International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 11 (2016) pp. 395-405

Student's Engagement In Computer-Based Learning: Skill-Challenge Balancing Technique for Learning Programming

Norliza Katuk, Nur Haryani Zakaria, Alawiyah Abd Wahab and Osman Ghazali

School of Computing, Universiti Utara Malaysia,
06010 UUM Sintok, Kedah, Malaysia
k.norliza | haryani | alawiyah | osman@uum.edu.my

***Abstract*: Students' engagement in computer-based learning activity is hard to regulate and maintain at an optimal level. Hence, many students do not benefit from it, and they tend not to use it again after the first attempt. This is a challenge towards ensuring the sustainability of future computer-based learning. In order to overcome this issue, this research proposed skill-challenge balancing (SCB) as a technique to help students to engage in the computer-based learning activities. The technique was employed in a web-based learning tool named LearnJava. An experimental study was conducted using LearnJava for learning a two-hour basic Java lesson. Prior to the interaction, the students were given with a pre-test and post-test questions related to the topic as a mechanism to measure learning. Then a self-administered questionnaire was given to the student as an instrument to measure engagement. The results of the study suggested that learning process happened when the students interacted with LearnJava. Further, the students were engaged in the learning activities offered in LearnJava. The finding of this study can be beneficial for computer-based instructional designers for designing applications that can help students to engage with the learning process.**

***Keywords*: web-based learning, Java programming, cognitive engagement, optimal engagement**

I. Introduction

Java is a powerful programming language that was developed by Sun Microsystems in 1995. In Universiti Utara Malaysia (UUM), Java programming course is taken by students who enrol in Information Technology (IT), Multimedia, Education, Business Mathematics and Decision Science programmes. Students learn Java through a combination of lectures, laboratory exercise, and tutorials. The content covers basic syntax to the extent that covers areas such as threaded programming and Graphical User Interface (GUI). Java can be used to develop various software applications, and it can be run in a cross-platform environment. It can be used in a variety of computing platforms, from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end. Nowadays, Java is used almost everywhere [1, 2] including in mobile phones, web servers, and enterprise applications.

The main reason that Java is widely used is due to its portability [3, 4] and flexibility [5, 6]. It is available for free from Sun's Solaris. It is also available for Microsoft Windows platforms, Linux, and Macintosh. Since Java programs run on multi-platform, students who develop programs in Java do not need to change their code to adapt to different operating systems. Furthermore, Java is growing to be a significant model for concurrency learning, networking, computing, interactive, and, object-oriented design [7].

As mentioned above, Java programming is a fundamental course for students who enrol in IT-related programs. Our initial investigation suggested that students have difficulty in learning to program. This is shown in the examination results for the Introductory to Programming (STIA1014) where 38% of 130 students failed (i.e., C- or below) in semester September 2014. This is quite an alarming scenario for School of Computing (SOC). Hence, the solution to this problem through technological support is deemed necessary. To address this issue, a computer-based learning software can be introduced to support teaching and learning for the course.

An important aspect of computer-based learning is engagement [8]. Unlike traditional classroom learning, student's engagement with a computer-based learning activity is difficult to observe. Hence, it is hard to regulate individual student engagement to an optimal level [9]. In the current computer-based environment, an individual student's engagement in a particular learning activity is entirely dependent on the student's intrinsic motivation. In other words, most current computer-based learning systems do not have the capabilities to control and manipulate student engagement at an optimal level. The absence of this mechanism is a challenge to ensure the sustainability of future e-learning. Therefore, a technique that can enhance students' engagement in computer-based systems particularly for learning Java is highly needed. This has become the motivation for this study.

The aim of this research is to promote a technique that can be embedded in computer-based learning system named skill-challenge balancing (SCB). The technique aims to improve students' engagement when they use computer-based learning especially in learning Java programming course. Further, we interested to study whether the technique could help the students to engage in the computer-based learning activity.

The paper is divided into six sections. First, we present the overview of the computer-based learning experience and its relationship to engagement. Then, it is followed by an overview of SCB technique and its implementation in a learning tool named LearnJava. The following section explains the methodology for conducting the study. The results of the experimental study are presented in the subsequent section. Then, the results are discussed and concluded in the last section.

II. Computer-based Learning Experience and Students' Engagement

Many past studies measured students' performance in e-learning environment; however, only a few studies investigated their learning experiences. As students are the target audience of e-learning systems, their experiences are important to improve the quality of learning [10].

In many studies, students' experiences with e-learning are examined in various contexts. For example, Deepwell and Malik [11] investigated experience in the perspective of students' expectations of the technology, the lecturers' engagement with technology and how the technology might support processes of transition in the higher education sector. Paechter et al. [12] defined experience from the context of course design, interaction with the instructors, interaction with students, individual learning processes and course outcome. Gilbert et al. [10] stated that students' experiences are similar to students' satisfaction towards learning in the e-learning environment. In aggregation, it seems that students' experiences are involved with students' perceptions on a particular issue concerning e-learning. It includes how a student perceives about the design of a course, the design of user interface, interaction with tutors, interaction with peer students, learning processes, and learning outcomes.

The student experiences in this research refer to some states or conditions, which a student might undergo during his or her individual computer-based learning processes and interactions. It measures the student's learning conditions and internal cognitive states while engaging in a particular e-learning activity. Simply said, student experiences can be described by how much an individual student engages in a particular computer-based learning activity.

Indeed, it is not able to measure precisely how much a student engages in a particular computer-based learning activity. In the traditional classroom setting, students' engagement can be mostly observed by a teacher. Hence, an experienced teacher could easily know whether or not a student is fully engaged in a learning activity. For instance, a teacher might see the situation through the student's gestures or face reading in responding to a learning activity. Usually, a teacher will then take some actions so that the student could engage again in the activity and achieve an optimal engagement in the activity.

As mentioned above, student experiences can be described by "how much an individual student engages in a particular computer-based learning activity". It is very true that this question could lead to some subjective answers. One may say that he or she is fully engaged, a little bit engaged, or not engaged at all. The states of engagement (or disengagement) are very elusive and difficult to quantify. Hence, some studies described engagement through combinations of a few characteristics such as attention, concentration, control, and enjoyment, to name but a few. Others have tried to examine engagement or disengagement through the use of some possible cognitive or behavioural states.

As an example, Sharafi et al. [13] suggested the engagement mode (EM) model in describing engagement in IT acceptance. The model describes five engagement modes in which a user may experience using an IT product: (i) enjoying/ acceptance, (ii) ambition/curiosity, (iii) avoidance/hesitation, (iv) frustration/anxiety, and (v) efficiency/productivity. The EM model assumes that when a subject (e.g., an IT user) is engaged in an object (e.g., IT systems), he or she may experience different modes of engagement. It depends on three factors: (i) the positive or negative effects of the object, (ii) locus of control between subject and object, and (iii) dimensions of motivation. A subject who has high extrinsic motivation and is capable of controlling an IT product may gain efficiency/productivity from the technology. On the other hand, an extrinsically motivated user might experience frustration or anxiety when he or she is unable to have control of the IT product. Generally, frustration/anxiety and avoidance/hesitation fall under negative experiences while pleasure/acceptance, efficiency/productivity, and ambition/curiosity are considered as positive experiences.

