MDPI

Article

A Cognitive Diagnostic Module Based on the Repair Theory for a Personalized User Experience in E-Learning Software

Akrivi Krouska [ID]**, Christos Troussas *** [ID] **and Cleo Sgouropoulou**

Department of Informatics and Computer Engineering, University of West Attica, 12243 Athens, Greece; akrouska@uniwa.gr (A.K.); csgouro@uniwa.gr (C.S.)
* Correspondence: ctrouss@uniwa.gr

Abstract: This paper presents a novel cognitive diagnostic module which is incorporated in e-learning software for the tutoring of the markup language HTML. The system is responsible for detecting the learners' cognitive bugs and delivering personalized guidance. The novelty of this approach is that it is based on the Repair theory that incorporates additional features, such as student negligence and test completion times, in its diagnostic mechanism; also, it employs a recommender module that suggests students optimal learning paths based on their misconceptions using descriptive test feedback and adaptability of learning content. Considering the Repair theory, the diagnostic mechanism uses a library of error correction rules to explain the cause of errors observed by the student during the assessment. This library covers common errors, creating a hypothesis space in that way. Therefore, the test items are expanded, so that they belong to the hypothesis space. Both the system and the cognitive diagnostic tool were evaluated with promising results, showing that they offer a personalized experience to learners.

Keywords: adaptive content; diagnostic model; error diagnosis; learner experience; personalized guidance; repair theory; student bug

1. Introduction

During the last decades, Adaptive Educational Hypermedia Systems (AEHS) have prevailed in the field of online learning, since they can form a depiction of each unique user's objectives, interests, and cognitive ability [1]. Moreover, they can be utilized to tailor the learning environment to their needs and preferences. Usually, the students' goal is to learn all the learning material or at least a significant part of it. That means that students' knowledge level is a determinant for their interaction with the system and it can alter based on their performance [2]. For instance, the knowledge level of a user can vary greatly in comparison to others, but also in other cases, it can increase quickly. As such, the same educational material can be vague for a beginner learner and at the same time trivial and boring for an advanced learner. Moreover, especially for beginners, it needs to be noted that they start using the system knowing nothing about the specific subject being taught, and most of the material will lead to subjects that are completely new to them. These users need guidance to find the "right" educational path. The guidance can take the form of diagnosis of misconceptions and/or errors.

As mentioned earlier, the knowledge of the users on the subject seems to be the most important characteristic for error diagnosis in most AEHSs. Almost all adaptive presentation techniques, e.g., fuzzy weights [3,4], artificial neural networks [5,6], multiple-criteria decision analysis [7,8], are based on user knowledge as the main source of personalization. User knowledge is a variable for each user. This means that an AEHS, being based on user knowledge, must recognize changes in the user knowledge state updating the user model accordingly.

One possible way for depicting the knowledge of students in comparison to the knowledge held by the system is the approach of the overlay modeling. The overlay

model is one of the most often used and popular student models. It has been proposed by Stansfield et al. [9] and incorporated in several different learning technology systems. The overlay model is based on the idea that a learner's knowledge of the domain may be partial, yet valid. As a result, the student model is a subset of the domain model [10], which shows expert-level knowledge of the subject [11], according to overlay modeling. The discrepancies between the student's and expert's sets of knowledge are thought to be due to the student's lack of skills and knowledge, and the instructional goal is to minimize these differences.

A disadvantage of the overlay model is its inability to represent possible misunderstandings (misconceptions) of the user [12]. For this purpose, the buggy model has been proposed representing the user's knowledge as the union of a subset of the field of knowledge and a set of misunderstandings. The buggy model helps to better correct the user's mistakes since the existence of an image for the wrong knowledge is very useful from a pedagogical point of view.

In the bug catalog model, there is a large library of predefined misinterpretations that are used to add the relevant misinterpretations to the user model. A disadvantage of this model is the difficulty of creating the library of misinterpretations. The user's misinterpretations are detected during the assessment process. Usually the library contains symbolic rules—conditions and actions that are performed when they are activated.

Analyzing the related literature, there is strong evidence that the field of error diagnosis in e-learning software has been poorly researched for the learning of different concepts, the most prevalent of which are language learning and computer programming. Concerning language learning, there have been several research efforts that present the error diagnosis process can diagnose, among others, grammatical, syntactic, vocabulary mistakes by using techniques, such as approximate string matching, convolutional sequence to sequence modeling, context representation, etc. [13–19]. For example, the work of [19] proposes a sequence-to-sequence learning approach using recurrent neural networks for conducting error analysis and diagnosis. The main idea is that errors may hinder in specific words and with this approach, the error correction can happen successfully. Another example is the work of [13]. The authors employed a context representation approach to detect grammatical errors emerging from the vagueness problems of words. In the work of [14], the authors used the Clause Complex model to analyze the learners' errors emerging from grammatical differences in language learning. The work of [15] proposes a framework of hierarchical tagging sets to perform annotation of grammatical mistakes in language learning. Finally, the authors of [16] performed classification on spelling mistakes in two categories, i.e., orthographic and phonological errors. Concerning computer programming, the researchers perform error diagnosis for identifying either syntax or logic errors, by employing different intelligent techniques, such as fuzzy logic, periodical advice delivery about program's behavior, concept maps, highlighting similarities [20–26]. It must be noted that all the aforementioned mistakes (e.g., grammatical or vocabulary in language learning systems, and syntax or logical in programming learning systems) may emerge from different causes, such as negligence or incomplete knowledge [27–29].

