Embedding Adaptation Levels within Intelligent Tutoring Systems for Developing Programming Skills and Improving Learning Efficiency

Mohamed A. Elkot
Department of Computer Science
College of Science and Arts in Alrass, Qassim University, Qassim, Saudi Arabia

***Abstract*—Intelligent Tutoring Systems (ITSs) represent the virtual learning environment that provides learning needs, adapts to the characteristics of learners according to their cognitive and behavioral aspects, to reach desired learning outcomes. The purpose of this study is to investigate the impact of embedding some adaptation levels within intelligent tutoring systems on developing Object-Oriented Programming skills (OOP), as well as on learning efficiency for students of the computer science department, Faculty of Science and Arts at Qassem University. In this context, the author developed an Intelligent Tutoring System (ITS) that provides multiple levels of adaptation (Learner level, links level) to support automatic adaptation to each of the students' characteristics and investigate the effectiveness of the system on dependent variables. The random sample consisted of (n=44) students. Those students were divided into two similar groups, Experimental an (ITS), and Control (face-to-face, traditional). The findings revealed that there was a noticeable improvement in the students' performance for the experimental group than the control group how used face-to-face method for the programming skills and learning efficiency.**

Keywords*—Intelligent Tutoring Systems (ITS); programming skills; adaptive e-learning; learning style; learning efficiency

I. Introduction

Intelligent tutoring systems (ITSs) based on artificial intelligence play a critical role as expert systems that provide an electronic learning environment that meets the needs of the learners and automatically adapts to their cognitive and behavioral characteristics, to achieve the highest educational return that can be reached [1]. (ITSs) will stimulate the learning process, stimulate the learner's motivation, direct his energies, and raise his motivation to continue the learning process [2]. This has been confirmed by many previous research and studies that have indicated the importance of these systems in learning programming languages that require higher cognitive levels in Thinking and problem-solving skills.

Many available current traditional e-learning management systems are intended for all learners, they do not take into account the individual characteristics and differences of learning abilities, experiences, skills, learning style, and the available educational materials through these systems do not allow for learning that takes into account the needs of individual learners as it is called "one-size-fits-all". In contrast, (ITSs) are adaptive learning systems developed to create a dedicated virtual environment to the learner and his individual needs at a suitable time and place, taking into account his abilities and skills. “One-on-one-teaching” [3, 5].

Adaptation is the solution to many of the limitations and weaknesses of traditional e-learning systems [6]. The main purpose of the adaptation process is to enhance learning by meeting the different needs of individual learners, [7] points out those adaptive systems are interactive systems that adapt their behavior according to the pattern and characteristics of each learner for the learning process and develop their skills. [8] think that in light of these characteristics; many previous studies have dealt with (ITSs) in many ways, especially with regard to the important question of how to achieve adaptation based on the learner characteristics? Thus, many studies have developed these systems based on specific characteristics such as learning style, motivation, cognitive traits, knowledge, learners' preferences… etc., and a few studies went to design adaptation process by combining multiple parameters together to build intelligent systems [9], these studies include:

A study [1] provided an introduction to integrating some learning styles into adaptive e-learning system based on hypermedia to measure the impact of that system at the learners' level. The results showed a clear improvement in learners' performance of the course using JavaScript.

In [4] study, which was concerned with the development of an adaptive e-learning system, based on the integration of some characteristics of adaptation levels, namely learning style and knowledge of the level of learners. To teach object-oriented programming and integrate with the level of presentation to display content by two factors, namely, the sequence and non-sequence in the presentation of content. The statistical results showed a significant improvement in students' learning levels of programming when they were exposed to the targeted adaptive learning system.