A more systematic definition of engagement is found in a study by O'Brien and Toms [14]. The study described engagement as a quality of user experiences characterised by ten attributes: challenge, positive affect, endurability, aesthetic and sensory appeal, attention, feedback, variety/novelty, interactivity, and perceived user control. Through an exploratory study, the research suggested that engagement in computer-based systems is a process comprised of four stages: (i) point of engagement, (ii) period of sustained engagement, (iii) disengagement, and (iv) reengagement. A person will remain in the engagement stage as long as he or she can maintain his or her attention and interest in the computer-based system. On the other hand, if a person could not sustain his or her attention towards the system, the stage changes from engagement to disengagement. Disengagement from a particular computer-based system results in either positive (e.g., feeling of success and accomplishment) or negative (e.g., uncertainty, frustration, boredom) experiences.

It is important to bear in mind that individual engagement in a particular activity is forced by either intrinsic or extrinsic factors [15]. Intrinsic factors motivate a person to perform a particular activity for no apparent reinforcement, rather for the sake of the activity itself [16]. The person chooses to perform a particular activity (or task) because of the sense of accomplishment or satisfaction derived when the activity is completed. Engagement forced by the intrinsic factors would give a feeling of enjoyment to a person.

III. Skill-Challenge Balancing Technique

A. Overview

The skill-challenge balancing (SCB) technique aims to improve students' engagement in e-learning systems [17, 18]. Flow theory [20, 21] was the underlying principle of SCB, in which it suggested that an optimal experience could be achieved when the level of the given challenge matches the individuals' levels of skill in performing particular activities as shown in Figure 1. The given level of challenge that is equivalent to the current level of skill will ensure that a person in optimal engagement or also called as flow. The unequal level of the given challenge with a person's level of skill will lead to either boredom or anxiety. It is also important to note that individual levels of skill are progressing over time and similarly for the level of the given challenges.

The main concept in the SCB technique is to allow a flexible adjustment of the given level of challenge. In order to keep the students in an optimal cognitive engagement, the given levels of the challenge must always be comparable to the students' current level of knowledge. In other words, students' current levels of knowledge (or skill) must be able to cope with the given levels of challenge.

The core of the SCB technique is to allow the students to have self-assessments of their individual levels of knowledge throughout the computer-based learning session as described in self-determination theory [22]. The students are given a chance to self-evaluate whether the learning unit is too easy or too difficult for them. If the students find that the learning unit is too easy, they can choose to move forward to a higher level of difficulty of the learning unit. On the other hand, if the students find that the learning unit is too difficult, they are able to move backwards to a lower level of difficulty of the learning unit.

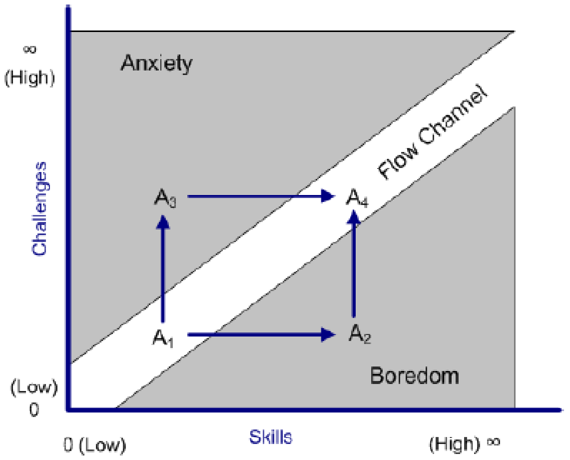

Figure 1. Flow, anxiety, and boredom states by Csikszentmihalyi [20, 21]

The technique was proven effective for maintaining students' engagement in computer-based learning system for Basic Computer Network; a type of declarative knowledge [17-19, 23-28]. In this paper, we extend the implementation of SCB technique for learning Java Programming course, a type of procedural knowledge. We are interested in studying the effect of the SCB technique on students' engagement while they are interacting with computer-based learning system for Java programming.

B. The Implementation of SCB

The SCB was implemented in a web-based application named LearnJava, an application for learning Java programming language. As described in the previous sub-section, SCB works by allowing flexible adjustment of the level of challenge. During the process of learning Java, students must learn the basic of programming knowledge, such as translation of the code into machine language. Then, students must learn step-by-step of the syntax before they can construct a simple Java program. Students also learn how to use variables and constants in the program to perform simple tasks. In other words, the process of learning programming language starts with understanding simple concepts and then followed by complex concepts, one before another. Students would not be able to comprehend and develop a program if they started with the complex concepts before the fundamentals. The progression from simple to complex processes shows that challenges are increasing from an easy to complex level. In a learning process, students' skills and knowledge should also be progressing at the same pace with the increasing level of learning difficulties.

In LearnJava, students can perform two main activities of learning that are (1) browsing and understanding the lecture notes, and (2) answering the quiz. Figure 2 shows the basic flow of how SCB is implemented in LearnJava. When students successfully login to the system, they can choose the learning activity. Students may start the learning activity by answering the quiz. If students do not know the answer to the quiz, they are allowed to browse and learn the learning contents associated with the question. As they completed the learning contents or they found the answer, the students can get back to the quiz and continue answering it until they complete the whole chapter. This is how the flexible adjustment of challenge in SCB works and being demonstrated in LearnJava.

LearnJava is a web-based learning tool to support classroom learning for students who enrol in Introductory to Programming at SOC, UUM. The application covers eleven chapters from Introduction to Computer and Programming to Graphical User Interface. Figure 3 shows the coverage of topics in LearnJava. The learning content is presented in English as the programming course was also conducted using the language due to the university's policy for internationalization.