From the presented literature review, it can be inferred that researchers fail to adequately blend theories and models with intelligent techniques to support the process of error diagnosis. In our approach, we employed the Repair theory [30,31] to explain how students can learn with specific attention to the learning way and the reasons of their misconceptions. To extend the efficiency of the presented module, we incorporated a buggy model, associated to the assessment process. This model holds several possible reasons for learners' misconceptions, such as carelessness or knowledge deficiency. The novelty of our approach is not only the use of buggy modeling based on the Repair theory, but also the exploitation of the exported diagnosis for recommending the optimal learning path to students. In particular, the system using the diagnostic mechanism detects the possible reason of students' misconceptions and provides tailored descriptive feedback about the

score achieved, the test duration, and the learning path that should be followed in order for the students to improve their learning outcomes and knowledge bugs detected.

2. Diagnosis of Student Cognitive Bugs and Personalized Guidance

When evaluating the student performance, the e-learning systems mainly consider only the number of incorrect answers and based on the score achieved, they construct the student profile [32]. However, this score is not representative of the actual student knowledge and skills, since an incorrect answer does not always imply a cognitive gap, but it may occur due to student carelessness [33]. As such, the reason why students fail to answer correctly in tests is of great importance in order to provide them the proper guidance for increasing their learning outcomes.

To this direction, the paper presents an integrated student bugs diagnostic mechanism, embodied into an e-learning system, for detecting the student cognitive bugs and providing personalized guidance. The novelty of this approach is that not only is it based on Repair theory incorporating additional features, such as student carelessness and completion time of tests, in its diagnostic mechanism, but it also recommends to students the optimal learning path according to their misconceptions using descriptive feedback on the test and adapting learning content.

Considering the Repair Theory, the presented diagnostic mechanism uses a buggy rule library to explain the causes of students' bugs, observed during the assessment process. This library includes the common bugs, creating a hypothesis space in that way. Hence, the test items are developed in order that they appertain to the hypothesis space. The buggy rules were constructed by 10 computer science professors in Greek Universities, who have taught the HTML language for at least three years. In particular, they were asked through interviews to describe the most usual misconceptions the students made during the instruction of the course. Their answers were recorded and classified, producing a draft version of the buggy rules. In the second round of interviews, the experts were asked to update and/or confirm the rules. This process was repeated one more time. After that, the final version of the buggy rules was produced, including 96 potential misconceptions.

The diagnostic mechanism utilizes a repository of tests associated with the course lessons. Each test consists of a set of questions; each of which is related to a certain concept of the lesson. Every question's answer is characterized by the degree of student carelessness ranging from 0 (indicating a possible knowledge gap) to 1 (suggesting a choice by mistake) and the buggy rule explaining the student misconception. Thus, when students give an incorrect answer, the system can detect if there is a misconception and in which part of the lesson. Every question item of a test has an alternative one referred to the same concept. Hence, when a student, taking a test, gives a wrong answer that has a high degree of carelessness, the system delivers the alternative question. If the student answers this question correctly, then the system supposes that the first wrong answer was due to student carelessness. In this case, the mistake is not calculated to the final score and the system just informs the student to be more careful. If the student answers incorrectly in the second chance he/she has, then the system assumes that there is a knowledge gap on the concept to which the questions are referred, regardless of their degree of student carelessness. Moreover, the system considers the final score and the completion time of the test in order to provide to students a full report of hints for improving their learning outcomes. Figure 1 illustrates the entity–relationship model of the diagnostic mechanism. The algorithmic representation of diagnostic mechanism is shown in Algorithm 1.

