Students' Characteristics of Student Model in Intelligent Programming Tutor for Learning Programming: A Systematic Literature Review

Rajermani Thinakaran[1]
Faculty of Data Science and Information Technology
INTI International University
Nilai, Negeri Sembilan, Malaysia

Suriayati Chuprat[2]
Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia
Kuala Lumpur, Malaysia

***Abstract*—This study describes preliminary results of a research related to Intelligent Programming Tutor (IPT) which is derived from Intelligent Tutoring System (ITS). The system architecture consists of four models. However, in this study student model mainly student characteristic was focused. From literature, 44 research articles were identified from a number of digital databases published between 1997 to 2022 base on systematic literature review (SLR) method. The findings show that the majority 48% of IPT implementation focuses on knowledge and skills. While 52% articles focused on a combination of two to three student characteristics where one of the combinations is knowledge and skill. When narrow down, 25% focused on knowledge and skills with errors or misconceptions; 4% focused on knowledge and skill with cognitive features; 5% focused focus on knowledge and skill with affective features; 2% focused on knowledge and skill with motivation; and 9% based on knowledge and skill with learning style and learning preferences as students' characteristics to build their student model. Whereas 5% focused on a combination of three student characters which are knowledge and skill with cognitive and affective features and 2% focused on knowledge and skill with learning styles and learning preferences and motivation as students' characteristics to construct the tutoring system student model. To provide an appropriate tutoring system for the students, students' characteristic needs to decide for the student model before developing the tutoring system. From the findings, it can say that knowledge and skills is an essential students' characteristic used to construct the tutoring system student model. Unfortunately, other students' characteristic is less considered especially students' motivation.**

***Keywords*—Intelligent tutoring system; intelligent programming tutor; student characteristics; student model**

I. Introduction

Intelligent Tutoring System (ITS) is a computer software system that can mimic the methods and dialog of natural human tutors, generate real time and on-demand instructional interactions as and when required by individual students. The implementation of ITSs also incorporate computational mechanisms and knowledge representations in the fields of Artificial Intelligence (AI) which addresses how to reason about intelligence together with multimedia and internet; Psychology on the other hand, consists of Cognitive Science which addresses how people think and learn, while the Education field focuses on how to provide the best support for teaching and learning [1] as illustrated in Fig. 1.

Fig. 2 depicts the evolution of ITS from the 1960s to the year 2000. The introduction of AI techniques and Expert Systems technology to CBI (Computer-Based Instruction) gave rise to ITS [2]. Early 2000, internet has become a central core to the educative environment, thus ITS incorporated with web platform so that the ITS can be accessed anytime and anywhere and known as Adaptive Web-Based Educational System (AWBES) [1].

An ITS architecture basically consists of four models [3] as illustrated in Fig. 3 which are 1) Domain Model - known as the expert or cognitive model. This model contains procedures, theories and problem-solving tactics of the domain to be learned; 2) Student Model - known as user or learner model. Considered as the main component of an ITS. The component gives special responsiveness to the students' cognitive and affective states and their progress in their learning process; 3) Tutoring Model - known as Pedagogical Model or Instructional Model. The model accepts information from the Domain Model and uses Student Model for making decisions on tutoring plans and actions; 4) Interface Model - provides the interface with which the students interact with the ITS.

The main aim of ITS is to improve students' learning process [4]. An ideal condition of the learning process is where students can receive lessons, resolve exercises and obtain immediate feedback. The feedbacks and hints are provided based on the analysis of the responses to each problem-solving step given by students.

In recent years, the development and improvement of ITS has been growing rapidly. Among some of the improvements include improvements on the problem-solving system that can support and help to give feedbacks and hints to students; improvements on model tracing that assesses students' current knowledge that facilitates the next step in order to support problem solving. In addition, improvements on knowledge tracing were also carried out that allows assessment of students' skills and knowledge level in order to release a new tutorial to facilitate learning and finally improvements on tutorial dialogues to support problem solving [5].

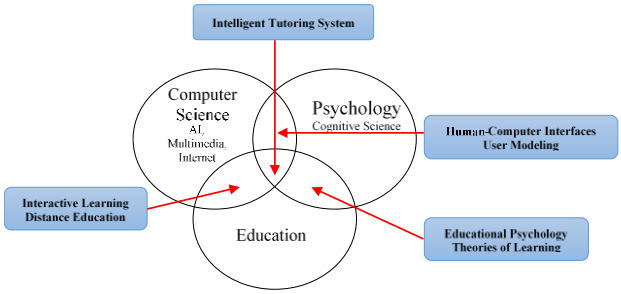

Fig. 1. The Development of ITS.