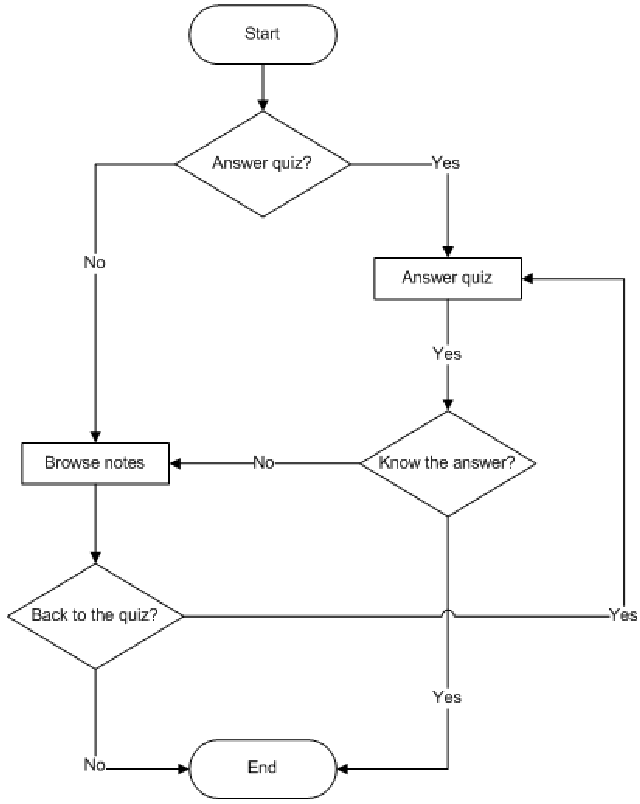

Figure 2. The flow of SCB implementation

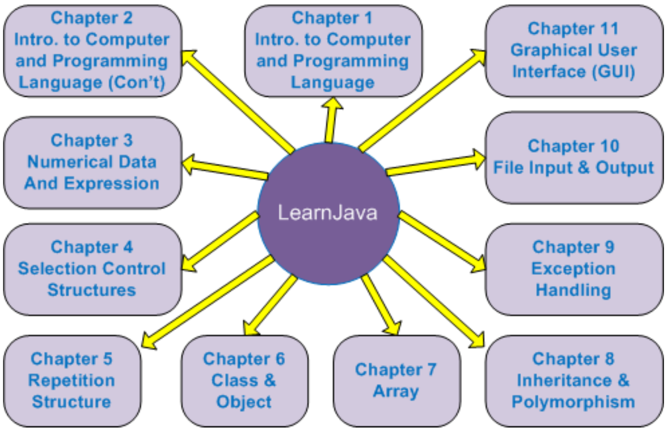

Figure 3. Topics covered in LearnJava

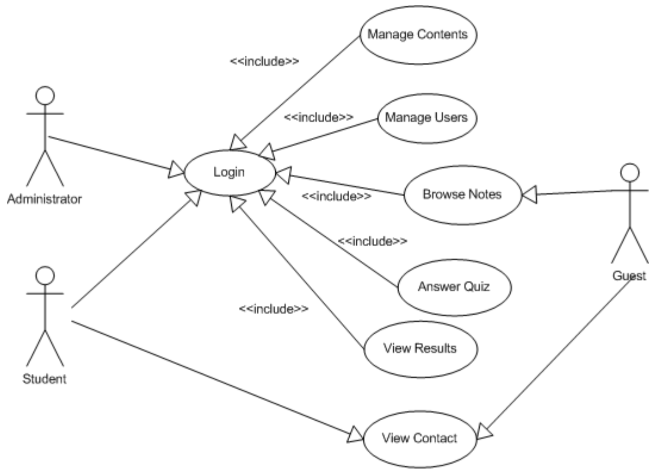

Figure 4. The use case diagram for LearnJava application

LearnJava was developed in Hypertext PreProcessing (PHP) scripting language. It runs on Apache web server with MySQL database for storing learning contents and users' information. LearnJava comprises of a menu such as Contact, Sign-In, Admin, Notes, Quiz, My Results and Home. The users of the system are administrator, students, and guests. Figure 4 shows the use case diagram for the system. The hierarchy of LearnJava application is visualized in a web-site map diagram in Figure 5.

LearnJava is available online through http://sotlsoc.net. Figure 6 shows the main interface of LearnJava. Once students are successfully authenticated, they will be presented with the horizontal menu as in Figure 7. The screenshot in Figure 8 shows the example of an interface for answering the quiz. When students browse and learning the learning contents, they are presented with the interface in Figure 9. When students completed answering the quiz of a particular chapter, they can view the results (i.e., report card) as the screenshot in Figure 10.