Algorithm 1 Diagnostic Mechanism

1: student test time = 0
2: mistakes = 0
3: start time = time
4: **do**
5:     Display question(test)
6:     Get answer
7:     **if** is in correct(answer) AND degree of carelessness (answer) < 0.5 **then**
8:         Display alternative question in same concept(test, question)
9:         **if** is correct (answer) **then**
10:             Print "Be careful with your answers!"
11:         **else**
12:             Get concept, buggy rule related to answer
13:             Store concept, buggy rule in student profile
14:             mistakes + = 1
15:         **endif**
16:     **endif**
17:     **if** last question (test) **then**
18:         Student test time = time—start time
19:     **endif**
20: **until** student test time <> 0
21: student test score = Calculate score (mistakes)
22: Print report on score (student test score)
23: Print report on test duration(student test time)
24: Print report on concepts(student profile)
25: Print report on bugs(student profile)
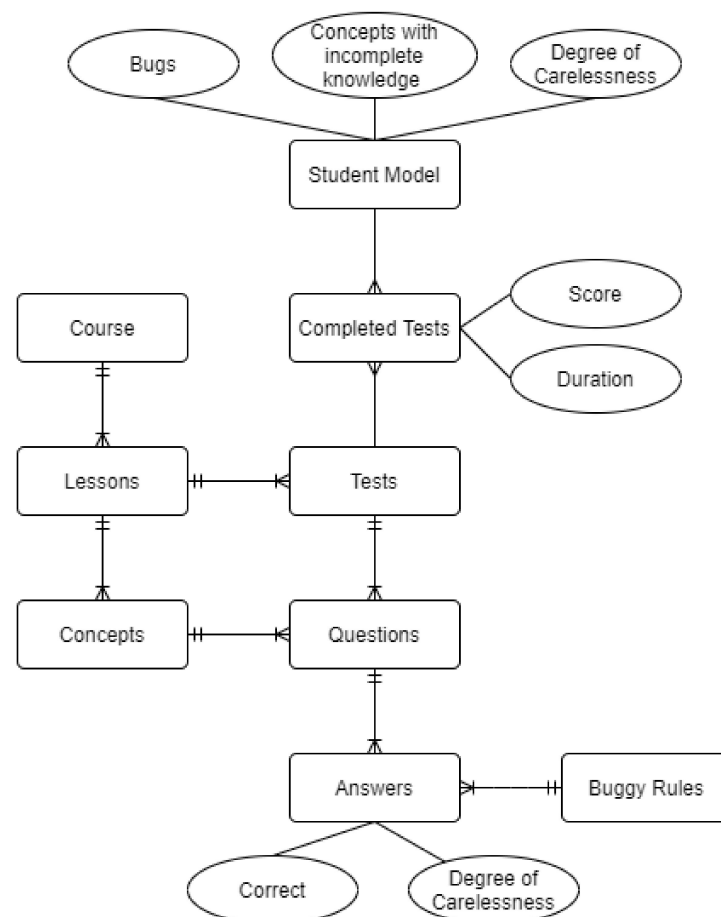
Figure 1. Entity-Relationship model.

At the end of a test, the system provides personalized guidance, which constitutes the optimal learning path that can lead the student to improve his/her performance. This descriptive feedback reports on the following:

- Success rate on the test, giving a corresponding motivation message.
- Time taken for completing the test, which is compared to the average time all students needed to fill in the test.
- Concepts in which the student had made a mistake that indicated a misconception.
- The misconceptions detected by the diagnostic mechanism.

Table 1 depicts the structure and rationale behind the personalized guidance.

Table 1. The structure and rationale of the personalized guidance.

| Score Feedback	Score < 50%	50% ≤ Score < 70%	70% ≤ Score < 85%	Score ≥ 85%
Icon	🙁	😐	🙂	😃
Motivation message on the score	You have made many mistakes. You must study the lesson again from scratch to be better prepared for the test. Your score is xx%	You are close to success. Study harder to improve your skills. Your score is xx%	Bravo! You are very good. Keep up the good work. Your score is xx%	Congratulations! Excellent job. Continue like this. Your score is xx%
Comment on test duration	Average duration of all student to complete the test < Completion time of the student: The test was completed in Xm Xs. This duration is greater than the average one. Try to be more confident of your answers. Average duration of all student to complete the test ≥ Completion time of the student: The test was completed in Xm Xs. This duration reflects a satisfactory completion of the test.			
Lesson's Concepts	The system recommends to student to study again the sub-units of the lesson where was detected a bug.			
Student Bugs	The systems delivers the detected misconceptions according to the buggy rule library.			

3. Examples of Operation

The course that has been chosen for learning through the presented system is the HTML language. The reason for this choice is that although this language can be characterized as quite easy to learn and to use, it has many peculiarities emerging from the pages' structure and plenty of tags. The HTML elements are blocks, namely tags, written using angle brackets, and may include other tags as sub-elements. Moreover, each element may consist of a number of attributes related to the type of tag. These markups may make the understanding of the language difficult, leading to several misconceptions. These bugs in student cognitive state were documented by the experts, based on their experience in teaching the HTML language. A sample of the buggy rules emerged from this process is illustrated in Table 2.

Table 2. A sample of buggy rules.

	Buggy Rules
1	You have misunderstood the tag "<" with "#".
2	You have confused the body section with the head section.
3	You are confused about the i tag and the b tag.
4	You have misunderstood the attribute face of font tag.
5	You have confused the p tag with the paragraph tag.
6	You have confused the tag with the tag.
7	You are confused about the start attribute and the type attribute of tag.
8	You are confused about the tag.