These systems have great credit in improving the cognitive and skillful level of learners. Especially in programming courses, as it was cited in previous research and studies [3,10] which led the author to think about developing an (ITS) based on the integration of some parameters (link generation, learning style), which are presented through different levels of adaptation (Learner level, Link level), which contribute to increase the amount of interaction with learners and identify the deficiencies in building their knowledge, skills, and rebuild them and determine the performance expectations, especially with object-oriented programming courses that represent difficulty in understanding and implementation, in addition, those courses represent the most challenge for students of the department of computer science [10,11, 12, 13] as learners need greater interaction with them in each step of problem-solving, writing a program, and training in problem-based thinking, so the aim of this study is to answer the following questions:

The main question "What is the impact of embedding adaptation levels within intelligent tutoring systems for developing programming skills and improving learning efficiency?"

1) What is the effect of integrating learning style and link generation with the proposed an (ITS) on the development of programming skills for students of computer science?

2) What is the effect of integrating learning style and link generation with the proposed an (ITS) on improving the learning efficiency for students of computer science?

II. Review of the Related Literature

A. Adaptation Levels within Intelligent Tutoring Systems

There are many levels of adaptation underpinning the building (ITSs), and the names of those levels differed from one study to another. In this study, the following names will be accredited for the following levels:

The first level of adaptation levels, which is Learner Level, which includes many characteristics and parameters like (learning style, motivation, cognitive traits, knowledge, learners' preferences, etc.), where this level represents the biggest challenge and the basis on which the system is built. This level is closely related to the next four other levels as indicated by [14]. The second level is the navigation or link level, which means the process of linking the learning objects (LOs) of the content and sequencing to facilitate the learner interaction with the system. This level includes many elements such as (Direct guidance, link hiding, and link generation)[15]. The third level is the content level, which involves the design of instructional content. It depends on the design and careful evaluation of (LOs) that can be reused and recalled with more than one learner at the same time depending on the type of characteristics of each individual learner. The fourth level is the presentation level. It is the way of displaying the content of each screen, as it is automatically generated to suit the nature of the learner individually, and it differs from a screen to another according to learners' classification. Finally, the level of learning path, which classifies learners into paths according to the characteristics of each learner to facilitate the recall and generation of learning elements appropriate for each learner individually and the following Fig. 1 shows adaptation levels in (ITSs).

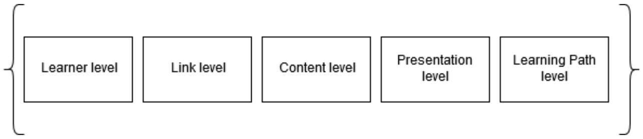

Fig. 1. Shows Levels of Adaptation in (ITSs).

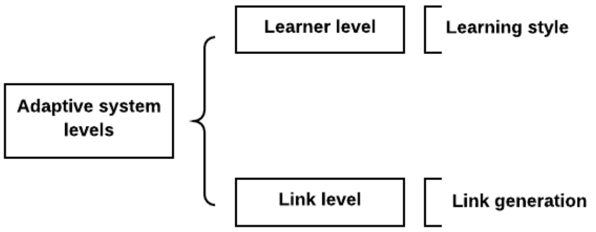

Fig. 2. Shows the Target Adaptation Levels in the Current Study.

The previous adaptation levels are very important in the design and construction of (ITSs) [4, 14]. These levels are linked together and overlapping, especially the link level, which the author believes the importance of this level in guiding the learner during the learning process within the system, specifically in terms of how content elements are sequenced, hidden and show links at the right time for each individual learner. Therefore the current study will be limited to merging two parameters from two different levels, the Learner Level (Learning style) with the Link level (link generation) as it is illustrated in Fig. 2.