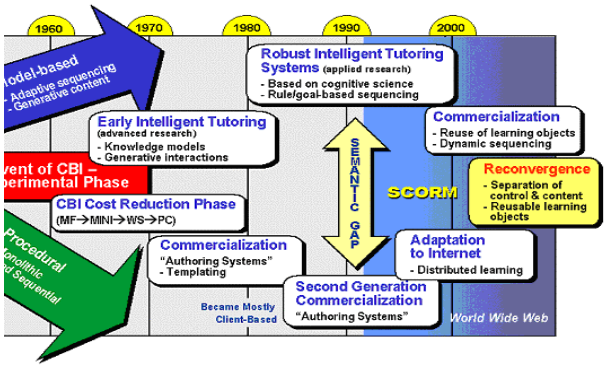

Fig. 2. The Evolution of ITS [2].

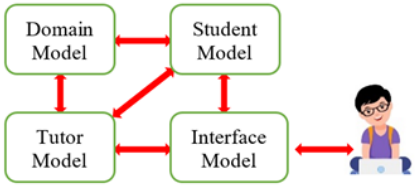

Fig. 3. ITS Architecture.

Educators agree that the most effective form of teaching is through one-to-one interaction with students [6]. In this context, ITS has an upper advantage as it provides personalized tutoring that is tailored to students' needs [5], [7], [8]. According to the findings from previous studies conducted by Kulik and Fletcher [9] and Colchester et al. [10], ITSs have successfully raised students' performance compared to those who were taught in conventional classes.

Chrysafiadi and Virvou [7] claim that, ITS to be become more adaptive and personalize, students' characteristic as student model need to be considered. The students' characteristic comprises of knowledge and skill; errors or misconceptions; learning styles and learning preferences; cognitive features; motivation; and affective features.

Knowledge refers to familiarity with theoretical concepts and factual information and skills refer to the proficiencies developed through practice [7]. During the learning process, errors or misconceptions can be identified. The concept of error or misconceptions can be defined as a process or fact that does not match a given norm [11]. Learning Styles and Learning Preferences - refer to how a student identifies, gathers and processes their learning materials [12]. While Cognitive features refer to students' aspects such as attention, knowledge, ability to learn and recall memory, opinion, attention, collaborative skills, capabilities to solve problems and make conclusions, analyzing abilities and critical thinking [7]. Literally motivation is the desire to do things. Motivation plays a significant role in students' learning process [13]. Emotional factors are known as affective features such as sadness, happiness, frustrations, anger, interest, boredom, distractions, aims and confusion [14]. Subsequently, affective features can be based on students' motivations [7].

Among the stated students' characteristics, motivation is considered as the main factor for engaging students in their learning [15], [16] and in academic performance [17]. Meanwhile, Abuhmaid [18], Hamzah et al., [19] and Sundar and Kumar [20] argue that students' motivation is an important factor in ensuring the success of ITS implementation. Abuhmaid [18] also pointed that motivation factors need to be considered when designing any ITS materials. Studies on the relationship between motivating factors and learning have been a prominent research topic in the field of education as well as studies focusing on eLearning [15].

McGill [21], for example, studies the use of robots to influence students' motivation when learning introductory programming. In order to ensure students feel motivated to use eLearning, Hamzah et al., [19] applied the ARCS+G (Attention, Relevance, Confidence, Satisfaction + Gamification) as motivational design model in the development process in their study. Another study conducted by Nikou and Economides [22] examined the impact of using mobile devices during the learning activity on students' learning motivation. Results obtained by Abuhmaid [18] reveal that utilizing the flipped learning strategy in an eLearning environment has a significant improvement on students' motivation to learn. Tambunan, Rusdi and Miarsyah [23] suggest that a combined usage of eLearning with a Problem Based Learning (PBL) model and motivation is an effective way to improve students' learning outcomes. Even though some may argue that a game environment can be used to ensure a good transfer of knowledge in a fun way, Yedri et al., [24] claim that a balance between learning transfer and motivation is the major key to success.

A. Intelligent Tutoring System for Learning Programming

Programming tools have been actively researched in their effectiveness to support teaching and learning. Pears and his colleagues [25] have summarized these programming tools into five categories one of which is ITSs. As highlighted in the previous section, ITSs provide many benefits in students' learning process.

An IPT (Intelligent Programming Tutor) is a specific implementation of an ITS for learning programming. The ideas behind the use of IPT are to create a learning process where students can receive tutelage, resolve exercises and receive instant feedbacks imitating one-to-one human tutoring.

As explained above, IPTs is derived from ITS' ideas in which students' characteristics also need to be considered in creating a conducive and effective learning process. In the following section, an exhaustive systematic literature review (SLR) was carried out to identify what types of student characteristics were used to design the IPT.

II. METHOD

In this section, a SLR (Systematic Literature Review) was carried out to obtain answers to the following question: What types of student characteristics were used to design the IPT? SLR was conducted in this study as it is a process that can be used for recognizing, evaluating and interpreting research materials to answer several research questions [26].

