UNIVERSITI TEKNOLOGI MARA

i-learn UNIVERSITI TEKNOLOGI MARA®

INTERNATIONAL JOURNAL ON E-LEARNING AND HIGHER EDUCATION

VOLUME 4
JANUARY 2016
ISSN 1985-8620

INTERNATIONAL JOURNAL ON E-LEARNING AND HIGHER EDUCATION

International Journal on e-Learning and Higher Education is jointly published by the I-learn Centre and UiTM Press, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia.

International Journal On E-Learning And Higher Education

Vol. 4	January 2016	ISSN 1985-8620

Enhanced Collaborative e-Learning Model with Cognitive Assessment and Open Learner Model

Mahfudzah Othman[1*] and Nurzaid Muhd Zain

[1, 2]Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA Perlis,
02600 Arau, Perlis, Malaysia
fudzah@perlis.uitm.edu.my
[*]Corresponding Author

ABSTRACT

This paper is focused on proposing a model of an enhanced collaborative e-learning system by including the elements of cognitive assessment and open learner model. The main objective is to provide a new andragogical tool to foster self-regulated learning for programming subjects through the online collaborative e-learning platform. The proposed model includes the cognitive assessment module in collaborative e-learning platform such as the assessment of logical thinking skills and metacognitive knowledge among the beginners in programming field. The goal is to provide the students with evaluation mechanisms towards their own cognitive abilities. The open learner model is also included in this model in order to provide qualitative performance representations to the students, where they can reflect their own milestones for each programming topic and monitor their own progress as they improve their programming skills. The open learner model will adapt the SMILI Open Learner Modeling Framework which highlights four important elements that need to be identified, which are the context and evaluation, the learners content that can be viewed, the presentations of the open learner model and the users that gain access to the system. Meanwhile, the iterative waterfall model is being proposed in this paper to be used as the overall methodology that comprises six main phases which are feasibility study, requirement gathering and analysis, design, implementation, evaluation and maintenance.

Keywords: *cognitive assessment, collaborative e-learning, learner model, open learner*

INTRODUCTION

Computer programmers are still in demand these days due to the increasing needs of application software, mobile applications and web applications. Despite the fact that computer programmers are still relevant these days, learning to program is generally considered hard, and the enrolments in programming courses in tertiary levels were reported to decline worldwide (Koorse et al., 2015). Over the past few years, high failure rates in programming courses have also been recorded and reported in many colleges and universities from all around the globe. A recent study done by Watson and Li (2014) that involved 15 different countries including the United States, Australia, United Kingdom, Finland, China and Indonesia has revealed that the average passing mark for computing programming courses was only at 67 percent.

For many years, researchers and academicians have doubled their efforts to identify the factors that contribute to the high failure rates in computer programming courses. The most common cause is the nature of the programming subject itself that demands high level of intellectual capabilities and often being related to the engineering activities as claimed by Valentin et al. (2013). Other possible causes are the lack of interest and motivation in learning programming, lack of prior knowledge, the technical nature of the programming languages used and differences in teaching and learning styles and strategies as mentioned by Kalelioglu and Gulbahar (2014).

Meanwhile, cognitive abilities have also been claimed as one of the factors that determine the success or failure in introductory programming courses (Othman et al., 2015). Among the cognitive abilities required in becoming a successful programmer are metacognitive and problem-solving skills, analytical and critical thinking, as well as reasoning and logical thinking skills (Havenga et al., 2013; Osman & Maghribi, 2015). By using these skills, students should be able to analytically and logically analyze the given problems and provide the right solutions. These steps are likely to be the most important steps in Program Development Life Cycle (PDLC) (Mazlan & Othman, 2015). Previous studies have also revealed that students with under developed cognitive abilities will fail to grasp the basic problem-solving concepts of programming, consequently will lead them to be less motivated or even withdrawing from the course (Mazlan & Othman, 2015; Othman et al., 2015).

The typical traditional teaching and learning method in programming classes or practical sessions have also played an influential factor that determines students' levels of understanding and engagement in learning programming. The linear approach that starts by introducing the students with the basic of programming language and later guide them towards finding the best strategies in solving programming problems only involved the use of blended materials and lecture notes (Osman & Maghribi, 2015; Zain & Paidi, 2015). Minimal interaction between students and lecturers will eventually lead the students to become passive information receivers, especially when large groups of students are involved (Zain & Paidi, 2015).