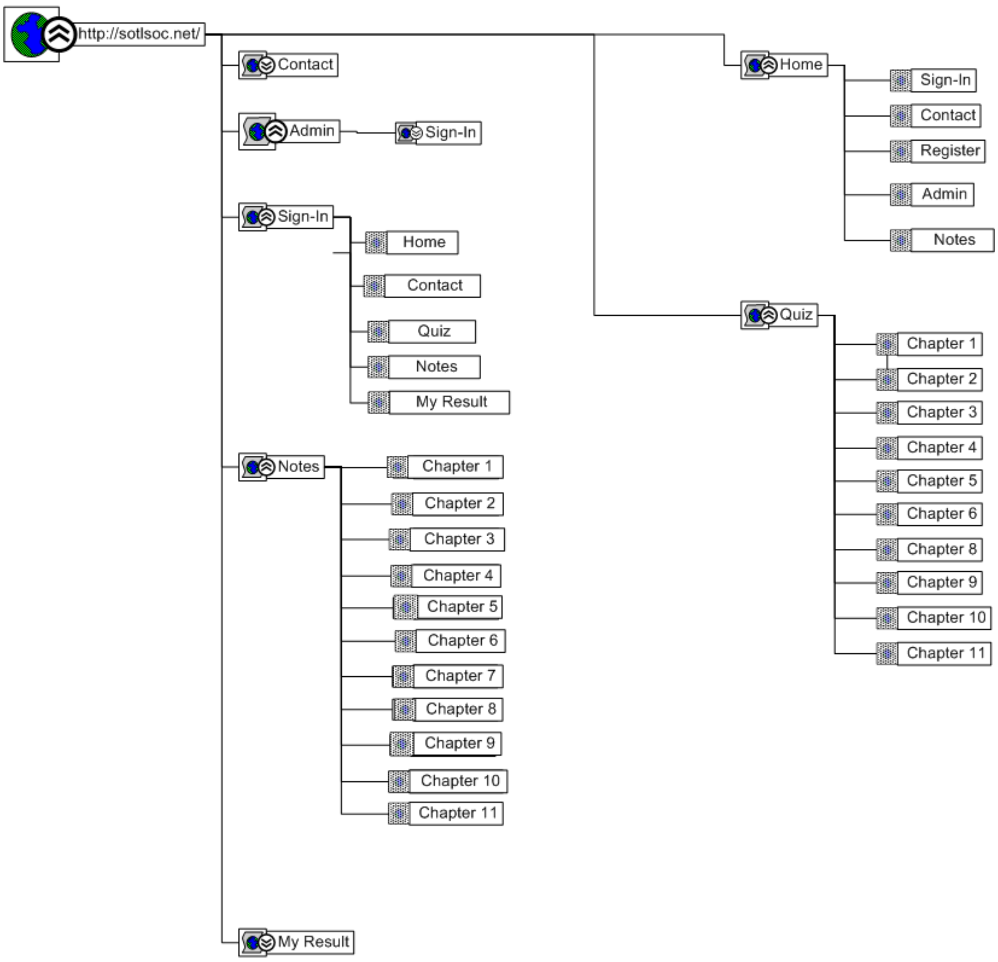

Figure 5. The hierarchy of LearnJava application

Figure 6. The main interface of LearnJava

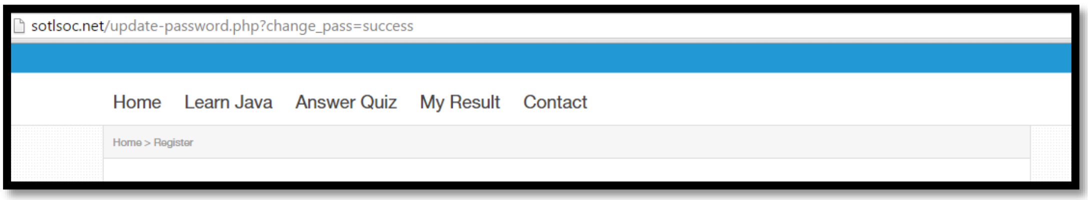

Figure 7. The students' menu in LearnJava

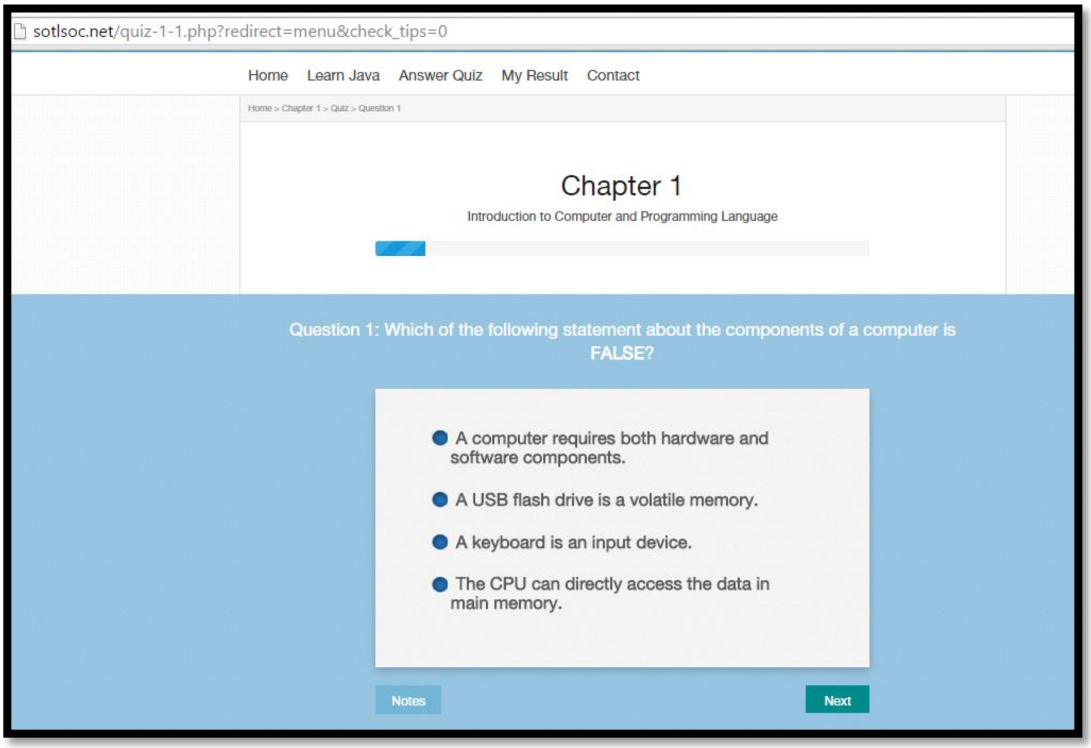

Figure 8. The interface for answering the quiz

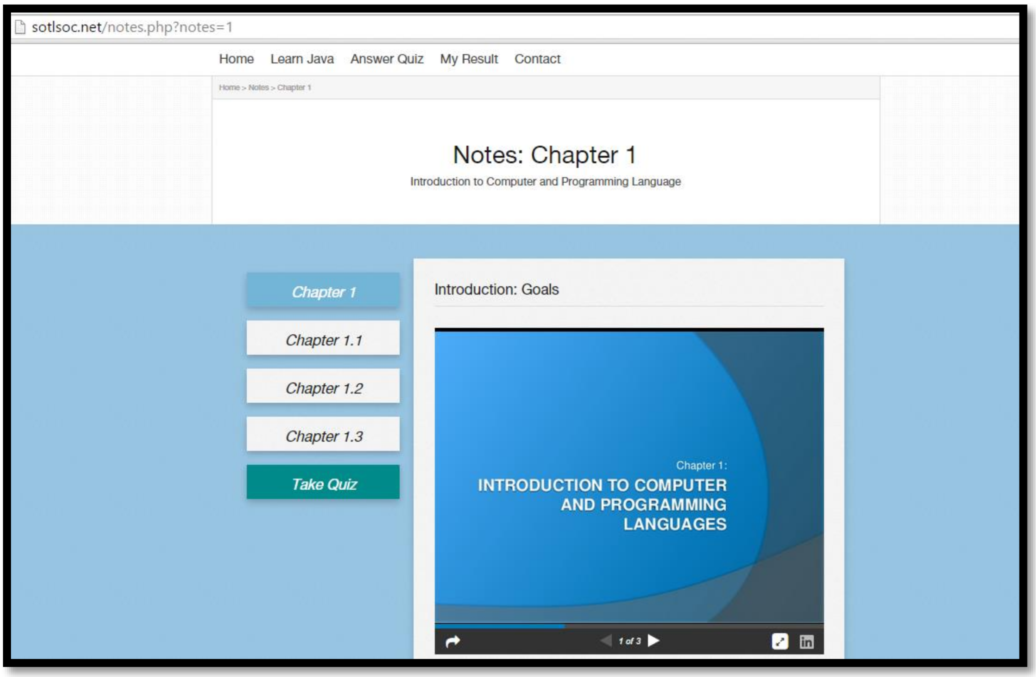

Figure 9. The interface for browsing notes

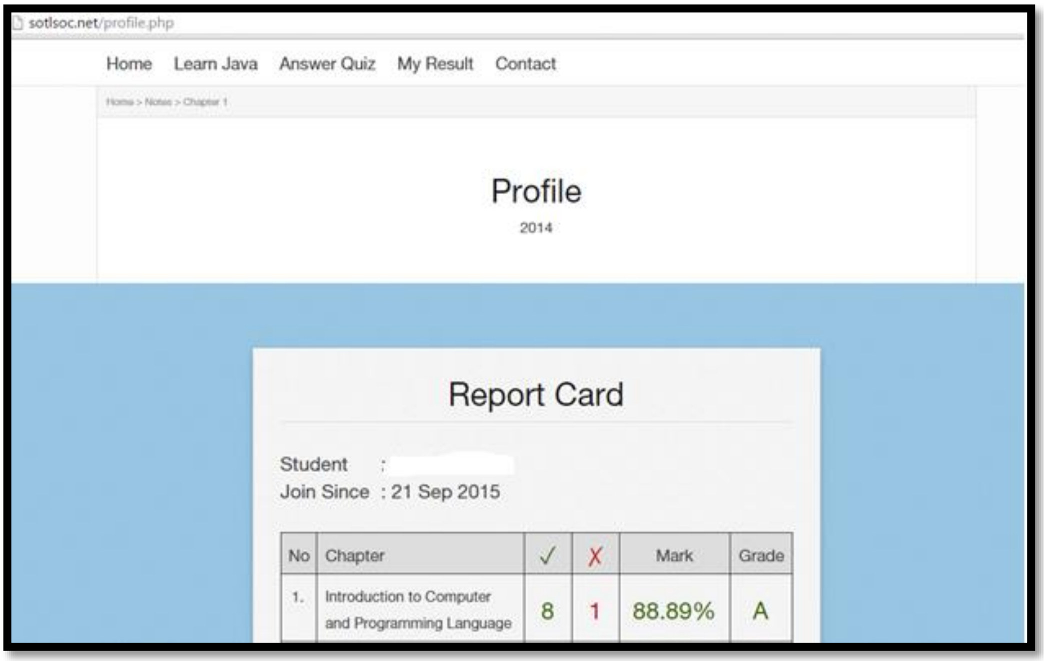

Figure 10. The interface for student's report card (My Result)

IV. Methodology

A. Method

A laboratory experimental study was conducted in April 2015 to measure the effect of SCB on students' engagement when they use the web-based learning application for learning Java programming. The research intends to explore how students perceived engagement when they use LearnJava with SCB technique. Specifically, the hypothesis for the study is "*Students were engaged in the computer-based learning process when they used LearnJava with SCB*".

B. Participants

The participants of the study were recruited among students who enrolled in STIA1014 Introductory to Programming which comprised of forty-nine students from two groups. They were first-year students taking IT, Multimedia, and Business Mathematics programs. Ninety percent of them took IT program. Among them were 27 females and 22 males. The average age of the participants was 20.83 years.

C. Materials

The materials used for the study comprised of the LearnJava tool (i.e., the tool that was explained in the previous section), a pre-test question, a post-test question, and a self-administered questionnaire for measuring engagement. The students were given a lecture on Chapter 1 of the course syllabus two days prior participating in the study by the same instructor. They used LearnJava as a complement to support the classroom teaching and learning.

The pre-test and post-test question contained the same questions. The students were given a thirteen-line of simple Java code that had six syntax errors. The students must spot the errors and write the correct code. They were also asked to write the output of the code which consisted of “*println*” command of simple variable declarations. The purpose of these pre-test and post-test questions was to measure the students’ prior and post knowledge and to confirm whether learning process had happened when the students used the learning tool. The pre-test and post-test questions were in a paper-based format.

The other paper-based instrument was the self-administered questionnaire. It was adapted from Webster et al. [29] and Park et al. [30]. It contained twelve items that used a five-point Likert scale with one represented “strongly disagree” and five for “strongly agree”. The twelve questions were actually measuring engagement from four dimensions that are control, attention focus, curiosity, and intrinsic interest. The questions are as in Table 1.

Dimensions	**Engagement Perspectives**
Control	When using LearnJava, I felt in control over everything
	I felt that I had no control over my learning process with LearnJava
	LearnJava allowed me to control the whole learning process
Attention focus	When using LearnJava, I thought about other things
	When using LearnJava, I was aware of distractions
	Using LearnJava, I was totally absorbed in what I was doing
Curiosity	Using LearnJava excited my curiosity
	Interacting with LearnJava made me curious
	Using LearnJava aroused my imagination
Intrinsic interests	Using LearnJava bored me
	Using LearnJava was intrinsically interesting
	LearnJava was fun for me to use

Table 1. The questionnaire for measuring engagement

D. Procedure