To answer the question stated above, PICOC as proposed by Petticrew and Robert [27] was used in the study. PICOC comprise of five elements which are Population, Intervention, Comparison, Outcomes and Context. Table I shows a summary of PICOC for this study.

TABLE I. SUMMARY OF PICOC

Population	Student
Intervention & Comparison	Intelligent Programming Tutor or Intelligent Tutoring System
Outcomes	Student characteristics in Intelligent Programming Tutor or Intelligent Tutoring System
Context	Reviews of all studies of Intelligent Programming Tutor or Intelligent Tutoring System within the domain of Programming subject

The identification of primary sources from journals, conferences and online databases is important to ensure a wide coverage of potential sources. A survey of literature included all research works published from online database such as ACM Digital Library, Google Scholar, IEEE Explore, ISI Web of Knowledge, Science Direct and Springer. These online databases were selected to be used in this study as they are the most popular and frequent databases used by previous researchers in investigating the use of eLearning. In addition, references retrieved from SLR articles were analyzed to identify any literature that may have been ignored or overlooked during the search.

Using Booleans of AND, OR in the keyword combinations conducted include keywords such as intelligent programming tutor; intelligent tutoring system AND programming; and programming AND intelligent tutoring system. The use of the Boolean OR is to incorporate alternative synonyms and spellings while the usage of the Boolean AND is to link the major terms.

A search of these databases and journals allow the data collection to be inclusive and comprehensive. A total of 73 papers were selected which the searching technique described above. These articles were reviewed while papers that were not categorized under any refereed journal articles such as proceedings or editor-reviewed papers were excluded from the final analysis. The focus of this review was on refereed articles which assist in ensuring the quality and relative rigor of data sources. Therefore, research papers that did not conduct an in-depth discussion of their ITS in programming particularly those that did not investigate student characteristic(s) were omitted. Papers that were not published in English language and gray papers such as those without any bibliographic information (publication date/type, volume and issue numbers) were also excluded. Duplicated papers were also excluded (only the most recent, complete and improved one is included) from the SLR in this study. In total, 44 relevant papers which were published between 1997 and 2022 were gathered and thoroughly examined in this study.

III. RESULT AND DISCUSSION

A. Data Analysis

In Table II, the 44 articles were organized based on what type of student characteristics was used to construct the user model; what modelling technique was applied to construct the user model for the intended IPT system; the subject domain and learning environment.

B. Synthesis of SLR on Student Characteristics for Student Model in IPTs

Fig. 4 illustrates how students' characteristics were considered in constructing user model mainly in IPTs base on 44 articles which was revealed in Table II.

From Fig. 4, 48% or 21 articles were found mainly focused on knowledge and skills as student characteristic for their user model. The objective of this characteristic is to improvised students' theoretical concepts knowledge and programming proficiencies skills in particular topic of programming subject. To develop the user model base on knowledge and skills as student characteristic, different researchers use different modelling technique such as Bayesian network, Markov Decision Process, Regression model, Rule Base and K* classifier. However, the common modelling technique is Rule Base because it is easier to build due to improved authoring tools and remain a popular option.

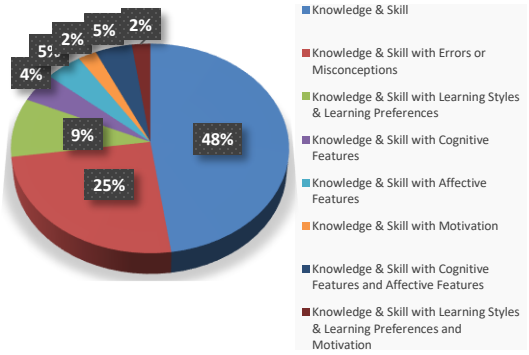

Fig. 4. Students' Characteristics for user Model in IPTs.