In order to better understand the functionality of the diagnostic module and the adaptive feedback delivered, an example of operation is provided comparing the interaction of two users with the system. In particular, Student A and Student B took the third Test which corresponds to the "HTML Lists" lesson. Figures 2 and 3 illustrates their results on the test. Both students reached a score of 68%; however, they received different feedback. The system stimulates Student A to study further the "Ordered Lists" and suggests that it might be useful a revision on <li> tag. Student A bugs concern the <ul> and <ol> tags, as well as the start and type attributes of <ol> tag. On the other hand, the system recommends that Student B should study "Nested Lists" and "Description Lists"; while his misconceptions refer to the <dl>, <dt>, and <dd> tags, as well as the elements that can be included in <li> tag.
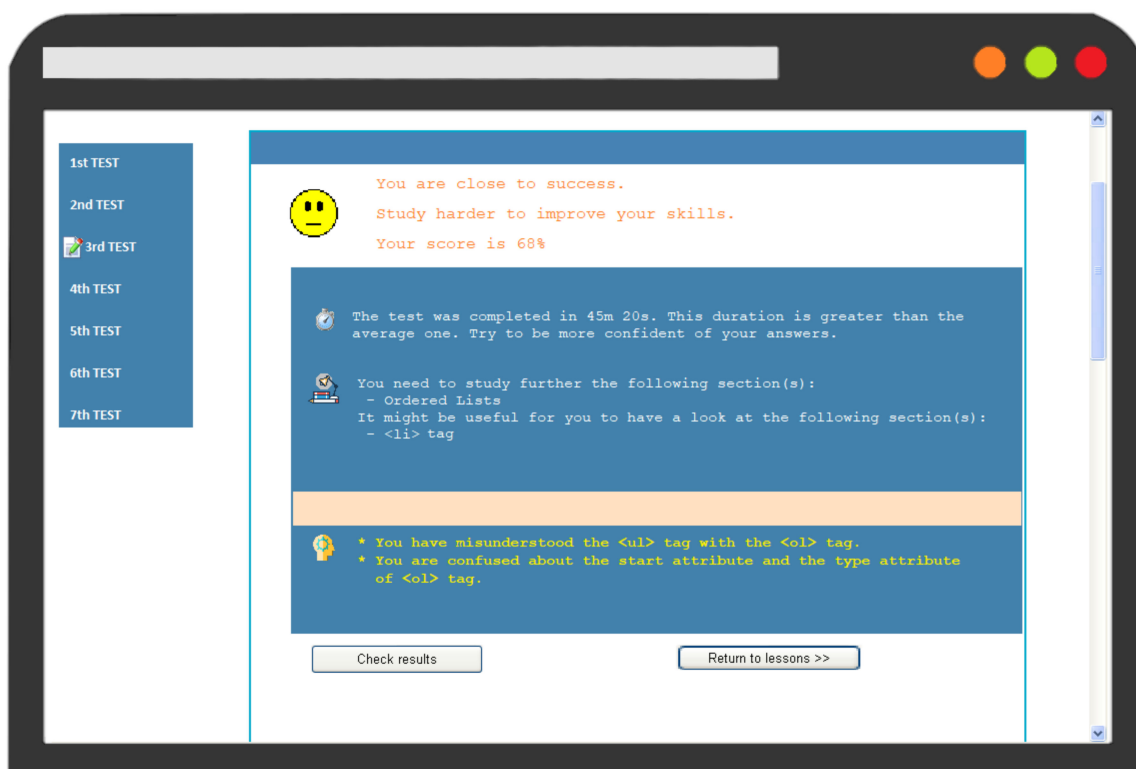
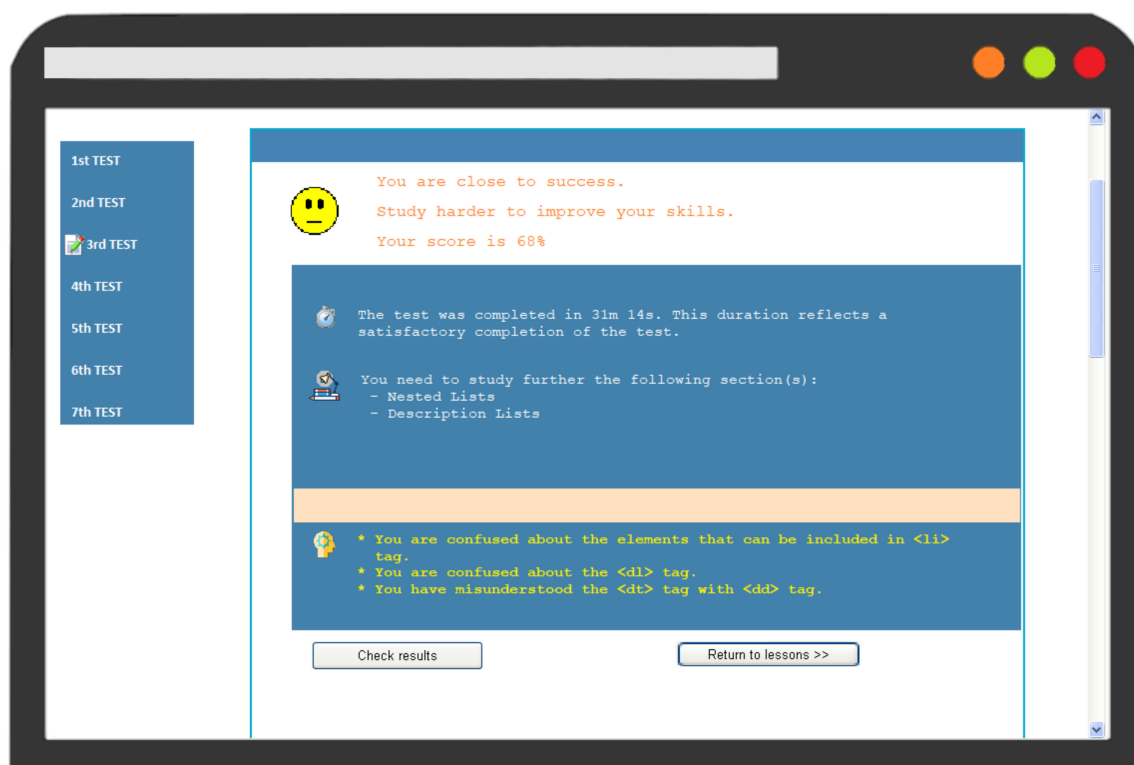**Figure 2.** Feedback to Student A on third Test.