B. Learning Objects (LOs) and Repositories

Learning objects are customizable and reusable interactive digital units in different contexts [15]. (ITSs) depend on Learning objects (LOs). To create learning content appropriate to the needs of learners, as well as, individual characteristics of students and their different capabilities [13], as the primary purpose is to reused with more than one learner to reduce the cost and speed of recall without loss of information. These objects are characterized by reusability, accessibility, scalability, manageability, durability, interoperability and manageability[16]. Whereas from a technological perspective, it is based on an idea of object-oriented programming, which we could use it many times in multi situations. This feature allows the idea of creating new and unique learning objects that can be used multiple times and there are several criteria that determine the structure of learning objects of the most important is SCORM standard ”Shareable Content Object Reference Model” [17]. The design of these (LOs) is based on the idea of Metadata generation and reusability [15]. These objects require storage in a database called digital repositories, which act as a platform that store, retrieve, collect, and publish learning objects. Consequently, the current study depends on designing learning objects for intelligent tutoring systems by SCORM standards.

C. (ITSs) and Programming Skills

In fact, learning the programming languages, especially object-oriented programming (OOP) is one of the areas that are difficult to understand and implement. Besides, it represents the most difficult challenge for students [18, 19]. Because learners need more interaction with each step of problem-solving, writing programs, and the learner's need for help, ongoing mentoring, and problem-based thinking training. In contrast, (ITSs), which have been proven by many previous studies and research to be effective in teaching these courses, because they are adapted to the individual needs of the learner for each student individually according to their abilities, skills, and characteristics. In this regard, there are several relevant studies had adopted (ITSs) to learn computer courses including:

In [20] developed an adaptive e-learning system (AeLS) to learn computer graphics course for undergraduate students of a computer application program, they depend on the Felder Silverman learning style to build their system. Consequently, they divided learners into two groups (slow learners, face to face). Findings revealed that there is an obvious improvement with the slow learners group. In [19] developed an adaptive intelligent tutoring system to explore how adaptability is achieved in conventional intelligent tutoring systems and proposes an adaptive educational model that uses machine-learning techniques to define effective teaching methods appropriate for a particular student. The findings of the proposed model's test demonstrate the ability of the model to accurately identify the student subjects according to their learning style. [21] aimed to design an educational environment that supports smart classrooms, which included three aspects of content, collaboration, and hyperlinks. This proposal intended to obtain the data that helps in determining the appropriate learning materials for each learner according to his learning needs in programming languages subject, which includes the learning objects, knowledge, the skills of the students, as well as the educational activities. This proposal based on a system for managing learning objects (LOs) through a smart learning environment that meets learners' needs. Author in [22] provided an approach to factors influencing on designing and developing adaptive learning systems that include individual and group learning. This approach involves reinforcing and supporting learning to build intelligent environments based on artificial intelligence. Not only to provide appropriate methods that suggest learning materials that suit the needs of learners but to provide a good methodology for tracking student variables and status during the learning process, the finding indicated that the feasibility and effectiveness of the proposed approach. [23] aimed to provide an approach to design an adaptive learning system that not only depends on the learning style but also depends on the learning preferences. The experimental was applied for two groups. The firstly is the control group, that uses the traditional methods and the experimental group, which is used as a combination of variables within the adaptive learning environment, which provide learning according to learners' needs. The results showed the positive effect of the conducted of the system based on the integration of the third variables compared to the traditional system. Author in [24] investigated designing and developing adaptive courses used for learning programming basics which includes many kinds of personalized agents such as collect data from the user and tracking the response from students to present students appropriate materials of learning according to their learning styles. Author in [10] used intelligent tutoring system for learning java programming under multiple learning different methodologies such as problem-solving, knowledge space and overlays, they implemented the integration of cognitive and affective data to display emotions and empathy of the students, they used (ITS) open-source code in a programming language.

III. Method

For the current study, participants were (n=44) students. They were male students studying in the department of computer science at Qassim University, KSA. They split up randomly into two groups. The experimental group studied through the proposed system (ITS) and face-to-face group. There were (22) students in each group.

A. *Instrument*

The author developed the student's programming skills test in the Python Object-Oriented Programming course. The test consists of (15) questions in the inheritance unit. For the learning efficiency scale. The author relied on (SEL) scale for [25], which included (27) items to measuring self-efficacy for learning. While students' attitude towards an (ITS) was appeared by conducting personal interviews with students to know their views about the system, accessibility, and usability.