Over the past few years, varieties of new strategies, techniques and methods have been studied, introduced and employed in teaching and learning programming in order to overcome these issues. Some researchers have applied the use of concept maps and other visual instructional strategies and techniques (Osman & Maghribi, 2015; Zain & Paidi, 2015). Others include studies about the differences of students' logical thinking skills, cognitive abilities and personality profiles as the predictor of success in computer programming (Othman et al., 2015). Meanwhile, previous studies have also showed interest in incorporating collaborative learning or pair programming techniques in physical or online classrooms to encourage active learning and enhance students' engagement in learning programming (Othman et al., 2015; Zain & Paidi, 2015).

Recent studies have also utilized the Information and Communication Technologies (ICT) to enhance teaching and learning programming such as multimedia and interactive games, mobile applications and e-learning platforms (Tillmann et al., 2013; Valentin et al., 2013). Meanwhile, the emergence of numerous collaborative e-learning platforms has also shown positive outcomes in supporting and facilitating teaching and learning for programming courses such as the Online Collaborative Learning System (OCLS) designed by Othman et al. (2013), Supporting Collaboration and Adaptation in a Learning Environment (SCALE) by Verginis et al. (2011) and AutoLEP by Wang et al. (2011). All of these collaborative e-learning systems offer multi-benefits towards supporting the teaching and learning process for programming subjects as well as improving students' learning styles and strategies.

Although there have been countless efforts and studies done to improve the students' programming skills, particularly in using the collaborative e-learning platforms, most of the collaborative e-learning platforms do not provide users with cognitive assessments and open learner models. Cognitive assessments such as the Group Assessment Logical Thinking (GALT) test is usually being done using pen and paper with its main goal is to identify the differences of students' cognitive abilities (Roadrangka et al., 1983). Meanwhile, the existing open learner models are often presented separately from the e-learning systems and mostly developed to cater individual learner models rather than providing learner models for groups of learners (Clayphan et al., 2014).

Therefore, the purpose of this study is to propose an andragogical solution by designing a model for the enhanced collaborative e-learning system that includes both the cognitive assessment module and the open learner model. The aim of this study is to raise students' awareness towards their own personal and collaborative groups' achievements through self-assessment and self-regulated learning. The reflection of performance through the open learner model will allow the students to see their own developing knowledge, difficulties and learning process (Clayphan et al., 2013). Open learner model can also provide mechanism to observe learners' behaviour and qualitative representations of the learner's cognitive and affective knowledge (Bull & Kay, 2007). To date, there are only a few researchers who have embarked on combining the online collaborative platforms with the learner models such asBull and Kay (2007), Bull and Vatrapu (2011), Alotaibi and Bull (2012), Clayphan et al. (2014) and Kickmeier-Rust et al. (2014). Therefore, this open learner modeling technique with the cognitive assessment module in the enhanced collaborative e-learning system can be a potential medium to encourage metacognitive activities that will influence the collaborative groups to take greater responsibilities towards their learning in programming. Hence, it would help the students to improve their programming skills.

RELATED WORKS

Collaborative E-learning System for Programming

The development of the e-learning systems for teaching and learning programming are positively progressing towards the Computer-Supported Collaborative Learning (CSCL) system such as the Online Collaborative Learning System (OCLS) by Othman et al. (2013) and Supporting Collaboration and Adaptation in a Learning Environment (SCALE) by Verginis et al. (2011). These collaborative e-learning systems were claimed to be effective in facilitating online group collaborations and incorporate social networking features. For instance, with SCALE, students were expected to be actively involved with their own improvement in programming by referring to the feedbacks and recommendations given by the system. Students will become increasingly aware of their own performance in programming courses via SCALE (Verginis et. al., 2011). Other research involves the development of a web-based system named AutoLEP that was developed by Wang et al. (2011) to help novice programmers in attaining their programming skills by providing novel assessment mechanism that allows users to test and evaluate their own programs. Students' learning experiences were claimed to be improved via this system (Wang et al., 2011).