Figure 3. Feedback to Student B on third Test.

The reason why the reports to students are different is that although they made the same number of mistakes, they either gave different incorrect answers on the same question or made mistakes in different questions, referred to different sections of the lessons. As such, the system diagnosed different misconceptions and incomplete knowledge in sections for each student, providing individualized guidance regarding the learning path that should be followed. Moreover, it informed them about the bugs detected, helping them on their better handling.

4. System Evaluation

For evaluating the presented system, 80 undergraduate computer science students at a public university in Greece participated. Students' age ranged from 20 to 21 years old, having approximately equal computer skills and knowledge, as all of them were at the third year of their studies. Moreover, the sample consists of 44 (55%) male and 36 (45%) female students. The students were separated into two groups of 40 members with the same number of males and females, i.e., Group A and Group B.

The evaluation process took place during the tutoring of the "Web Programming" course, for a semester. In particular, Group A was taught the section concerned the "HTML Language" solely using the presented personalized e-learning system; while Group B used a conventional one without diagnosis of student bugs or personalization for this purpose. The reason of using the conventional system in the evaluation process is for assessing the potential of the cognitive diagnosis and personalized guidance used in our system in comparison to conventional approaches. All the students reacted passively to this new learning experience, completing successfully the required tasks/tests without dropouts.

Firstly, the system evaluation pertains to three dimensions, namely the user experience, the effectiveness of personalization and the impact on student learning [34]. Hence, a 10-point Likert scale questionnaire was conducted, including three questions for the assessment of the two first dimensions and four questions for the last dimension (Table 3). The questionnaire was delivered to students after the completion of the course and all of them answered it.

Table 3. Questionnaire of system evaluation.

Dimension		Questions
User Experience	1	Rate the user interface of the system. (1–10)
	2	Rate your learning experience. (1–10)
	3	Did you like the interaction with the system? (1–10)
Effectiveness of personalization	4	Did the system detect appropriately your misconceptions? (1–10)
	5	Rate the way the personalized guidance was presented. (1–10)
	6	Rate the learning content relevance to your personal profile. (1–10)
Impact on Learning	7	Would you like to use this platform in other courses as well? (1–10)
	8	Did you find the software helpful for your lesson? (1–10)
	9	Would you suggest the software to your friends to use it? (1–10)
	10	Rate the easiness in interacting with the software. (1–10)

The 10-point Likert scale answers were converted into three categories, namely Low ranging from 1 to 3, Average ranging from 4 to 7, and High ranging from 8 to 10; and, they were aggregated in the three dimensions. Figure 4 illustrates the evaluation results of Group A, concerned its interaction with the presented system, in comparison with Group B, which was interacting with the conventional system. The results reveal that the presented system is superior to the conventional one, regarding the three dimensions of the evaluation.
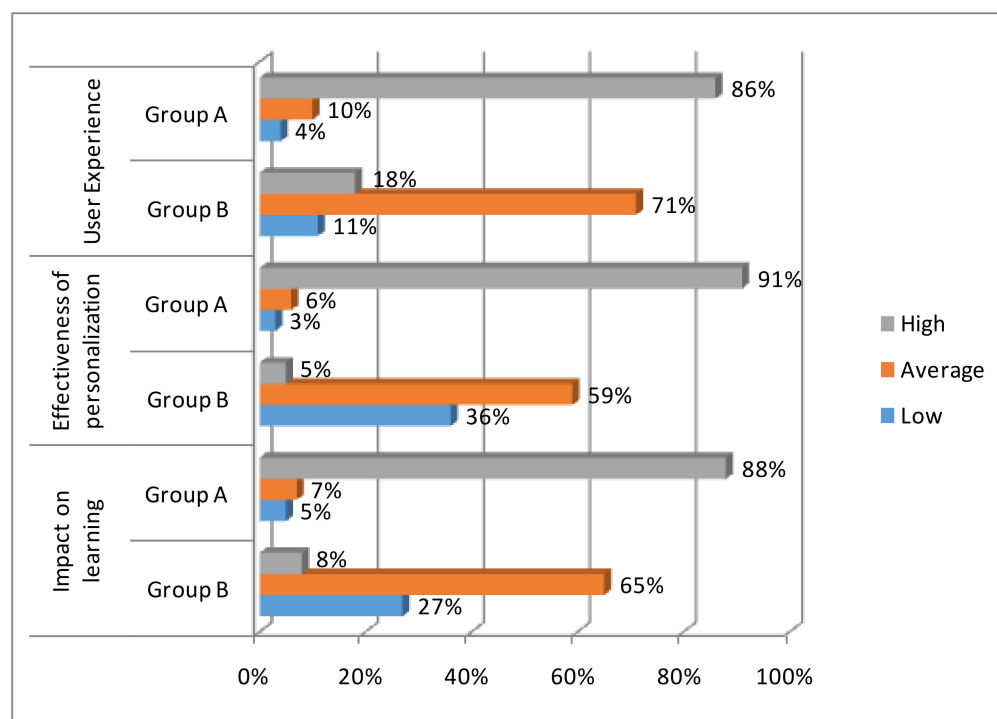
Figure 4. Evaluation results.