TABLE II. SUMMARY OF IPT REVIEW

Student Characteristic	IPT	Modelling Technique	Subject Domain and Learning Environment
Knowledge & Skill	ALLIGATOR [28]	-	To teach data flow diagram using visual programming environment with multiple informative and tutoring feedback components.
	AOMS [29]	-	To teach graph using C Programming
	BITS [30]	Bayesian network	To teach programming concept using C++.
	BOTS [31]	Markov Decision Process	To teach pseudocode in using game environment.
	ChiQat – Tutor [32]	Regression model	Using visualization to teach link list.
	COLLEGE [33]	-	Editing, compiling the source code, and executing the object code with aid of animation and visualization.
	CIMEL ITS [34]	Rule Base	To teach OOP concepts and observe students' progress and offer assistance based on pedagogical strategies adapted to the individual student.
	CPP-Tutor [35]	-	To teach programming concept using C++ and provide hints and feedback during problem-solving in tutorials.
	CS-I [36]	Rule Base	To teach programming concept using C++ and detect each students' level of understanding.
	DrJava [37]	-	To write, test and debug Java programs.
	EduJudge [38]	-	Submission, management and automatic evaluation of programming exercises.
	KSC-PaL [39]	K* classifier	Collaborate with students to solve problems on data structures mainly on linked lists, stacks, and binary search tree.
	Marmoset [40]	-	To write and test Java code and helps the instructor to monitor student progress.
	OmniCode [41]	-	To teach novice students basic programming concept using Python.
	ProgTool [42]	-	Using visualization to teach OOP concepts.
	PASS [43]	-	Assisting beginners in learning programming.
	PLTutor [44]	Rule Base	To teach syntax and semantic of JavaScript using visualization.
	PLWeb [45]	-	Assist instructors to design computer programming exercises and to help students to study and practice programming exercises.
	SCALE [46]	-	Teach multiple topics consisting of pseudocode, sequential search, binary search subprograms and recursive.
	ViLLe [47]	-	Developed to visualize programming syntax written by students in Java or C++.
	WebTask [48]	-	To write Java code in a method body, testing, and received feedback in animation and visualization environment.
Knowledge & Skill with Errors or Misconceptions	ADIS [49]	Constraint Based Modelling	To teach basic algorithms of linked-lists, stacks, queues, trees and graph by visually.
	ADIL [50]	-	To teach C language and explain logical errors.
	AutoLEP [51]	-	Helps students to find and work through bugs in C language and also provides immediate detailed feedback.
	Collab ChiQat [52]	-	Developed to teach linked lists, stacks, and binary search trees in collaboration environment.
	iList [53]	-	Helps students to learn linked lists in visualization form.
	iSnap [54]	Contextual Tree Decomposition algorithm	To teach programming control structure using block-based programming.
	INCOM [55]	Constraint Based Modelling	Help students on programming logic using Prolog.
	J-LATTE [56]	Constraint Based Modelling	To teach Java in terms of design and syntax.
	OOPs [57]	Constraint Based Modelling	Help students to understand and overcome their misconceptions in OOP and reinforce the correct learning methods.
	ProBot [58]	Rule Base	Use game concept to improve students' abilities in programming control structures.

	@KU-UZEM [59]	Constraint Based Modelling	To teach C language and to overcome misconceptions in terms of concept and syntax.
Knowledge & Skill with Learning Styles & Learning Preferences	ABITS [60]	Association Rule Mining And Fuzzy C-Means Clustering	Introduction to Java Programming.
	EDUCA [61]	Neural Networks	Introduction to Maya Programming Language.
	ELM-ART [62]	-	Introduction to LISP Programing. Sample based problem solving support, detailed analysis of student answers, solving support, reminder option.
	Protus [63]	-	Help students during programming learning process by advising students to take an appropriate action when needed, monitoring their progress and tracking student learning styles.
Knowledge & Skill with Cognitive Features	APT [64]	ACT-R theory	To write short programs in Lisp, Pascal or Prolog
	WPAS [65]	-	Supporting programming learning activities with various difficulty levels.
Knowledge & Skill with Affective Features	E-Learning 3.0 [66]	Fuzzy-Logic and Nayve Bayes classifier algorithm	To teach Java according student emotions.
	PIT [67]	-	Teach programming skill and provide feedback based on students' emotion.
Knowledge & Skill with Motivation	FITS [68]	Bayesian networks	To teach flowchart using game environment.
Knowledge & Skill with Cognitive Features and Affective Features	Java Sensei [69]	Neural Networks, Fuzzy Logic	To teach Java and analyze student cognitive and emotion level during using the system.
	JavaTutor [70]	ACT-R theory & machine learning techniques	Teach Java by interacting human-to-computer and body expressions.
Knowledge & Skill with Learning Styles & Learning Preferences and Motivation	LOs [71]	Rule Base	Used simulation-based to teach array sorting.

Whereas another 52% or 23 articles focused on a combination of two to three student characteristics where one of the combinations is knowledge and skill. From the study, it was identified that 25% or 11 articles focused on knowledge and skills with errors or misconceptions as students' characteristics to build the student model. The researchers aim is to improvise the students' knowledge by learning from mistakes. From the educational point of view, learning from mistake or error can be powerful learning process, especially for learning programming. Students learn much faster when they made mistake first, especially in programming. In other words, getting the incorrect answer helps them to remember the correct one. To develop the student model base on these two students' characteristics, the researchers has considered three different modelling techniques which are Constraint Based Modelling, Contextual Tree Decomposition algorithm and Rule Base. Among these three techniques, Constraint Based Modelling is most preferred by the researcher because the algorithm was originally developed as a hypothesis about how student learn from their mistakes.

Two articles or 4% focused on knowledge and skill with cognitive features as students' characteristics for their student model. The cognitive features are students' ability to learn and recall memory and also capabilities to solve problems and make conclusion were used to construct the user model. ACT-R (Adaptive Control of Thought—Rational) theory was used to develop the tutoring system student model. The theory using a cognitive architecture that uses production rules to model student problem solving processes.