The experiment was conducted in two sessions. The first session was conducted in the morning while the other session was on the afternoon of the same day. The first session was attended by twenty-eight students, and the rest of the students attended the afternoon session. Both sessions was conducted in a computer laboratory equipped with desktop computers with wired Internet connection. The students were given light refreshment at the end of the experiment. Each session was conducted in two hours. The order of the tasks during the experiment is as below:

1) Students sit individually with a computer connected to the Internet. They were not allowed to communicate with each other.
2) The facilitator explained the purpose and procedure
3) Students read the information sheet
4) Students signed the consent form
5) Students answered the pre-test
6) Students interacted with the system. Firstly, they answered Chapter 1 quiz, then, they browsed the relevant notes for the chapter.
7) Students answered the post-test
8) Students answered the post-task self-administered questionnaire

Each student used their student identification number to match the pre-test, post-test, self-administered questionnaire and their credentials in LearnJava system. The experiment was recorded using three video cameras that were located in three different locations in the laboratory. The purpose of recording the sessions was to analyse students reactions during the interaction with the learning tool. However, this is beyond the scope of this paper. The participants’ interactions with LearnJava were also logged by the system automatically. However, the data will not be analysed and presented here.

V. Results

This section explains the results derived from the experimental study as explained in the previous section. The section will first describe the data analysis procedure, followed by the students’ background, the analysis of pre-test and post-test, as well as the students’ engagement in computer-based learning.

A. Data preparation and analysis

The students’ answer on the paper-based pre-test and post-test were checked and marked. The results for their pre-test and post-test were matched according to the student identification number. Then, the students’ response to the self-administered questionnaire was recorded in Microsoft Excel sheets and matched with the pre-test and post-test. The data were checked for consistency and missing values. The Cronbach’s Alpha coefficient for the twelve items of the questionnaire is 0.632 suggesting that the data have an acceptable level of internal consistency. The data were analysed using non-parametric statistical tests such as Wilcoxon-Singed Rank and Chi-square tests due to the small sample size.

B. The students’ background

Analysis on the students’ background shows that 63% of the students had been using computers more than three years, while 22% had experience of using the computer less than two years. The rest had an experience of using the computer within two to three years. In terms of students’ experience with computer-based learning, 45% of them had used e-learning systems beforehand. The other 45% of the students never had an experience with e-learning systems. The other 10% of the students were not sure whether they had used it before.

C. Pre-test and post-test results

Firstly, we analyse the students’ scores for their pre-test and post-test. The purpose of this analysis is to identify whether learning happened when the students interacted with LearnJava. In this study, we assume that learning process within the computer-based learning environment happened if there is a difference in the average of the pre-test and post-test scores.