To date, most of these collaborative e-learning platforms do not provide the students with the cognitive assessment module and open learner models. Students need to be tested with programming questions that are equivalent to their cognitive abilities and also at the same time being guided to improve their cognitive milestones via the collaborative e-learning platform. Moreover, the reflection of their achievements in programming based on the cognitive levels that will be displayed by the open learner model will help the students to understand more about their own cognitive abilities. Eventually, they will find ways to improve which will lead them to enhance their programming skills too.

Cognitive Assessments

There are varieties of cognitive assessments that have been widely implemented in the field of education such as Metacognitive Awareness Inventory (MAI) and GALT. For instance, MAI is used to measure the students' metacognitive knowledge. In solving programming problems, metacognition role has been proven important where previous study has identified that student who performs well in programming, eventually facilitates more metacognitive skills than the lower achievers (Havenga et al., 2013). In addition, the linear model of problem-solving development lifecycle actually demands metacognitive control over planning, monitoring and evaluation of the solution of the problem, traits that are important in computer programming (Havenga et al., 2013). Therefore, it is equally important for the students to be aware of their own metacognitive levels of knowledge in order to foster active engagements with their own progress in learning computer programming.

Meanwhile, for this study, GALT test will be incorporated in the cognitive assessment module for the collaborative e-learning system. This logical thinking test has been widely utilized in the teaching and learning fields and the Cronbach's alpha reliability coefficient is 0.52, which is considered moderate to be used in this study. GALT test was first developed by Roadrangka et al. (1983) with six subscale measures for logical operations as depicted in Table 1. The subscale measures cover the tests for conservational reasoning, proportional reasoning, controlling variables, probabilistic reasoning, correlational reasoning and combinatorial reasoning.

Table 1: The Six Subscale Measures for Logical Thinking in GALT Test

Subscales	Item No.	Item Descriptor
Conservational reasoning	1	Piece of Clay
	2	Metal Weigh
Proportional reasoning	3	Glass Size
	4	Scale
Controlling variables	5	Pendulum Length
	6	Ball
Probabilistic reasoning	7	Square and Diamonds #1
	8	Square and Diamonds #2
Correlational reasoning	9	The Mice
	10	The Fish
Combinatorial reasoning	11	The Dance
	12	The Shopping Centre

Open Learner Model

There are many types of open learner models to be adapted in the online platforms either as simple or complex representations. For instance, the simple learner models will present simple information to the learners such as the learner's level of knowledge for every selected topic. This simple information will be represented using skill meters and covers the expert knowledge, problematic areas and misconceptions (Bull & Kay, 2007). Meanwhile, the complex learner models will present information that is more thorough to the learners by facilitating varieties of modeling techniques such as knowledge tracing in cognitive modeling or Bayesian networks (Bull & Kay, 2007). Previous online systems developed using the open learner models, involved works done by Bull and Kay (2007) called SMILI and OMLlets.

Open Learner Model for Collaborative Platforms

To this day, open learner models are normally being developed to represent learner models for individual learners. Thus, combining the open learner model in the collaborative e-learning platform has been seen as a new direction in Computer-Supported Collaborative Learning (CSCL) research.This will provide open learner model to groups of learners (Bull &Vatrapu, 2011). In recent study done by Kickmeier-Rust et al. (2014), an open learner model has been developed together with the competence-based feedback for collaborative language learning. Other than that, open learner model has also been developed for scaffolding students' reflection towards collaborative brainstorming (Clayphan et al., 2014). Another interesting study, conducted by Alotaibi and Bull (2012), has used the social network platform such as the Facebook with the combination of an open learner model named OLMlets to investigate the effectiveness of online interaction and collaboration. Open learner model offers many benefits to the learners as mentioned by Bull and Kay (2007), which some of the benefits are; the open learner model helps to promote metacognitive activities such as reflection, planning and self-assessment, supports navigation and facilitates groups collaborations.

Therefore, due to the fact that the open learner model has plenty of beneficial factors to offer to the students and lecturers, it can be seen as a potential effort in supporting groups collaboration and discussion as well as fostering self-regulated learning in the collaborative e-learning environment.