Another two articles or 5% focused focus on knowledge and skill with affective features as students' characteristics for their student model. This model was able to recognize and analyze student emotion such as frustration, boredom, engagement, confusion and excitement through students' facial expressions. The system was developed base on Fuzzy-Logic and Naive Bayes classifier algorithm.

One article or 2% focused on knowledge and skill with motivation as students' characteristics to construct the user model. The model was developed using Bayesian networks in game environment call tic-tac-toe. While another four article or 9% construct the student model based on knowledge and skill with learning style and learning preferences. The model able to track students' learning styles during their learning process by advising students to take an appropriate action when needed. Association Rule Mining and Fuzzy C-Means Clustering and Neural Networks were considered as modelling technique to construct the student model base on students' characteristics.

There are two articles or 5% focused on a combination of three student characters which are knowledge and skill with cognitive and affective features. The user model was developed to analyze students' cognitive and emotional conditions during the learning process. The model use ACT-R theory for knowledge representation while affective features were obtained through body expressions using sensor detection. The detection performed using machine learning techniques.

Lastly, one article (2%) focused on knowledge and skill with learning styles and learning preferences and motivation as students' characteristics to construct the tutoring system

student model. The model able to detect students' learning styles and preferences using some predefine rules during the learning process and to motivate the students, simulation and visualization was used in the system user interface design.

From Jamal and Naemah [72] point of view, the effective teaching of programming subjects can be achieved by providing an appropriate tutoring system for the students. To achieve this, what type of students' characteristic need to be decided for the student model before developing the tutoring system [7]. From Fig. 4, it can say that knowledge and skills is an essential students' characteristic used to construct the tutoring system student model. Unfortunately, other students' characteristic is less considered especially students' motivation.

From the findings presented in Table II and Fig. 4, it can be seen that only 4.0% or 2 articles [68], [71 considered motivation as a student characteristic for the student model in IPT. On the other hand, Hooshyar et al. [68] used the game approach to motivate students to learn programming algorithm while Tuparov, Tuparov and Jordanov [71] used simulation-based ITP to help motivate students to understand array sorting. These motivations only encourage students as per view only.

Based on the results obtained from the existing literature thoroughly discussed above, it can be concluded that there is a lack of focus on motivation as a students' characteristic for student model mainly in IPT and generally in ITSs. Since student motivation is an important factor [15], [73] in learning programming [74], therefore the same consideration needs to be considered at the IPTs level and also ITSs.

IV. Conclusion

IPT is derived from ITSs. To develop an IPT system, students' characteristics need to be considered first before construct the student model which is one of important model in ITS architecture. From this study, it was identified that motivation was less considered as students' characteristics in constructing student model for IPT and generally in ITSs. Motivation and learning are highly complex aspects of human behaviour. Motivation has been agreed as a crucial aspect affecting learning behaviour, learning process and learning achievement. So, the same concern need to be consider in tutoring system implementation where can bring numerous benefits.

References

[1] A. Alkhatlan and J. Kalita, "Intelligent tutoring systems: A comprehensive historical survey with recent developments," International Journal of Computer Applications (0975 - 8887), Volume 181 - No.43, March 2019.

[2] L. Samuelis, "Notes on the components for intelligent tutoring systems," Acta Polytechnica Hungarica, 4(2), pp. 77-85, 2007.

[3] A. K. Erümit and İ. Çetin, "Design framework of adaptive intelligent tutoring systems," Education and Information Technologies, 25(5), pp. 4477-4500, 2020.

[4] B. Vesin, M. Ivanović, A. Klašnja-Milićević, and Z. Budimac, "Personal Assistance Agent in Programming Tutoring System", In Agent and Multi-Agent Systems: Technologies and Applications, pp. 441-451, Springer International Publishing, 2015.

[5] E. Dehkourdy, A. Reza, R. Mohasanati, and S. Hakimnia, "Main Components of Intelligent Tutoring Systems", Life Science Journal, 10(8), 2013.

[6] H. Bui, "A Classification of Data-Driven Hint Generation Techniques for Code-Writing Intelligent Tutoring Systems," American Journal of Computer Science and Information Engineering, 4(2), pp. 16-23, 2017.

[7] K. Chrysafiadi, and M. Virvou, "Student modeling approaches: A literature review for the last decade', *Expert Systems with Applications*, 40(11), pp. 4715-4729, 2013.

[8] T. W. Price, Y. Dong, and D. Lipovac, "iSnap: Towards Intelligent Tutoring in Novice Programming Environments," In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp. 483-488, ACM, March 2017.

[9] J. A. Kulik, and J. D. Fletcher, J. D. "Effectiveness of intelligent tutoring systems: a meta-analytic review', Review of Educational Research, 86(1), pp. 42-78, 2016.