B. *Limitations of the Study*

The current study adopted the following limitations:

- Spatial Limitations: Faculty of Sciences and Arts, Qassim University, K.S.A.
- Python object-oriented programming language course (OOP).
- Human Limitations: Students in Level 8 and 9 in the Dep't. Of computer sciences.

C. *Design and Procedures*

There are many frameworks that can be used in designing adaptation systems such as, Generic Adaptation Framework (GEF) [14], Munich model [26], Adaptive Hypermedia Application model (AHAM) [27]. All these frameworks in general, should answer three questions: What can we adapt? to what do we adapt? In addition, how can we adapt? Accordingly, the current study adopted three components to answer the above questions, which used to design the proposed system (adaptation model, domain model, and learner model) as it is illustrated in Fig. 3. These components, in particular, constituted the common denominator for all previous frameworks [5].

1) Learner model: The learner model answers the question- What can we adapt? Accordingly, in this model, the adaptation system is based on learning style for each student, which refers to recognize the best way the learner perceives and interacts with the learning environment as well as, the individual's preferred method of learning [28]. In addition, knowing the needs of the learners and identifying their learning styles is essential in the design of content materials. [29]. Therefore, the present study relied on Felder- Silverman Learning style model (FSLSM). Which is one of the best models that presented many learning styles for learners, it includes four dimensions [Active: Reflective, Sensing: Intuitive, Visual: verbal, Sequential: Global], (FSLSM) considered more flexible than other most models. This model consists of 44 questions divided into four parts, each part involved a specific scope to reveal the learning style of each student, these questions are presenting after the login screen to the system. This domain also contains all the information and data for each individual student that is stored in the database by (Id).

2) Domain model: The domain model answers the question- To what do we adapt? It represents the educational content intended to be presented to the students through this system, this course is object-oriented programming using Python, where this content is presented as learning objects (LOs). These objects are stored classified in the system database (repository) according to the learning style for each individual learner.

3) Adaptation model: In view previous two models, this model answers the question- How can we adapt? Accordingly, an (ITS) relying on the two previous domains to build the process of adaptation, which is done in two ways. First is to determining the learning style of each learner to provide adaptation method based on the learning style, and then, call the learning object of each learner, as well as, all links related with advanced steps are hidden and presented in a timely manner, which depends on the interaction user with content. The second way is to monitor the screens to calculate the time taken. To provide feedback and instructions for each learner, in addition, providing feedback after the assessment and answer questions.

D. System Descriptions

A student is logged in by creating a new account via registration form to collect student's data, which is automatically stored in the database to take a unique ID number, then the student moves to the screen of the test to determine the learning style, which is consisted of 44 questions. Hence, saving the learning style code on the database and then the student moves to the educational content screen, which displays an educational module in a programming language with python shown to each student according to their learning style, and at the same time is tracking learner's behavior in terms of time spent and clicks on links. As well as, providing feedback, guidance, and counseling for each learner individually as shown in Fig. 4.

E. Experiment of the Study

The students were divided into two groups, each group twenty-two students (n=22). The pre-test was applied to the face-to-face control group as well as, the experimental group, which was implemented using an (ITS). The pre-test involved (15) questions about the inheritance unit within the python programming language. As well as, (SEL) scale to measuring self-efficacy, which includes (27) items. The experimental group was taught using an (ITS), whereas the control group learned face-to-face. After that, both of them exposed to the post-test. Fig. 4 shows the activity diagram for an (ITS).

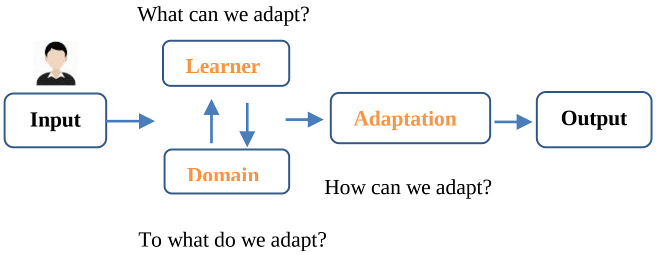

Fig. 3. Framework Architecture for Designing an (ITS).