PROPOSED MODEL OF AN ENHANCED COLLABORATIVE E-LEARNING WITH COGNITIVE ASSESSMENT AND OPEN LEARNER MODEL

This section discusses the proposed model of an enhanced collaborative e-learning platform with open learner model and cognitive assessment modules embedded into it as depicted in Figure 1. Based on Figure 1 below, the students will be divided into small collaborative groups and later engage with the collaborative activities by answering programming questions constructed in the e-learning contents. The lecturers will do the division of the collaborative groups manually by mixing the low achievers with the high achievers in each collaborative group. This is to provide a platform for the low achievers to discuss and learn more from their high achievers' friends.

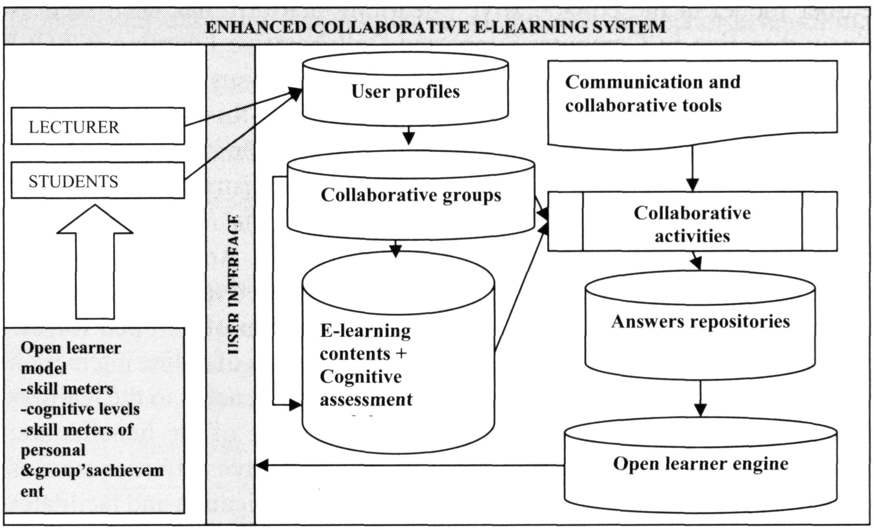

Figure 1: Enhanced Collaborative E-Learning System with Cognitive Assessment Module and Open Learner Model

Meanwhile, the development of the e-learning contents will include two cognitive assessment modules which are the logical thinking test module and the programming questions module that will cover all the main topics from the introductory programming subject and will be constructed based on Bloom's Taxonomy cognitive domains. His taxonomy cognitive domains have six main stages of cognitive developments, which are knowledge (C1), comprehension (C2), application (C3), analysis (C4), synthesis (C5), and evaluation (C6) which have been widely implemented in the field of education and is aimed to ensure that the students will achieve their cognitive milestones.

The results from both cognitive assessment modules will be displayed in the open learner model to represent the students' achievements in each programming topic. The open learner model proposed in this collaborative e-learning system will include displays of skills meters for individual achievements and collaborative groups' performance. The skill meters will not only highlight the positive achievements, but will also highlight the problematic areas or misconceptions for each individual student and their collaborative members in order to create awareness on their own and group development and progress in programming subject as well as encouraging self-regulated learning and active discussion in the virtual environment.

MATERIAL AND METHOD

In order to develop the enhanced collaborative e-learning system with the cognitive assessment modules and the open learner model as proposed in the previous section, an iterative waterfall model has been used as the methodology. The iterative waterfall model as depicted in Figure 2 below involves six main phases which are i) feasibility study ii) requirement analysis, iii) design, iv) implementation, v) evaluation and vi) maintenance.