[10] K. Colchester, H. Hagras, D. Alghazzawi, and G. Aldabbagh, "A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms," Journal of Artificial Intelligence and Soft Computing Research, 7(1), pp. 47-64, 2017.

[11] J. Ljubomir, "Teaching Introductory Programming: Agent-based Approach with Pedagogical Patterns for Learning by Mistake," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No.6, 2014.

[12] M. A. Ghazal, N. A. M. Zin, and Z. Muda, "Designing Domain Model For Adaptive Web-based Educational System According to Herrmann Whole Brain Model," Journal of Engineering Research and Technology, 3(3), 2016.

[13] T. Khan, K. Johnston, and J. Ophoff, "The Impact of an Augmented Reality Application on Learning Motivation of Students," Advances in Human-Computer Interaction, Volume 2019, 2019.

[14] C. Cunha-Pérez, M. Arevalillo-Herráez, L. Marco-Giménez, and D. Arnau, "On Incorporating Affective Support to an Intelligent Tutoring System: an Empirical Study," IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 13(2), pp.63-69, 2018.

[15] Y. Azliza, A. Noraida, Y. M. Hafiz, M.S.M. Yazid Sukinah and S. N. Suhana, "Learning Motivation Assessment Model: A Review," Australian Journal of Basic and Applied Sciences, 8(4) Special 2014, pp. 163-169, 2014.

[16] F. T. Leow, and M. Neo, "Peer Interaction and Students' Perceptions Towards Constructivist-Collaborative Learning Environment: Motivation and Affective Factor," In ICEL2016-Proceedings of the 11th International Conference on e-Learning: ICEl2016, pp. 87, Academic Conferences and publishing limited, June 2016.

[17] J. Cibulka, and G. A. Giannoumis, "Augmented and Virtual Reality for Engineering Education," In Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58), No. 138, pp. 209-219, Linköping University Electronic Press, Sptember 2017.

[18] A. Abuhmaid, "The Impact of Using Flipped Learning Strategy on Students' motivation for Learning," 10th annual International Conference of Education, Research and Innovation, Seville (Spain), 2017.

[19] W. M. A. F. W. Hamzah, N. H. Ali, M. Y. M. Saman, M. H. Yusoff, and A. Yacob, "Influence of gamification on students' motivation in using e-learning applications based on the motivational design model," International Journal of Emerging Technologies in Learning (iJET), 10(2), pp. 30-34, 2015.

[20] P.P. Sundar, and A. S. Kumar, "A systematic approach to identify unmotivated learners in online learning," Indian Journal of Science and Technology, 9(14), 2016.

[21] M. M. McGill, "Learning to program with personal robots: Influences on student motivation," ACM Transactions on Computing Education (TOCE), 12(1), pp. 4, 2012.

[22] S. A. Nikou, and A. A. Economides, (2016) "The impact of paper-based, computer-based and mobile-based self-assessment on students' science motivation and achievement," Computers in Human Behavior, 55, pp. 1241-1248, 2016.

[23] L. Tambunan, R. Rusdi, and M. Miarsyah, "Efectiveness of Problem Based Learning Models by Using E-Learning and Learning Motivation

Toward Students Learning Outcomes on Subject Circullation Systems," Indonesian Journal of Science and Education, 2(1), pp. 96-104, 2018.

[24] O. B. Yedri, L. El Aachak, A. Belahbib, H. Zili, and M. Bouhorma, "Motivation Analysis Process as Service Applied on Serious Games," International Journal of Information Science and Technology, 2(1), pp. 4-11, 2018.

[25] A. Pears, S. Seidman, C. Eney, P. Kinnunen, and L. Malmi, "Constructing a core literature for computing education research," ACM SIGCSE Bulletin, 37(4), pp. 152-161, 2005.

[26] R. Thinakaran, and R. Ali, "Work in progress: An initial review in programming tutoring tools," In 2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), pp. 1-4, IEEE, December 2015.

[27] M. Petticrew, and H. Roberts, Systematic reviews in the social sciences: A practical guide. John Wiley & Sons, 2008.

[28] Mosconi, M., Ottelli, D. and Porta, M. (2003) 'Alligator, a Web-based Distributed Visual Programming Environment', WWW (Posters), pp. 3.

[29] M. Gaeta, F. Orciuoli, and P. Ritrovato, "Advanced ontology management system for personalised e-Learning," Knowledge-Based Systems, 22(4), pp. 292-301, 2009.

[30] C. J. Butz, S. Hua, and R. B Maguire, "A web-based intelligent tutoring system for computer programming," In Web Intelligence, 2004. WI 2004. Proceedings, IEEE/WIC/ACM International Conference on pp. 159-165, IEEE, September 2004.

[31] A. Hicks, Y. Dong, R. Zhi, V. Cateté, and T. Barnes, "BOTS: Selecting Next-Steps from Player Traces in a Puzzle Game," In EDM (Workshops), June 2015.