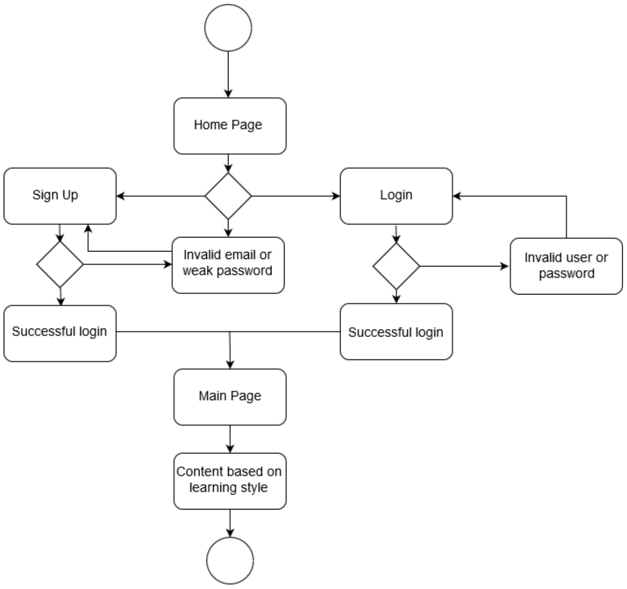

Fig. 4. Activity Diagram for an (ITS).

IV. Result and Discussion

To answer the study questions, the author used a descriptive statistical analysis program (SPSS). The first question asked if there was an effect of integrating learning style and link generation with the proposed an (ITS) on the development of programming skills for students. To check if there are initial differences between the pre-test and post-test for every single group. Paired-Sample T-test was used. In contrast, to comparing the two groups in the post-test, the Independent - Samples T-test was used.

Accordingly to Table I, there are significant differences between the mean scores of the pre-test (mean = 34.09) and the mean scores of the post-test (mean = 40.18) for experimental group, and T = 5.337 at a mean level (p = .000) less than the value of α = 0.05 in favor of the post-test. This indicates that there is an improvement in students' programming skills after they exposed to the technological variable. Therefore, the obvious explanation for this result is that integrating learning style and link generation within an (ITS), have an effect on the performance of students. In addition, this result is consistent with many studies, which used an (ITS) for learning programming languages, such as [4, 10, 21, 23].

In contrast, the results in the previous table of the control group. Also indicates that there are significant differences between the mean scores of the pre-test (mean = 36.14) and the mean scores of the post-test (mean = 36.91), and T = 1.436 at a mean level (p = .166) less than the value of α = 0.05 in favor of the post-test. Despite there is a statistically significant difference, we note that the difference between the mean scores of the pre-test and post-test is very small, which indicates a weak improvement. The reason behind this is that the teaching of programming in the traditional system face-to-face, with the presence of large numbers of students, reduces the chance of student interaction with the professor, in addition to the traditional learning does not take into account the characteristics and individual needs of each learner regardless of the learning style for each student. This is consistent with previous research and studies, which point out that, such as [4, 6, 7, 30].

Accordingly to Table II, there are significant differences between the mean scores of the post-test (mean = 39.91) and the mean scores of the post-test (mean = 36.77), t-test = (1.964), degrees of freedom df = 42, at a mean level (p = .051) not greater than the value of α = 0.05 in favor of the experimental group. This result indicates that the learning outcomes for developing the programming skills of the experimental group are better than the learning outcomes of the control group. This confirms that an (ITS) used had a positive impact on the level of learning among students.