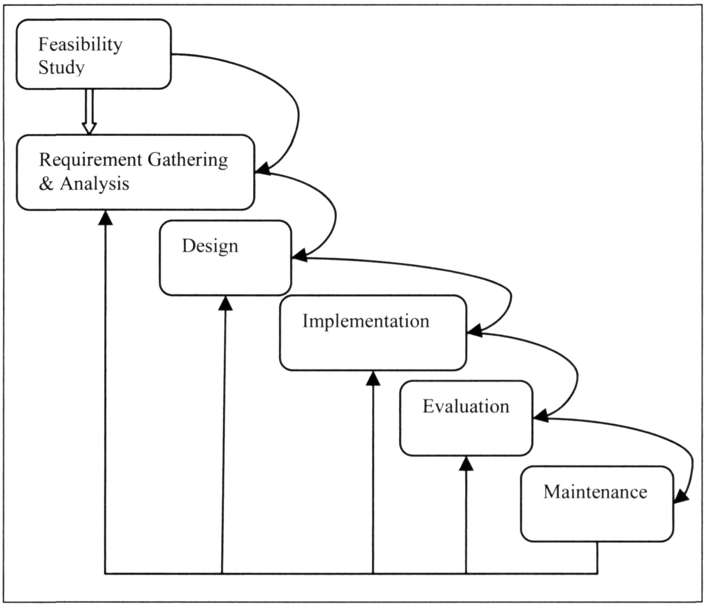

Figure 2: The Iterative Waterfall Model

Feasibility Study

During this phase, interviews and questionnaires were conducted to determine whether a new or improved system is a feasible solution. The activity in this phase involved the collection and analysis of different information and data items that were needed as input, process and outputs. Problems have also been defined and feasible solutions were strategized and evaluated in the feasibility study.

Requirement Gathering and Analysis

The second phase of this study involved requirement gathering and analysis where firstly, the research population and samples were determined. The population of this study are the Computer Science lecturers and students working and enrolled in Computer Science Department at UniversitiTeknologi MARA (UiTM) Perlis. The sample of this study consisted of male and female students enrolled in the first semester classes, where in each of these classes programming courses are taught to heterogeneous classrooms with no grouping or ability tracking. The goal of this requirement gathering and analysis is to identify the users' views and needs to ensure the developed system can fulfill the requirements needed. Finally, the analysis of the requirements was represented using the Entity-Relationship Diagram (ERD) and Data Flow Diagram (DFD).

Design

The overall design and development of the cognitive assessment modules in the e-learning contents involved the use of web-based application tools such as MySQL for the database, PHP and Apache web server. The original version of GALT test that comprised 12 questions was designed and transformed to suit the online interface where the students' answers will be saved in the database. The individual scores for the logical thinking test were then displayed in the final open learner model.

Meanwhile, the design of the open learner model for the enhanced collaborative e-learning system was adapted from the SMILI Open Learner Modeling Framework as proposed by Bull and Kay (2007). By referring to this framework, four elements were identified, which are the context and types of evaluation, the contents that are open for the learners to view, the graphical representation of the open learner model and the actors that will be granted the full access to the system.

The context and types of evaluation for the open learner model designed and developed for this study mainly focused on the overall interactions of the learners with their learner models. This was done by firstly determining the topics in the Fundamentals of Computer Problem Solving subject as the e-learning contents and the types of the programming questions that will be asked in the enhanced collaborative e-learning system. In this study, both multiple-choice and short structured questions were constructed based on Bloom's Taxonomy cognitive domains and represented in the e-learning contents for the collaborative activities.

Meanwhile, based on the SMILI Open Learner Modeling Framework, the open learner model designed for this study will support the accuracy, learner reflection on knowledge and understanding, learners monitoring their learning, as well as group collaboration. The learners were also being granted to navigate between questions if they are not satisfied with their learner models. This is to encourage the learners to identify their own problematic areas or misconceptions in order for them to improve and finally achieve their targets.

For the graphical representation of the open learner models designed for the enhanced collaborative e-learning system, the simple presentation such as skill meters were used to display the individuals or groups' achievements. Three stages will be displayed on the skill meters based from the questions answered in the collaborative activities, which are the percentage of correct answers, misconceptions, and questions that have not been answered. Lastly, the main actors identified for the enhanced collaborative e-learning system are the lecturers as the system administrators and the students as the learners. Both actors were given full interactions with the collaborative activities, cognitive assessment and open learner models.

Implementation, Evaluation and Maintenance

Currently, the development of the enhanced collaborative e-learning system with the cognitive assessment and open learner model is in its design and development stage. Therefore, the future work for this study will involve the implementation of this enhanced collaborative e-learning system in the programming classes in UiTM Perlis. Usability testing and user acceptance testing have also been scheduled to be conducted once the system has been fully developed and implemented.