Analysis of the pre-test and post-test results shows that on average, the students scored higher in their post-test. The mean score for the pre-test was 4.1633 (out of 10), and the post-test was 4.8367. Table 2 shows the means, standard deviations, the minimum and the maximum scores of the pre-test and post-test. The Wilcoxon-Signed Rank test on the pre-test and post-test scores suggested that the difference was statistically significant ($Z=-3.146$, $p=0.002$). The median was

4 and 5 for the pre-test and post-test respectively. The test statistics is presented in Table 3.

	Means	Standard Deviations	Minimum Score (0)	Maximum Score (10)
Pre-test	4.1633	2.50289	0	9
Post-test	4.8367	2.45244	0	9

Table 2. The means and standard deviations for the pre-test and post-test (N=49)

Further observation on the data shows that almost fifty percent of the students achieved a higher score in the post-test compared to their pre-test. The maximum difference in the score of the tests reached up to five marks. This can be concluded that learning process happened for the students within the computer-based environment. Basically, it was a retention process of what they have learnt in the classroom.

D. Students' engagement in computer-based learning

The heart of the study is to measure students' engagement when they use LearnJava with SCB for learning programming. The analysis of the data was conducted using Chi-square tests. Therefore, the students' response to the self-administered questionnaire on the five-point Likert scale was classified into three nominal groups; "Agree", "Disagree", and "Neutral". "Strongly Agree" was classified as Agree and "Strongly Disagree" was classified as "Disagree". Depending on the polarity of the questions (i.e., positive or negative), we only selected either "Agree" or "Disagree" to represent the students' engagement.

The frequency of the nominal groups (i.e., agree, disagree, and neutral) for each of the twelve questions was counted, and its percentage was calculated. Then the Chi-square test was applied to see whether there is any significant difference in terms of students' engagement in learning with LearnJava. The analysis of the students' engagement is presented based on the four dimensions of engagement (i.e., control, attention focus, curiosity, and intrinsic interests). Table 4 shows the test statistics for the twelve items. Ten out of the twelve questions were statistically different and marked in asterisk (*).

In terms of control, 71% of the students agreed that they were in control of everything when the used LearnJava. 69% of the students agreed that LearnJava allowed them to control their learning process. Looking at the students' attention focus, 67% of the students claimed that they were totally absorbed with answering the quiz in LearnJava. However, only 49% of the students were unaware of other distractions while in the learning process with LearnJava. Analysis on the students' curiosity shows that approximately 60% agreed that LearnJava increased their curiosity. However, LearnJava was intrinsically interesting to help the students in learning their programming. More than 70% of the students agreed about this. Figure 11 shows a bar graph representing the results.

Generally, the students agreed that LearnJava helped them to increase their curiosity in learning Java programming language. Further, LearnJava was also intrinsically interesting to help them engage in the computer-based learning activities. However, there were mix results on the control and attention focus that LearnJava can contribute towards students' engagement in computer-based learning. Some items in these two dimensions were not significantly different. Therefore, the results suggested that students' engagement in LearnJava was mainly attributed by curiosity and intrinsic interests.

VI. Discussion and Conclusion

In this section, we revisit our objective and hypothesis for the research. The research intended to promote SCB technique that aims to improve students' engagement when they use computer-based learning, especially for Java programming course. The research also investigated whether the technique could help the student to engage in the computer-based learning activity. The hypothesis for the research was "s*tudents were engaged in the computer-based learning process when they used LearnJava with SCB".*

The statistical tests conducted on the data suggested that learning process happened when the students used LearnJava. This is proven by the higher average score in post-test achieved by the students compared to their pre-test. In general, students participated in the computer-based learning activity and learnt from the given materials and activities.

In terms of engagement in computer-based learning, the SCB technique in LearnJava was able to regulate the students' curiosity and intrinsic interests. However, mix results were found in control and attention focus, where a few items were not statistically significant. There could be a reason to explain the results. In terms of control, the scenario can be explained by the prior knowledge principle [31, 32] and expertise reversal effect [33, 34]. Since the activities in the learning tool were programmed sequentially, the students have limited capability to control the sequence and path of the learning. This has made the activities presented to the students in a linear form. Past research has suggested that students with prior knowledge and skills in using computer prefer to have non-linear navigation or path [31, 35]. As the students had learnt the topic beforehand and they had knowledge and skills in using computer and web browser, this could be the reason why the SCB technique was unable to give the students control over their learning activity.

The students' attention focus could be improved by having a one-to-one session in the lab, although it could be resource consuming. Another alternative is to allow the students to run the task within their convenience time. As we had the experiment in a computer laboratory where students sit next to each other in rows, the students could be distracted by the video camera that recorded their actions and the other friends' actions who asked for help from the facilitator. In summarizing the study, the overall results showed that ten out of twelve items of engagement were statistically significant. This can be used to represent the students' engagement; hence, the hypothesis was supported.

In a nutshell, the SCB technique promoted in this study was intended to help students engage in learning Java programming through computer-based learning tool. The basic idea of SCB is to allow students to have a flexible adjustment of the level of difficulties so that they engage in the computer-based learning activities. The technique was demonstrated in a learning tool named LearnJava, and the students' engagement was measured by self-administered questionnaire. To measure the learning process, a pre-test and post-test were given to the students prior and after their interactions with the tool. The results suggested that students engaged in the learning activities when they used LearnJava with SCB. Specifically, the dimensions of engagement that contributed to this result were curiosity and intrinsic interests.

Pre-test	Post-test	Pre-test – Post-test					Test Statistics
Median	Median	Negative Rank		Positive Rank		Tie	
(Md)	(Md)	N	Mean Rank	N	Mean Rank		
4	5	6	13.75	24	15.94	19	Z=-3.146, p=.002

Table 3. The results of Wilcoxon-Signed Rank test for the pre-test and post-test

Engagement Perspectives	Agree		Disagree		Neutral		Test Statistics (df =2)	
	Fre-quency	%	Fre-quency	%	Fre-quency	%	Chi-Square (X^2)	p value
When using LearnJava, I felt in control over everything	35	71.4	1	2	13	26.5	36.408	0.000*
I felt that I had no control over my learning process with LearnJava	20	40.8	16	32.7	13	26.5	1.510	0.470
LearnJava allowed me to control the whole learning process	34	69.4	4	8.2	11	22.4	30.163	0.000*
When using LearnJava, I thought about other things	23	46.9	11	22.4	15	30.6	4.571	0.102
When using LearnJava, I was aware of distractions	24	49	6	12.2	19	38.8	10.571	0.005*
Using LearnJava, I was totally absorbed in what I was doing	33	67.3	4	8.2	12	24.5	27.469	0.000*
Using LearnJava excited my curiosity	33	67.3	4	8.2	12	24.5	27.469	0.000*
Interacting with LearnJava made me curious	28	57.1	7	14.3	14	28.6	14.000	0.001*
Using LearnJava aroused my imagination	27	55.1	6	12.2	16	32.7	13.510	0.001*
Using LearnJava bored me	6	12.2	36	73.5	7	14.3	35.551	0.000*
Using LearnJava was intrinsically interesting	36	73.5	4	8.2	9	18.4	36.286	0.000*
LearnJava was fun for me to use	41	83.7	1	2	7	14.3	56.980	0.000*