TABLE. I. DISPLAY THE DIFFERENCES BETWEEN PRE-TEST AND POST-TEST IN BOTH EXPERIMENTAL AND CONTROL GROUPS OF PROGRAMMING SKILLS FOR STUDENTS, USING THE PAIRED-SAMPLE T-TEST

	Control group					
	Mean	***SD***	***df***	***P-value***	***t***	***N***
Pre	36.14	5.276	21	.166	1.436	22
Post	36.91	5.255	21			22
	Experimental group					
	Mean	***SD***	***df***	***P-value***	***t***	***N***
Pre	34.09	6.796	21	.000	5.337	22
Post	40.18	5.086	21			22

TABLE. II. DISPLAY THE DIFFERENCES BETWEEN POST-TESTS FOR EXPERIMENTAL AND CONTROL GROUPS OF THE PROGRAMMING SKILLS, USING THE INDEPENDENT - SAMPLES T-TEST

	Experimental and control groups					
	Mean	***SD***	***df***	***P- value***	***t***	***N***
G1	39.91	5.362	42	.051	1.964	22
G2	36.77	5.228	41.9			22

TABLE. III. SHOWS THE DIFFERENCES BETWEEN PRE-TEST AND POST-TEST IN BOTH EXPERIMENTAL AND CONTROL GROUPS OF THE LEARNING EFFICIENCY FOR STUDENTS, USING THE PAIRED-SAMPLE T-TEST

	Control group					
	Mean	***SD***	***df***	***P- value***	***t***	***N***
Pre	30.50	9.324	21	.008	-2.944	22
Post	32.81	8.699	21			22
	Experimental group					
	Mean	***SD***	***df***	***P- value***	***t***	***N***
Pre	27.86	8.747	21	.000	-10.173	22
Post	40.14	7.692	21			22

To answer the second question: What is the effect of integrating learning style and link generation with the proposed an (ITS) on improving the learning efficiency for students of computer science? The author also used Paired-Sample T-test to check if there are initial differences between the pre-test and post-test for every single group as shown in Table III.

It is clear from Table III, there are significant differences between the mean scores of the pre-test (mean = 27.86) and the mean scores of the post-test (mean = 40.14) for experimental group, and T = 10.173 at a mean level (p = .000) less than the value of α = 0.05 in favor of the post-test. This result indicates that there is an improvement in students' learning efficiency after they exposed to the technological variable. Thus, the explanation for this result is that using an (ITS), has an effect on enhancing learning efficiency for students, this result is consistent with [31]. Besides, the results in the previous table of the control group. Also indicates that there are significant differences between the mean scores of the pre-test (mean = 30.50) and the mean scores of the post-test (mean = 32.81), and T = 2.944 at a mean level (p = .008) less than the value of α = 0.05 in favor of the post-test. Undoubtedly, there is an improvement in self-learning efficiency. However in a small percentage, and this is evident from the difference between the means. In addition, this is due to traditional learning, which does not take into account the characteristics and individual needs of each learner.

Undoubtedly, in Table IV, there are significant differences between the mean scores of the post-test (mean = 32.41) and the mean scores of the post-test (mean = 40.41), t-test = (3.234), degrees of freedom df = 42, at a mean level (p = .002) less than the value of α = 0.05 in favor of the experimental group. The current result indicates that the learning outcomes for enhancing the learning efficiency of the experimental group are better than the learning outcomes of the control group. This confirms that an (ITS) used had a positive impact on the level of learning among students.

Finally, all the students were invited to sharing in interviews. Regarding students' attitude towards an (ITS) was appeared by conducting personal interviews with students to know their views about the system, accessibility, and usability. The questions of the interviews were as the following:

1) What do you think about the merits of using an (ITS) on the development of programming skills*?*

2) What do you think about the merits of using an (ITS) on improving the learning efficiency for students*?*

3) What are the difficulties that you faced during learn from an (ITS)*?*

4) What are your suggestions for improving the system?

After the interviews, there seems to be some agreement among students about the following:

1) Ease of use of the system.

2) The lessons are clear.

3) Ease of code application.

4) Lessons need to increase the number of exercises.