RESULTS AND DISCUSSION

This section discusses on the design and development of the cognitive assessment modules and open learner model for the enhanced collaborative e-learning system.

Cognitive Assessment Module: Logical Thinking Test

For the first cognitive assessment module, each student needs to answer a logical thinking test. The logical thinking test used in this study is the GALT test that has been described in the previous section. Figure 3 below depicts an example of the logical thinking test that has been embedded in the collaborative e-learning system. Altogether, there were 12 questions that the students need to answer and the results will be displayed in the open learner model after they have finished all the cognitive assessment modules.

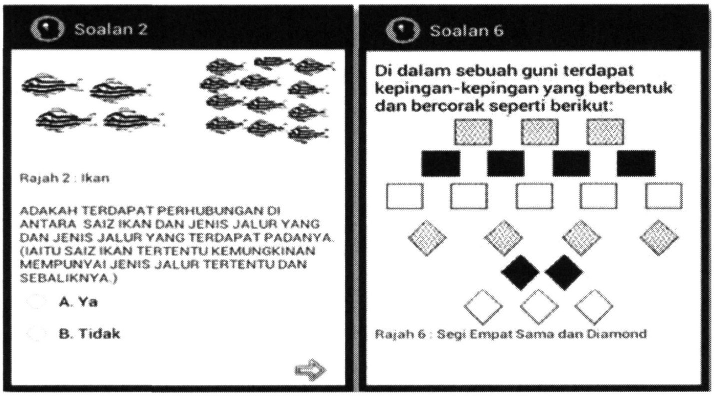

Figure 3: Example of Logical Thinking Test

Cognitive Assessment Module: Programming Questions based on Bloom's Taxonomy Cognitive Domains

In this module, each of the students in the collaborative groups was asked to answer series of questions posted by the lecturers in the system, where each of the questions represented each topic from the Fundamentals of Computer Problem-solving subject. There were two types of questions that have been constructed, namely the multiple-choice questions and short structured questions. The students were also been given options to answer questions with different cognitive levels. Figure 4 below shows the example of the multiple-choice questions with the cognitive levels. The collaborative group members were allowed to discuss with each other via the chat room available in the system. This is to support group's collaboration, communication and discussion among the group's members from dispersed locations.

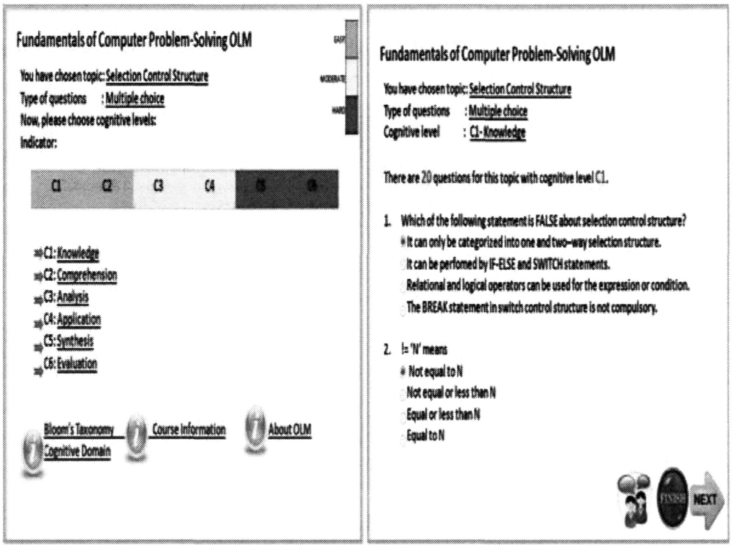

Figure 4: Example of Multiple-Choice Questions with Cognitive Levels

Open Learner Model

This module will represent the students with their individual and group's open learner models as depicted in Figure 5. Each open learner model represents the skill meter that reflects the individual and collaborative group's achievements for logical thinking tests and for each programming topic that they have previously answered. The open learner model designed for this study will display three stages of performance for each student in the collaborative group, which are the correct answer, labeled in blue colour, misconceptions with red colour and have not answered questions in green colour. The goal is for them to be aware of their own performance and progress in the particular programming topic, and if they are not satisfied with the initial results, they can always return to the cognitive assessment modules and execute the test again. The open learner model is also aimed to foster self-regulated learning where the students will be increasingly aware with their own learning progress and how they can improve their learning and understanding in programming through the collaborative e-learning platform.