The "User Experience" of the presented system had 86% of high rating, indicating that students had a positive learning experience; whereas, the conventional one was rated 18% for high scores. In addition, 91% of Group A students declared the personalization mechanism was extremely helpful in establishing a learner-centered environment, as emerged by the results of the category "Effectiveness of Personalization". On the other hand, the low rating of Group B in this category indicates the lack of personalization in the conventional system. Finally, unlike the conventional version, the findings for the category "Impact on Learning" of the presented system are quite encouraging, demonstrating a 88% success rate for our software's pedagogical affordance. Analyzing the results of

the evaluation research, there is strong evidence that our presented method can further improve the adaptivity and personalization of e-learning software by incorporating error diagnosis mechanisms, laying the groundwork for more individualized tutoring systems.

Secondly, in order to determine whether the personalization mechanisms used in the presented system have an effect on students comparing to conventional systems, the two-sample *t*-test between Group A and Group B was applied in questions 4–6.

Based on the *t*-test findings (Table 4), it can be concluded that there is a statistically significant difference between the means of the two trials when it comes to the aforementioned questions (Q4, Q5, Q6). In further detail, the system used by Group A was found to detect significantly more appropriately students' misconceptions than the conventional one used by Group B (Q4: t(39)= 16.19, p <0.05). Moreover, there was a significant difference in the rating of the personalized guidance for Group A (Mean: 8.73, Variance: 2.72) and Group B (Mean: 5.45, Variance: 1.38), where t(39) = 10.78 and $p = 2.92 \times 10^{-13}$ (Q5); as well as, in the relevance of learning content delivered for Group A (Mean: 8.48, Variance: 3.18) and Group B (Mean: 5.6, Variance: 1.89), where t(39) = 10.81 and $p = 2.7 \times 10^{-13}$ (Q6). These results suggest that the proposed system outperforms its conventional version in terms of appropriate detection of learners' misconceptions, personalized guidance and delivery of learning content. These outcomes were anticipated, given that the presented system, based on buggy model and Repair theory, provides tailored feedback to students about their misconceptions and the actions they should take in order to improve them. As a result, a student-centered learning environment is provided, with enhanced knowledge acquisition and learning outcomes. On the other hand, the conventional system only delivers the final score, lacking in descriptive analysis of test results and thus, students are totally helpless regarding the learning path they should follow.

Table 4. *T*-test results on Q4, Q5, and Q6.

	Q4		Q5		Q6	
	Group A	**Group B**	**Group A**	**Group B**	**Group A**	**Group B**
Mean	8.65	5.23	8.73	5.45	8.48	5.6
Variance	3.36	2.18	2.72	1.38	3.18	1.89
Observations	40	40	40	40	40	40
Pooled Variance	0.69		0.11		0.46	
Hypothesized Mean Difference	0		0		0	
Degree of Freedom	39		39		39	
t Stat	16.19		10.78		10.81	
P(T $\leq$ t) two-tail	6.58×10^{-19}		2.92×10^{-13}		2.7×10^{-13}	
t Critical two-tail	2.023		2.023		2.023	

Finally, for evaluating the leaning outcomes, the final score of the course, emerged from the average of all chapters' tests, was calculated for each student in Group A and Group B, and the two-sample *t*-test between the final scores of these two groups was applied. The goal of this experiment is to investigate whether the students who used the presented system achieved higher performance than those who used the conventional version.

Analyzing the *t*-test results on learning outcomes (Table 5), it can be observed that there is a statistically significant difference between the means of the two groups. In particular, students who used the presented system had higher final scores (Mean: 82.45, Variance: 167.99) than did those using the conventional one (Mean: 70.05, Variance: 170.05), t(78) = 4.27 and $p = 5.55 \times 10^{-5}$. These results suggest that the approaches used for detecting students' misconceptions and for providing the tailored descriptive feedback can enhance learning process and lead students to achieve higher performance.

Table 5. *T*-test results on learning outcomes.