5) Students cannot copy code from lesson pages.

TABLE. IV. DISPLAY THE DIFFERENCES BETWEEN POST-TESTS FOR EXPERIMENTAL AND CONTROL GROUPS OF THE LEARNING EFFICIENCY, USING THE INDEPENDENT - SAMPLES T-TEST

	Experimental and control groups					
	Mean	***SD***	***df***	***P- value***	***t***	***N***
G1	32.41	8.687	42	.002	3.234	22
G2	40.41	7.692	41.3			22

From the aforementioned, it may be deduced that an (ITS) used with the integration of some parameters had a clear impact on the development of programming skills among students, as well as on enhancing the learning self-efficacy among students of computer science.

V. CONCLUSIONS

The current study presented the effect of embedding adaptation levels (learner level and link level) via an intelligent tutoring system on developing programming skills and improving learning efficiency for students of computer science. Findings revealed that an (ITS) facilitated for students in improving their programming skills performance. As well as, there is also a positive effect on enhancing the learning self-efficacy, which is statistically evident from previous findings. As a future study, the current study recommends considering deeply at other parameters within other adaptation levels via intelligent tutoring systems. Specifically the presentation level and learning path level.

ACKNOWLEDGMENT

The author (Mohamed Ali Elkot) gratefully acknowledge Qassim University, represented by the Deanship of Scientific Research, on the material support for this research under the number (5144) during the academic year 1439 H / 2018AD.

REFERENCES

[1] Y. Eltigani, A. Mustafa, and S. M. Sharif, An approach to Adaptive E-learning Hypermedia System based on Learning Styles (AEHS-LS): Implementation and evaluation, vol. 3, no. 1. 2011.

[2] H. Peng, S. Ma, and J. M. Spector, “Personalized Adaptive Learning: An Emerging Pedagogical Approach Enabled by a Smart Learning Environment,” Springer Singapore, 2019, pp. 171–176.

[3] S. Jiménez, R. Juárez-Ramírez, V. H. Castillo, J. José, and T. Armenta, Springer briefs in human computer interaction Affective Feedback in Intelligent Tutoring Systems A Practical Approach..

[4] M. T. Alshammari, “Adaptation based on Learning Style and knowledge Level in E-Learning Systems adaptation based on learnung style and knowledge level in e-learnin systems by School of Computer Science,” 2016.

[5] P. Brusilovsky, “Adaptive hypermedia for education and training,” Adapt. Technol. Train. Educ., pp. 46–66, 2012.

[6] S. Graf et al., “Adaptivity and Personalization in Learning Systems based on Students’ Characteristics and Context,” 1st Int. Symp. smart Learn. Environ., pp. 33–36, 2012.

[7] A. Jameson, “Adaptive Interfaces and Agents,” pp. 105–130, 2009.

[8] S. Gross, B. Mokbel, B. Hammer, and N. Pinkwart, “Learning Feedback in Intelligent Tutoring Systems,” KI - Künstliche Intelligenz, vol. 29, no. 4, pp. 413–418, 2015.

[9] H. M. Truong, “Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities,” Comput. Human Behav., vol. 55, pp. 1185–1193, 2016.

[10] M. L. Barrón-Estrada, R. Zatarain-Cabada, F. G. Hernández, R. O. Bustillos, and C. A. Reyes-García, “An Affective and Cognitive Tutoring System for Learning Programming,” 2015, pp. 171–182.

[11] M. A. Alket, “A Network-Based Peer Evaluation Strategy,” Int. J. Mod. Educ. Comput. Sci., vol. 9, no. 4, pp. 32–42, 2017.

[12] D. Weragama and J. Reye, “The PHP intelligent tutoring system,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7926 LNAI, pp. 583–586, 2013.

[13] K. R. Premlatha and T. V. Geetha, “Learning content design and learner adaptation for adaptive e-learning environment: a survey,” Artif. Intell. Rev., vol. 44, no. 4, pp. 443–465, 2015.