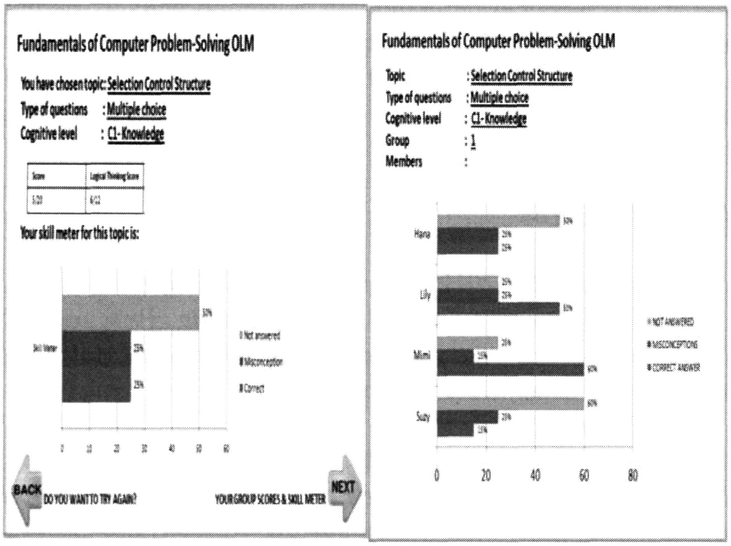

Figure 5: Example of Individual and Collaborative Group's Skill Meters

CONCLUSION AND FUTURE WORK

As a conclusion, the proposed model of the enhanced collaborative e-learning system can also be used as a platform for group collaborations, cognitive assessments and representations of open learner models for individual or groups of learners. The open learner model concepts proposed in this study will represent the learners' cognitive achievements such as logical thinking skills, and cognitive developments based on Bloom's Taxonomy cognitive domains. Other than that, the open learner model designed for this study will also help to highlight the students' problematic areas or misconceptions in each programming topic based on the questions that they have answered in the cognitive assessment modules. From there, students can directly cater the problematic topics by doing extra exercises in the e-learning system or simply execute the cognitive assessment modules again until their skill meters show some improvements. The graphical display of the individual or group's performance will be constantly changing as the students improve their programming skills and progressing in this subject. By referring to these achievements, students will be increasingly aware of their own performances and become more engaged with their own personal achievements in programming as well as actively participate in group discussions via the collaborative e-learning platform. Future work for this study will embark on the full development, implementation and evaluation of this system that involves usability and user acceptance tests.

REFERENCES

Alotaibi, M. & Bull, S. (2012). Combining Facebook And Open Learner Models To Encourage Collaborative Learning. *Proc. of 20th ICCE Conference on Computer-supported Collaborative Learning (CSCL) and Learning Sciences,* Singapore, 71-75.

Bull, S. & Kay, J. (2007). Student Models That Invite The Learner In: The SMILI Open Learner Modeling Framework.*Int. Journal of Artificial Intelligence in Education, 17*(2), 89-120.

Bull, S. & Vatrapu, R. (2011). Supporting Collaborative Interaction With Open Learner Models: Existing Approaches And Open Questions, *Proc. of 9th International Computer-Supported Collaborative Learning (CSCL)*, China, 761-765.

Clayphan, A., Martinez-Maldonado, R. & Kay, J. (2013). Designing Olms For Reflection About Group Brainstorming At Interactive Tabletops. *Proc. of the Workshop on Intelligent Support for Learning in Groups (ISLG) - International Conference on Artificial Intelligence in Education (AIED 2013)*, Memphis: USA.

Clayphan, A., Martinez-Maldonado, R., Kay, J. & Bull, S. (2014). Scaffolding Reflection For Collaborative Brainstorming, In Trausan-Matu, S, Boyer, KE, Crosby, M, Panourgia, K, editors. *Intelligent Tutoring Systems*, Springer International Publishing, Switzerland, 615-616.

