. 129–131.