[14] E. Knutov, Generic Adaptation Framework for unifying adaptive web-based systems. 2012.

[15] L. P. A. Yela, A. D. O. Escobar, M. M. M. Anota, and J. M. Castillo, “Design of a mobile learning object repository (ROAM),” 2016 IEEE Int. Eng. Summit, IE-Summit 2016, 2016.

[16] V. Dagiene, D. Gudoniene, and R. Bartkute, “The integrated environment for learning objects design and storing in semantic web,” Int. J. Comput. Commun. Control, vol. 13, no. 1, pp. 39–49, 2018.

[17] P. Kharat, G. Shekhar, and P. Gharpure, “Authoring tool based on learning object standards,” Proc. - 2015 Int. Conf. Commun. Inf. Comput. Technol. ICCICT 2015, pp. 1–4, 2015.

[18] I. T. Machin Torres Ive, “Intelligent Tutor for Programming System Using Multiple Intelligences,” IEEE Lat. Am. Trans., vol. 16, no. 2, pp. 634–638, 2018.

[19] M. Dlamini and W. S. Leung, Enhancing object-oriented programming pedagogy with an adaptive intelligent tutoring system, vol. 963. Springer International Publishing, 2019.

[20] L. Joseph and S. Abraham, “Adaptive e-Learning System for Slow Learners Based on Felder-Silverman Learning Style Model,” vol. 1, pp. 123–135, 2019.

[21] D. Rivero-Albarrán, F. Rivas-Echeverria, L. Guerra, B. Arellano, and S. Arciniegas, “Design of a Recommender System for Intelligent Classrooms Based on Multiagent Systems,” 2018, pp. 973–982.

[22] D. Shawky and A. Badawi, “A Reinforcement Learning-Based Adaptive Learning System,” vol. 4, Springer International Publishing AG, 2018, pp. 221–231.

[23] A. Siddique, Q. S. Durrani, and H. A. Naqvi, “Designing Adaptive E-Learning Environment Using Individual Differences,” Pak. J. Sci., vol. 69, no. 1, pp. 101–110, 2017.

[24] B. Vesin, M. Ivanović, A. Klašnja-Milićević, and Z. Budimac, “Personal Assistance Agent in Programming Tutoring System,” in Smart Innovation, Systems and Technologies, vol. 38, 2015, pp. 441–451.

[25] J. E. Klobas, S. Renzi, and M. L. Nigrelli, “A scale for the measurement of self-efficacy for learning (SEL) at university,” pp. 1–24, 2007.

[26] R. J. Sternberg and J. E. Davidson, Conceptions of giftedness. Cambridge University Press, 2005.

[27] N. Koch and M. Wirsing, “The Munich reference model for adaptive hypermedia applications,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2347 LNCS, no. May, pp. 213–222, 2002.

[28] T. Sheeba and R. Krishnan, “Automatic Detection of Students Learning Style in Learning Management System,” in Smart Technologies and Innovation for a Sustainable Future, Advances in Science, Technology & Innovation, Springer International Publishing, 2019, pp. 45–53.

[29] M. Abdullah, W. H., R. M., M. Alzahrani, and M. Sadik, “The Impact of Learning Styles on Learner’s Performance in E-Learning Environment,” Int. J. Adv. Comput. Sci. Appl., vol. 6, no. 9, pp. 24–31, 2015.

[30] D. Hooshyar, R. B. Ahmad, M. Yousefi, M. Fathi, S. J. Horng, and H. Lim, “SITS: A solution-based intelligent tutoring system for students’ acquisition of problem-solving skills in computer programming,” Innov. Educ. Teach. Int., vol. 55, no. 3, pp. 325–335, 2018.

[31] T. Saba, “Implications of E-learning systems and self-efficiency on students outcomes: a model approach,” Human-centric Comput. Inf. Sci., vol. 2, no. 1, pp. 1–11, 2012.