[32] N. Green, B. Di Eugenio, R. Harsley, D. Fossati, O. AlZoubi, and M. Alizadeh, "Student behavior with worked-out examples in a computer science intelligent tutoring system," In International Conference on Educational Technologies, November 2015.

[33] C. Bravo, M. J. Marcelino, A. J Gomes, M. Esteves, and A. J. Mendes, "Integrating Educational Tools for Collaborative Computer Programming Learning," J. UCS, 11(9), pp. 1505-1517, 2005.

[34] S. H. Moritz, F. Wei, S. M. Parvez, and G.D. Blank, "From objects-first to design-first with multimedia and intelligent tutoring," In ACM SIGCSE Bulletin, Vol. 37, No. 3, pp. 99-103, ACM, June 2005.

[35] S. Naser, "Evaluating the effectiveness of the CPP-Tutor an intelligent tutoring system for students learning to program in C++," Journal of Applied Sciences Research, 5(1), pp. 109-114, 2009.

[36] J. P. Yoo, S. J. Seo, and S. K. Yoo, "Designing an Adaptive Tutor for CS-I Laboratory," In International Conference on Internet Computing, pp. 459, 2004.

[37] E. Allen, R. Cartwright, and B. Stoler, "DrJava: A lightweight pedagogic environment for Java," In *ACM SIGCSE Bulletin*, Vol. 34, No. 1, pp. 137-141, ACM, February 2002.

[38] E. Verdú, L. M. Regueras, M. J. Verdú, J. P. Leal, J. P. de Castro, and R. Queirós, "A distributed system for learning programming on-line," Computers & Education, 58(1), pp. 1-10, 2012.

[39] C. Howard, P. Jordan, B. Di Eugenio, and S. Katz, S. "Shifting the load: A peer dialogue agent that encourages its human collaborator to contribute more to problem solving," International Journal of Artificial Intelligence in Education, 27(1), pp. 101-129, 2017.

[40] J. Spacco, D. Hovemeyer, W. Pugh, F. Emad, J. K. Hollingsworth, and N. Padua-Perez, "Experiences with marmoset: designing and using an advanced submission and testing system for programming courses," ACM Sigcse Bulletin, 38(3), pp. 13-17, 2006.

[41] H. Kang, and P. J. Guo, "Omnicode: A Novice-Oriented Live Programming Environment with Always-On Run-Time Value Visualizations,' 2017.

[42] M. Goyal, "Development of agent-based intelligent tutoring system for teaching object-oriented programming concepts," In Proceedings of the 9th International Conference on Education and Information Systems, Technologies and Applications (EISTA 2011), pp. 17-22, 2011.

[43] K. M. Law, V. C. Lee, and Y. T. Yu, "Learning motivation in e-learning facilitated computer programming courses," Computers & Education, 55(1), pp. 218-228, 2010.

[44] G. L. Nelson, B. Xie, and A. J. Ko, "Comprehension First: Evaluating a Novel Pedagogy and Tutoring System for Program Tracing in CS1," In Proceedings of the 2017 ACM Conference on International Computing Education Research, pp. 2-11, ACM, August 2017.

[45] S. H. Tung, T. T. Lin, and Y. H. Lin, "An exercise management system for teaching programming," Journal of Software, 8(7), pp. 1718-1725, 2013.

[46] I. Verginis, A. Gogoulou, E. Gouli, M. Boubouka, and M. Grigoriadou "Enhancing learning in introductory computer science courses through SCALE: An empirical study", IEEE transactions on education, 54(1), pp. 1-13, 2011.

[47] T. Rajala, M. J. Laakso, E. Kaila, and T. Salakoski, T. "Effectiveness of Program Visualization: A Case Study with the ViLLE Tool," Journal of Information Technology Education, 7, 2008.

[48] G. Rößling, "A family of tools for supporting the learning of programming," Algorithms 2010, 3(2), pp.168-182, 2010.

[49] K. Warendorf, and C. Tan, "ADIS-An animated data structure intelligent tutoring system or Putting an interactive tutor on the WWW," In Proceedings of Workshop Intelligent Educational Systems on the World Wide Web at AI-ED, Vol. 97, pp. 54-60, August 1997.

[50] A.M. Zin, S. A. Aljunid, Z. Shukur, and M. J. Nordin, "A Knowledge-based automated debugger in learning system," Proceedings of the 4th International Workshop on Automated Debugging, (WAD' 01), ACM Press, Munich, 2001.

[51] T. Wang, X. Su, P. Ma, Y. Wang, and K. Wang, "Ability-training-oriented automated assessment in introductory programming course," Computers & Education, 56(1), pp. 220-226, 2011.

[52] R. Harsley, D. Fossati, B. Di Eugenio, and N. Green, "Interactions of Individual and Pair Programmers with an Intelligent Tutoring System for Computer Science," In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp. 285-290, ACM, March 2017.