**** Statistically significant, p<0.005***

Table 4. The results for frequency and percentage of engagement items

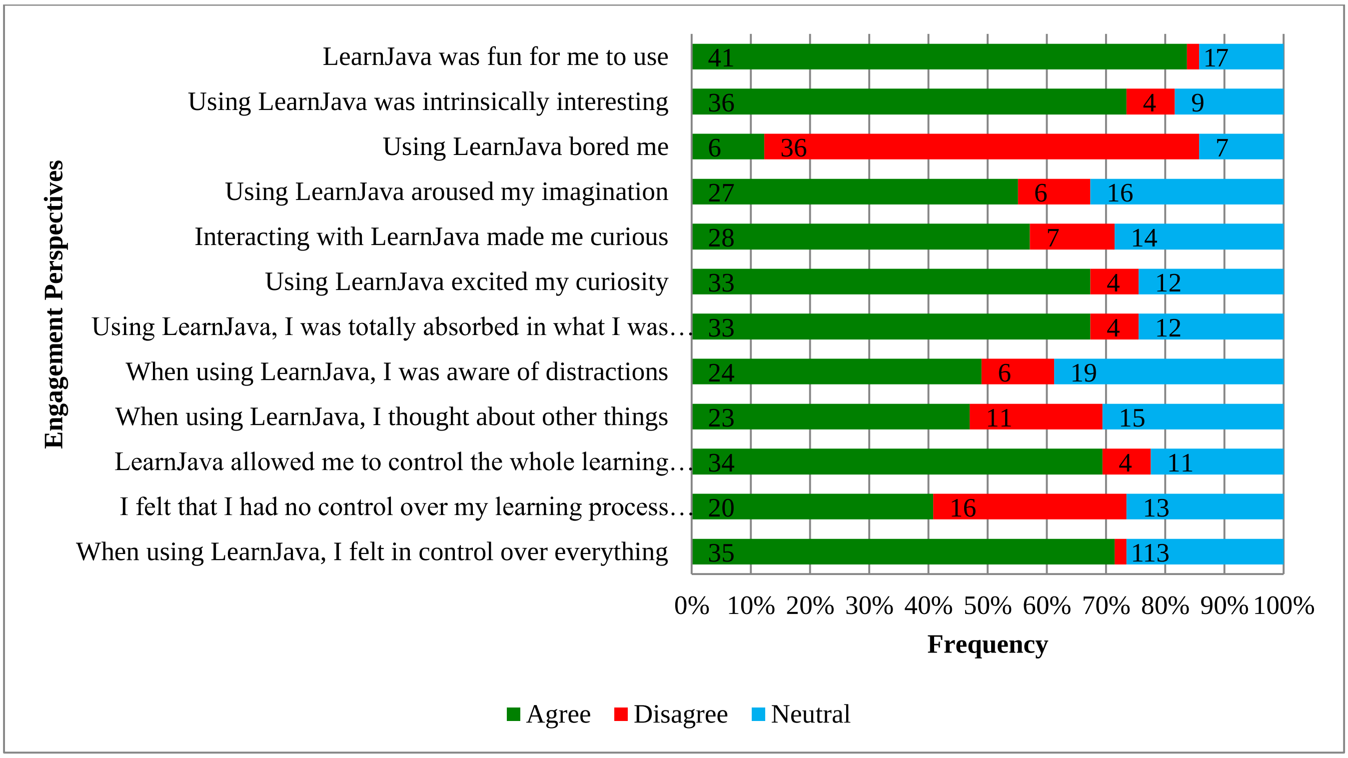

Figure 11. The results for frequency and percentage of engagement items

The findings of this study contribute to the area of computer-based instructional design for designers to consider students' engagement and experience within the learning tools. Further, the SCB technique can be improved by considering issues such as linear and non-linear navigations. Other subject domains could also be explored to see whether SCB would have a similar effect as in learning Java.

Acknowledgment

This research was funded by University Teaching and Learning Center, Universiti Utara Malaysia under the Scholarship of Teaching & Learning (SoTL) scheme (S/O Code: 13008).

References

[1] N. Fricke, C. Lilienthal, M. Lippert, S. Roock, and H. Wolf, "Independence day for Java developers," *Systems Implementation 2000*, pp. 86, 2016.

[2] W. Jackson, "The Interactivity of Digital Audio: Programming," in *Digital Audio Editing Fundamentals*: Springer, pp. 119-130, 2015.

[3] A. Arcuri, J. Campos, G. Fraser, E. Daka, J. Dorn, W. Weimer, and R. Abreu, "Unit Test Generation During Software Development: EvoSuite Plugins for Maven, IntelliJ and Jenkins," presented at IEEE International Conference on Software Testing, Verification and Validation (ICST), 2016.

[4] S. Khan, H. Ali, and S. Saleem, "Research On Auto Mobile-Pc Upload Images Application Through Bluetooth Using Java," *VAWKUM Transactions on Computer Sciences*, vol. 8, pp. 1-6, 2015.

[5] J. E. Bakkali, T. E. Bardouni, S. Safavi, H. Boukhal, M. Kaddour, K. Benaalilou, and E. Chham, "ERSN-OpenMC, a Java-based GUI for OpenMC Monte Carlo code," *Journal of Radiation Research and Applied Sciences*, 2015.

[6] N. Wittayasooporn, A. Panichevaluk, and Y. Zhao, "PicoGrid: A Web-Based Distributed Computing Framework for Heterogeneous Networks Using Java," *Engineering Journal*, vol. 19, pp. 153-165, 2015.

[7] S. Hadjerrouit, "A constructivist approach to object-oriented design and programming," presented at ACM SIGCSE Bulletin, 1999.

[8] L. Tomas, M. Lasen, E. Field, and K. Skamp, "Promoting Online Students' Engagement and Learning in Science and Sustainability Preservice Teacher Education," *Australian Journal of Teacher Education*, vol. 40, pp. 5, 2015.

[9] D. Clark, "Psychological myths in e-learning," *Medical teacher*, vol. 24, pp. 598-604, 2002.

[10] J. Gilbert, S. Morton, and J. Rowley, "e-Learning: The student experience," *British Journal of Educational Technology*, vol. 38, pp. 560-573, 2007.