Havenga, M., Breed, B., Ments, E., Govender, D., Govender, I., Dignmu, F. &Dignum, V. (2013). Metacognitive And Problem-Solving Skills To Promote Self-Directed Learning In Computer Programming: Teachers' Experiences. *SA-Educ JOURNAL*, *10*(2), 1-14.

Kalelioglu, F. & Gulbahar, Y. (2014). The Effects Of Teaching Programming Via Scratch On Problem Solving Skills: A Discussion From Learners' Perspective, *Informatics in Education*, *13*(1), 33–50.

Kickmeier-Rust, M.D., Bull, S. &Meissl-Egghart, G. (2014). Collaborative Language Learning In Immersive Virtual Worlds: Competence-Based Formative Feedback And Open Learner Modeling, *International Journal of Serious Games; 1*(2), 67-74.

Koorse, M., Cilliers, C.& Calitz, A. (2015). Programming Assistance Tools To Support The Learning Of IT Programming In South Africa Secondary Schools, *Computers & Education*, *82*(2015), 162-178.

Mazlan, U.H. & Othman, M. (2015). Fundamentals Of Algorithm Design Course: Issues, Challenges & Proposed Teaching-Learning Approaches,*Proc. Colloquium in Computer & Mathematical Sciences Education (CCMSE 2015)*, August 2015, 18-22.

Osman, M.N. & Maghribi, M. (2015). Capturing Best Practice In Teaching Introductory Programming Course: A Case Study For Non-Computer Science Students. *Proc. Colloquium in Computer & Mathematical Sciences Education (CCMSE 2015)*, August 2015, 1-6.

Othman, M. Ekhsan, H.M., Osman, M.N., Hamid, J.N., Zain, N.M, Jamaluddin, M.N.F, Mazlan, U.H., Jan, N.Y.C & Ahmad, H. (2015). Enhancing Logical Thinking And Reasoning Skills Through Collaborative Learning In Programming, *Proc. Colloquium in Computer & Mathematical Sciences Education (CCMSE 2015)*, August 2015, 12-17.

Othman, M., Othman, M. & Hussain, F.M. (2013). Designing Prototype Model Of An Online Collaborative Learning System For Introductory Computer Programming Course. *Procedia-Social and Behavioral Sciences 90*, 293-302.

Roadrangka, V., Yeany, R.H. & Padila, M.J. (1983). *The Construction And Validation Of A Group Assessment Of Logical Thinking (GALT).* Paper Presented At The Meeting Of The National Association for Research in Science Teaching, Dallas, TX.

Tillmann, N., de Halleux, J., Xie, T., Gulwani, S., Bishop, J. (2013). Teaching And Learning Programming And Software Engineering Via Interactive Gaming, *Proc. 35th International Conference on Software Engineering (ICSE)*, May 2013, 1117-1126.

Valentin, L.F., Pardo, A. & Kloos, C.D. (2013). Addressing Drop-Out And Sustained Effort Issues With Large Practical Groups Using An Automated Delivery And Assessment System, *Computers & Education*, *61*, 33-42.

Verginis, I., Gogoulou, A., Gouli, E., Boubouka, M. & Grigoriadou, M. (2011). Enhancing Learning In Introductory Computer Sciences Courses Through SCALE: An Empirical Study. *IEEE Trans. Edu.*, *54*(1), 1-13.

Wang, T.T., Su, X.H., Ma, P.J., Wang, Y.Y. & Wang, K.Q. (2011). Ability-Training-Oriented Automated Assessment In Introductory Programming Course. *Computers & Education, 56*, 220–226.

Watson, C. & Li, F.W.B. (2014). Failure Rates In Introductory Programming Revisited, *Proc. 2014 Conference On Innovation &Technology In Computer Science Education (ITiCSE '14)*, 39-44.

Zain, N.M. &Paidi, Z. (2015). Pair Programming: An Overview, *Proc. Colloquium In Computer & Mathematical Sciences Education (CCMSE 2015)*, August 2015, 7-11.