[53] D. Fossati, B. Di Eugenio, S. Ohlsson, C. Brown, and L. Chen, "Data driven automatic feedback generation in the iList intelligent tutoring system," Technology, Instruction, Cognition and Learning, 10(1), pp. 5-26, 2015.

[54] T. W. Price, Y. Dong, and D. Lipovac, "iSnap: Towards Intelligent Tutoring in Novice Programming Environments," In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp. 483-488, ACM, March 2017.

[55] N. T. Le, W. Menzel, and N. Pinkwart, "Evaluation of a Constraint-Based Homework Assistance System for Logic Programming," In Proceedings of the 17th International Conference on Computers in Education, 2009.

[56] J. Holland, A. Mitrovic, and B. Martin, "J-LATTE: A Constraint-based Tutor for Java," Proceedings of the 17th International Conference on Computers in Education, ICCE 2009, pp. 142–146. Asia-Pacific Society for Computers in Education, Hong Kong, 2009.

[57] J. Gálvez, E. Guzmán, and R. Conejo, R. (2009) "A blended E-learning experience in a course of object oriented programming fundamentals," Knowledge-Based Systems, 22(4), pp. 279-286, 2009.

[58] J. Moreno, "Digital Competition Game to Improve Programming Skills," Journal of Educational Technology & Society, 15(3), pp. 288, 2012.

[59] U. Kose, and O. Deperlioglu, "Intelligent learning environments within blended learning for ensuring effective c programming course", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.1, pp. 105 – 124, 2012.

[60] T. T. Sampathkumar, R. Gowri, and V. Venkateswaran, "Designing an adaptive distributed tutoring system based on Students' learning style and collaborative learning using intelligent agents. International Journal of Computer Applications, 87(17), 2014.

[61] R. Z. Cabada, M. L. B. Estrada, and C. A. R. García, "EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network," Expert Systems with Applications, 38(8), pp. 9522-9529, 2011.

[62] G. Weber, and P. Brusilovsky, "ELM-ART–An interactive and intelligent web-based electronic textbook," International Journal of Artificial Intelligence in Education, 26(1), pp. 72-81, 2016.

[63] B. Vesin, M. Ivanović, A. Klašnja-Milićević, and Z. Budimac, "Personal Assistance Agent in Programming Tutoring System," In Agent and Multi-Agent Systems: Technologies and Applications, pp. 441-451, Springer International Publishing, 2015.

[64] A. Corbett, "Cognitive mastery learning in the ACT programming tutor," AAAI Technical Report SS-00-01, 2000.

[65] W. Y. Hwang, R. Shadiev, C. Y. Wang, and Z. H. Huang, "A pilot study of cooperative programming learning behavior and its relationship with students' learning performance," Computers & Education, 58(4), pp. 1267-1281, 2012.

[66] R. Z. Cabada, M. L. B.cEstrada, F. G.cHernández, R. O. Bustillos, and C. A. Reyes-García, "An affective and Web 3.0-based learning environment for a programming language," Telematics and Informatics, 2017.

[67] Tiam-Lee, T. J. and Sumi, K. (2018, June) Adaptive Feedback Based on Student Emotion in a System for Programming Practice. In International Conference on Intelligent Tutoring Systems, pp. 243-255, Springer, Cham., June 2018.

[68] D. Hooshyar, R. B. Ahmad, M. Yousefi, M. Fathi, S. J. Horng, and H. Lim, "Applying an online game-based formative assessment in a flowchart-based intelligent tutoring system for improving problem-solving skills," Computers & Education, 94, pp.18-36., 2016.

[69] 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," In Mexican International Conference on Artificial Intelligence, pp. 171-182, Springer, Cham, October 2015.

[70] J. B. Wiggins, K. E. Boyer, A. Baikadi, A. Ezen-Can, J. F. Grafsgaard, E. Y. Ha, and E. N. Wiebe, "JavaTutor: an intelligent tutoring system that adapts to cognitive and affective states during computer programming," In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, pp. 599-599, ACM, February 2015.

[71] G. Tuparov, D. Tuparova, and V. Jordanov, "Teaching sorting and searching algorithms through simulation-based learning objects in an introductory programming course," Procedia-Social and Behavioral Sciences, vol. 116, pp. 2962-2966, 2014.

[72] O. Jamal, and A. Naemah, "The Uncommon Approaches of Teaching the Programming Courses: The Perspective of Experienced Lecturers," Computing Research & Innovation (CRINN), Vol. 1, pp. 64, 2016.

[73] A. de Vicente, Towards Tutoring Systems that Detect Students' Motivation: An Investigation. PhD. thesis, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh, 2003.

[74] M. M. McGill, "Learning to program with personal robots: Influences on student motivation," ACM Transactions on Computing Education (TOCE), 12(1), pp. 4, 2012.