	Learning Outcomes	
	Group A	Group B
Mean	82.45	70.05
Variance	167.99	170.05
Observations	40	40
Pooled Variance	169.02	
Hypothesized Mean Difference	0	
Degree of freedom	78	
t Stat	4.27	
P(T ≤ t) two-tail	5.55×10^{-5}	
t Critical two-tail	1.99	

5. Conclusions

This paper describes a novel cognitive diagnostic module that has been included in an e-learning program for HTML instruction. The technology is responsible for identifying the learners' cognitive flaws and providing tailored instruction. This approach is unique in that it is based on the Repair theory and incorporates additional features into its diagnostic mechanism, such as student negligence and test completion times; it also employs a recommender module that suggests students optimal learning paths based on their misconceptions using descriptive test feedback, as well as the flexibility of learning materials. Using the Repair theory, the diagnostic mechanism explains the source of errors, noticed by the student during the assessment. This library covers typical blunders, effectively generating a hypothesis space. As a result, the test items are enlarged to include the hypothesis space. The buggy rules were developed by a group of computer science academics with a great experience in teaching HTML. They were specifically asked through interviews to characterize the most common misconceptions that learners had, during the course's training.

Our approach was fully evaluated using a well-known model and student's *t*-test. The results are very promising, showing that the system assisted students in a high degree to better understand their misconceptions. Based on the evaluation results, our approach was reported to have a positive impact on learning, to create a personalized learning environment for students and offer an optimal user experience.

Future work includes the extension of the buggy modeling so that the e-learning software can cope with different misconceptions and reasons of learners' mistakes. Moreover, future research plans include the alteration of recommendations to learners in terms of level of detail. Finally, part of our future work is to further evaluate the efficiency and acceptance of our system using qualitative techniques, such as interviews, and additional quantitative ones, such as pretest-posttest design.

Data Availability Statement: The data used to support the findings of this study have not been made available because they contain information that could compromise research participant privacy/consent.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Somyürek, S. The new trends in adaptive educational hypermedia systems. *Int. Rev. Res. Open Distrib. Learn.* **2015**, *16*. [CrossRef]
2. Somyürek, S.; Brusilovsky, P.; Guerra, J. Supporting knowledge monitoring ability: Open learner modeling vs. open social learner modeling. *Res. Pract. Technol. Enhanc. Learn.* **2020**, *15*, 1–24. [CrossRef]
3. Troussas, C.; Krouska, A.; Sgouropoulou, C. Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education. *Comput. Educ.* **2020**, *144*, 103698. [CrossRef]
4. Cuong, B.C.; Lich, N.T.; Ha, D.T. Combining Fuzzy Set—Simple Additive Weight and Comparing with Grey Relational Analysis For Student's Competency Assessment in the Industrial 4.0. In Proceedings of the 2018 10th International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City, Vietnam, 1–3 November 2018; pp. 294–299.
5. Saad, M.B.; Jackowska-Strumillo, L.; Bieniecki, W. ANN Based Evaluation of Student's Answers in E-tests. In Proceedings of the 2018 11th International Conference on Human System Interaction (HSI), Gdansk, Poland, 4–6 July 2018; pp. 155–161.
6. Troussas, C.; Krouska, A.; Virvou, M. A multilayer inference engine for individualized tutoring model: Adapting learning material and its granularity. *Neural Comput. Appl.* **2021**, 1–15. [CrossRef]
7. Troussas, C.; Krouska, A.; Sgouropoulou, C. Improving Learner-Computer Interaction through Intelligent Learning Material Delivery Using Instructional Design Modeling. *Entropy* **2021**, *23*, 668. [CrossRef] [PubMed]
8. Huang, Y.-M.; Hsieh, M.Y.; Usak, M. A Multi-Criteria Study of Decision-Making Proficiency in Student's Employability for Multidisciplinary Curriculums. *Mathematics* **2020**, *8*, 897. [CrossRef]
9. Stansfield, J.C.; Carr, B.; Goldstein, I.P. *Wumpus Advisor I: A First Implementation of a Program that Tutors Logical and Probabilistic Reasoning Skills*; Massachusetts Institute of Technology: Cambridge, MA, USA, 1976.
10. Martins, A.C.; Faria, L.; de Carvalho, C.V.; Carrapatoso, E. User modeling in adaptive hypermedia educational systems. *Educ. Technol. Soc.* **2008**, *11*, 194–207.
11. Liu, Z.; Wang, H. A Modeling Method Based on Bayesian Networks in Intelligent Tutoring System. In Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design, Melbourne, Australia, 26–28 April 2007; pp. 967–972.
12. Qodad, A.; Benyoussef, A.; El Kenz, A.; Elyadari, M. Toward an Adaptive Educational Hypermedia System (AEHS-JS) based on the Overlay Modeling and Felder and Silverman's Learning Styles Model for Job Seekers. *Int. J. Emerg. Technol. Learn.* **2020**, *15*, 235–254. [CrossRef]
13. Zhao, J.; Li, M.; Liu, W.; Li, S.; Lin, Z. Detection of Chinese Grammatical Errors with Context Representation. In Proceedings of the 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), Guiyang, China, 22–24 August 2018; pp. 25–29.
14. Lin, X.; Ge, S.; Song, R. Error analysis of Chinese-English machine translation on the clause-complex level. In Proceedings of the 2017 International Conference on Asian Language Processing (IALP), Singapore, 5–7 December 2017; pp. 185–188.
15. Lee, L.-H.; Chang, L.-P.; Tseng, Y.-H. Developing learner corpus annotation for Chinese grammatical errors. In Proceedings of the 2016 International Conference on Asian Language Processing (IALP), Tainan, Taiwan, 21–23 November 2016; pp. 254–257.
16. Haridas, M.; Vasudevan, N.; Nair, G.J.; Gutjahr, G.; Raman, R.; Nedungadi, P. Spelling Errors by Normal and Poor Readers in a Bilingual Malayalam-English Dyslexia Screening Test. In Proceedings of the 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), Mumbai, India, 9–13 July 2018; pp. 340–344.
17. Troussas, C.; Chrysafiadi, K.; Virvou, M. Machine Learning and Fuzzy Logic Techniques for Personalized Tutoring of Foreign Languages. In Proceedings of the International Conference on Artificial Intelligence in Education, London, UK, 27–30 June 2018; pp. 358–362.
18. Khodeir, N.A. Constraint-based and Fuzzy Logic Student Modeling for Arabic Grammar. *Int. J. Comput. Sci. Inf. Technol.* **2020**, *12*, 35–53. [CrossRef]
19. Li, S.; Zhao, J.; Shi, G.; Tan, Y.; Xu, H.; Chen, G.; Lan, H.; Lin, Z. Chinese Grammatical Error Correction Based on Convolutional Sequence to Sequence Model. *IEEE Access* **2019**, *7*, 72905–72913. [CrossRef]
20. Henley, A.Z.; Ball, J.; Klein, B.; Rutter, A.; Lee, D. An Inquisitive Code Editor for Addressing Novice Programmers' Misconceptions of Program Behavior. In Proceedings of the 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), Madrid, Spain, 25–28 May 2021; pp. 165–170.
21. Lai, A.F.; Wu, T.T.; Lee, G.Y.; Lai, H.Y. Developing a web-based simulation-based learning system for enhancing concepts of linked-list structures in data structures curriculum. In Proceedings of the 2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), Kota Kinabalu, Malaysia, 2–4 December 2015; pp. 185–188.
22. Chang, J.-C.; Li, S.-C.; Chang, A.; Chang, M. A SCORM/IMS Compliance Online Test and Diagnosis System. In Proceedings of the 2006 7th International Conference on Information Technology Based Higher Education and Training, Ultimo, Australia, 10–13 July 2006; pp. 343–352.