[11] F. Deepwell and S. Malik, "On campus, but out of class: an investigation into students' experiences of learning technologies in their self-directed study," *Research in learning technology*, vol. 16, 2008.

[12] M. Paechter, B. Maier, and D. Macher, "Students's expectations of, and experiences in e-learning: Their relation to learning achievements and course satisfaction," *Computers & Education*, vol. 54, pp. 222-229, 2010.

[13] P. Sharafi, L. Hedman, and H. Montgomery, "Using information technology: engagement modes, flow experience, and personality orientations," *Computers in Human Behavior*, vol. 22, pp. 899-916, 2006.

[14] H. L. O'Brien and E. G. Toms, "What is user engagement? A conceptual framework for defining user engagement with technology," *Journal of the American Society for Information Science and Technology*, vol. 59, pp. 938-955, 2008.

[15] C. O. Walker, B. A. Greene, and R. A. Mansell, "Identification with academics, intrinsic/extrinsic motivation, and self-efficacy as predictors of cognitive engagement," *Learning and Individual Differences*, vol. 16, pp. 1-12, 2006.

[16] B. A. Marinak and L. B. Gambrell, "Intrinsic motivation and rewards: What sustains young children's engagement with text?," *Literacy Research and Instruction*, vol. 47, pp. 9-26, 2008.

[17] N. Katuk, R. Wang, and H. Ryu, "Enhancement of learning experience using skill-challenge balancing approach," in *AI 2011: Advances in Artificial Intelligence*: Springer, pp. 707-716, 2011.

[18] N. Katuk, "Learning experience in dynamic and non-dynamic curriculum sequencing systems." Auckland: Massey University, 2012.

[19] J. Kim, H. Ryu, N. Katuk, R. Wang, and G. Choi, "Enhancing Competence and Autonomy in Computer-Based Instruction Using a Skill-Challenge Balancing Strategy," *Journal of Educational Computing Research*, vol. 50, pp. 1-28, 2014.

[20] M. Csikszentmihalyi, *Finding flow: The psychology of engagement with everyday life*: Basic Books, 1997.

[21] M. Csikszentmihalyi, *Beyond boredom and anxiety.*: Jossey-Bass, 2000.

[22] E. L. Deci and R. M. Ryan, "Self-determination theory," *Handbook of theories of social psychology*, vol. 1, pp. 416-433, 2011.

[23] N. Katuk and H. Ryu, "Finding an optimal learning path in dynamic curriculum sequencing with flow experience," presented at Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on, 2010.

[24] N. Katuk, J. Kim, and H. Ryu, "Experience beyond knowledge: Pragmatic e-learning systems design with learning experience," *Computers in Human Behavior*, vol. 29, pp. 747-758, 2013.

[25] N. Katuk, "Progressive assessment of student engagement with web-based guided learning," *Interactive Technology and Smart Education*, vol. 10, pp. 116-129, 2013.

[26] N. Katuk, M. Omar, and N. Halim, "Improving Engagement in Hypermedia Learning: A Design for Navigation Support," presented at Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services, 2014.

[27] N. Katuk and N. H. Zakaria, "Engagement in Web-Based Learning System: An Investigation of Linear and Nonlinear Navigation," in *Pattern Analysis, Intelligent Security and the Internet of Things*: Springer, pp. 75-83, 2015.

[28] R. Wang, H. Ryu, and N. Katuk, "Assessment of students' cognitive affective states in learning within a computer-based environment: Effects on performance," *Journal of Information & Communication Technology*, vol. 14, pp. 153-176, 2015.
[29] J. Webster, L. K. Trevino, and L. Ryan, "The dimensionality and correlates of flow in human-computer interactions," *Computers in human behavior*, vol. 9, pp. 411-426, 1993.
[30] J. Park, D. Parsons, and H. Ryu, "To flow and not to freeze: Applying flow experience to mobile learning," *Learning Technologies, IEEE Transactions on*, vol. 3, pp. 56-67, 2010.
[31] S. Kalyuga, "Prior knowledge principle in multimedia learning," *The Cambridge handbook of multimedia learning*, pp. 325-337, 2005.
[32] C. E. Baukal and L. J. Ausburn, "Relationship of prior knowledge and working engineers' learning preferences: implications for designing effective instruction," *European Journal of Engineering Education*, pp. 1-21, 2016.
[33] A. Khacharem, B. Zoudji, and S. Kalyuga, "Expertise reversal for different forms of instructional designs in dynamic visual representations," *British Journal of Educational Technology*, vol. 46, pp. 756-767, 2015.
[34] O. Chen, S. Kalyuga, and J. Sweller, "The Expertise Reversal Effect is a Variant of the More General Element Interactivity Effect," *Educational Psychology Review*, pp. 1-13, 2016.
[35] T. J. Mitchell, S. Y. Chen, and R. D. Macredie, "Hypermedia learning and prior knowledge: domain expertise vs. system expertise," *Journal of Computer Assisted Learning*, vol. 21, pp. 53-64, 2005.

Author Biographies

Norliza Katuk obtained her Bachelor's degree in information technology from University Utara Malaysia in 2000 and her Master's degree in computer science from Universiti Teknologi Malaysia in 2002. She obtained her Doctoral degree in information technology from Massey University, New Zealand in 2012. She is a Senior Lecturer at Universiti Utara Malaysia. Her research interests cover multidisciplinary areas including information security and privacy, web-technology, disaster management and human-computer interaction. Dr. Norliza is an editorial board member of Journal of ICT, Universiti Utara Malaysia.

Nur Haryani Zakaria was born in Johor Bahru, Malaysia on 2nd December 1978. She obtained her Bachelor's degree in information technology from University Utara Malaysia in 2000 and her Master's degree in computer science from Universiti Teknologi Malaysia in 2002. She obtained her Doctoral degree in computer science from Newcastle University, United Kingdom in 2013. She started her career as a Tutor at Universiti Utara Malaysia in 2000 and is currently a Senior Lecturer there. Her research interests cover areas related to usable security and privacy, information security and human-computer interaction.

Alawiyah Abd Wahab received her B.Sc. in Business Administration (Management Information Systems) from the University of North Carolina at Charlotte, USA in 1994, M. Sc. in Information Systems from the University of Leeds, UK in 1996 and PhD in Information Systems from Newcastle University, UK in 2016, respectively. Her research interests include e-learning, e-commerce and web-based applications.

Osman Ghazali obtained his Bachelor's degree, Masters and Ph.D. in information technology from Universiti Utara Malaysia in 1994, 1996 and 2008 respectively. He is an academic staff at University Utara Malaysia since 1994. Currently, he is an associate professor at School of Computing. His research interest is internet technologies and applications, which includes computer networking, network security, cloud computing, and mobile and web applications. In addition, as a lecturer he also involves in research related to teaching and learning.