23. Barker, S.; Douglas, P. An intelligent tutoring system for program semantics. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), Las Vegas, NV, USA, 4–6 April 2005; Volume 1, pp. 482–487.

24. Khalife, J. Threshold for the introduction of programming: Providing learners with a simple computer model. In Proceedings of the 28th International Conference on Information Technology Interfaces, Cavtat, Croatia, 19–22 June 2006; pp. 71–76.

25. Troussas, C.; Krouska, A.; Virvou, M. Injecting intelligence into learning management systems: The case of adaptive grain-size instruction. In Proceedings of the 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 15–17 July 2019; pp. 1–6.

26. Krugel, J.; Hubwieser, P.; Goedicke, M.; Striewe, M.; Talbot, M.; Olbricht, C.; Schypula, M.; Zettler, S. Automated Measurement of Competencies and Generation of Feedback in Object-Oriented Programming Courses. In Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal, 27–30 April 2020; pp. 329–338.

27. Almeda, M. Predicting Student Participation in STEM Careers: The Role of Affect and Engagement during Middle School. *J. Educ. Data Min.* **2020**, *12*, 33–47.

28. Sumartini, T.S.; Priatna, N. Identify student mathematical understanding ability through direct learning model. *J. Phys. Conf. Ser.* **2018**, *1132*, 012043. [CrossRef]

29. Troussas, C.; Krouska, A.; Sgouropoulou, C. A Novel Teaching Strategy Through Adaptive Learning Activities for Computer Programming. *IEEE Trans. Educ.* **2021**, *64*, 103–109. [CrossRef]

30. Brown, J.; VanLehn, K. Repair Theory: A Generative Theory of Bugs in Procedural Skills. *Cogn. Sci.* **1980**, *4*, 379–426. [CrossRef]

31. Brown, J.S.; Burton, R.R. Diagnostic models for procedural bugs in basic mathematical skills. *Cogn. Sci.* **1978**, *2*, 155–192. [CrossRef]

32. Rashid, T.; Asghar, H.M. Technology use, self-directed learning, student engagement and academic performance: Examining the interrelations. *Comput. Hum. Behav.* **2016**, *63*, 604–612. [CrossRef]

33. Krouska, A.; Troussas, C.; Sgouropoulou, C. Fuzzy Logic for Refining the Evaluation of Learners' Performance in Online Engineering Education. *Eur. J. Eng. Res. Sci.* **2019**, *4*, 50–56. [CrossRef]

34. Alepis, E.; Troussas, C. M-learning programming platform: Evaluation in elementary schools. *Informatica* **2017**, *41*, 471–478.