

Article

Improving Learner-Computer Interaction through Intelligent Learning Material Delivery Using Instructional Design Modeling

Christos Troussas , Akrivi Krouska  and Cleo Sgouropoulou

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

* Correspondence: ctrouss@uniwa.gr

Abstract: This paper describes an innovative and sophisticated approach for improving learner-computer interaction in the tutoring of Java programming through the delivery of adequate learning material to learners. To achieve this, an instructional theory and intelligent techniques are combined, namely the Component Display Theory along with content-based filtering and multiple-criteria decision analysis, with the intention of providing personalized learning material and thus, improving student interaction. Until now, the majority of the research efforts mainly focus on adapting the presentation of learning material based on students' characteristics. As such, there is free space for researching issues like delivering the appropriate type of learning material, in order to maintain the pedagogical affordance of the educational software. The blending of instructional design theories and sophisticated techniques can offer a more personalized and adaptive learning experience to learners of computer programming. The paper presents a fully operating intelligent educational software. It merges pedagogical and technological approaches for sophisticated learning material delivery to students. Moreover, it was used by undergraduate university students to learn Java programming for a semester during the COVID-19 lockdown. The findings of the evaluation showed that the presented way for delivering the Java learning material surpassed other approaches incorporating merely instructional models or intelligent tools, in terms of satisfaction and knowledge acquisition.

Keywords: adaptive learning material delivery; Component Display Theory; content-based filtering; Intelligent Tutoring Systems; Multiple-Criteria Decision Analysis; online learning; Weighted Sum Model



Citation: Troussas, C.; Krouska, A.; Sgouropoulou, C. Improving Learner-Computer Interaction through Intelligent Learning Material Delivery Using Instructional Design Modeling. *Entropy* **2021**, *23*, 668. <https://doi.org/10.3390/e23060668>

Academic Editor: Salim Lahmiri

Received: 5 May 2021
Accepted: 25 May 2021
Published: 26 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The field of education has undergone tremendous changes during the last decades. Indeed, the incorporation of technology means has altered the way with which learners gain knowledge. Furthermore, injecting artificial intelligence into learning technology systems can offer a more personalized learning experience to students [1]. With these developments, digital learning environments can be better tailored to learners' personal and cognitive interests and preferences. Especially during the COVID-19 pandemic, the need for student-centric e-learning software is even more imperative, since face-to-face education has been fully replaced by digital learning [2]. Such urgent needs encompass the amelioration of the learning material and its adaptation to the learners so that it can be aligned to their special educational needs.

Adaptivity in the learning material embraces considerable pedagogical potential since it promotes the acquisition of knowledge in an environment that respects the cognitive characteristics of learners [3–5]. As such, learners can receive the educational material in a more accurate row and a more appropriate volume. According to the specific needs of each learner, the domain to be delivered can be tailored to these needs [6–8]. For instance, learners, who have poor knowledge on a topic, probably want to receive material sequentially with a high level of granularity; while learners with an advanced level of

knowledge, probably want to see more superficially topics that are well known for them. During the COVID-19 pandemic, the experience of tutors in providing specific learning material to learners cannot be exploited due to the absence of face-to-face interaction, and thus, it is significant to use technological means to cover this issue.

The delivery of adaptive learning material to learners is a topic of growing interest by researchers in the related literature [9–11]. Several research efforts have been made towards adapting the domain knowledge to students in view of enhancing the pedagogical potential of the e-learning software. For example, researchers have explored the delivery of learning material to students by taking into account their preferences, namely their learning styles [12–14]. However, this adaptation is targeted mainly to the way of the domain knowledge presentation using learning styles (e.g., material for visual or auditory learners, etc.), and not to the type of activities delivered to the students. There have been several research efforts that explored the adaptivity of the domain knowledge to students in terms of their remembering skills [15–17], knowledge acquisition ability [18], perceiving facts [19–21], incentive and appeal [22–24], as well as the tutoring routes [25–27], domain knowledge style [28,29], external resources recommendation [30,31] and sequential teaching material [32–34]. Concerning the technology to achieve adaptation of the learning material, the researches in the afore-presented literature utilized discrete-time stochastic control process, machine learning, weighted conceptual diagrams, ontology learning and supervised and unsupervised techniques [35,36].

In conclusion, according to a review work of 2020 [37], the field of delivering adaptive learning material to learners is a quite under-researched area and thus, there is significant room for improvement in this direction. The motivation for this research derived from the need to explore the research area of adaptive domain knowledge. The majority of the research efforts inadequately explore the presentation of adaptive learning material. As such, there is free research space for researching issues like the type of learning material in order to maintain the pedagogical affordance of the system. Furthermore, the exploration of the utilization of novel technologies to enhance adaptation, like Multiple-Criteria Decision Analysis (MCDA) and Weighted Sum Model (WSM), triggered this research.

The novelty of the manuscript lies in organizing learning material based on Component Display Theory (CDT) and sophisticatedly adapting it to university students of Java programming language. The learning theory of CDT concentrates on the cognitive domain and deals only with the micro-level strategies for teaching a concept, principle, procedure, etc. providing detailed instructional prescriptions about how to support specific instructional outcomes with the appropriate material. As such, this theory in conjunction with the adaptation of the material can lead the effectiveness of learning to be improved. The learning material is stored in a digital repository being characterized by metadata, such as required knowledge level, misconceptions that it can correct, and CDT level. Using content-based filtering and MCDA, and specifically WSM, the system provides adequate learning material to learners. Content-based filtering is one of the common methods in recommending items similar to user characteristics; whereas, WSM was used since it is one of the most accurate methods for evaluating a number of alternatives in terms of a number of decision criteria. The weights of the WSM are determined by experts regarding the learning goals set.

2. Applying CDT to Support Learning

The domain knowledge of the presented system is the computer programming concepts, and specifically the programming language Java. It consists of 12 chapters, ranging from preliminary to advanced topics.

The modeling of the domain knowledge material is made using the Component Display Theory [38], tailored to determined performance levels. Hence, particular strategies are used orchestrating the learning of a single concept using components such as definitions, examples, and practice exercises [39]. CDT organizes learning into two dimensions:

1. **Content dimension.** It refers to the information that should be learned. Content ranges from simple forms, namely facts, to more complex ones, namely principles. The content in CDT includes the following types:
 - Facts—information proved to be true.
 - Concepts—general notions about a particular subject.
 - Procedures—a series of actions that should be performed in a certain manner in order to solve a problem or accomplish a goal.
 - Principles—the rules or assumptions that describe how something happens or works.
2. **Performance dimension.** It refers to the abilities of the learners to apply the content. Performance ranges from the simplest form, namely remembering, to an advanced form, namely finding. The three types of performance are:
 - Remembering—the memorization of the information and its recall.
 - Using—the ability of the learner to apply the information to a particular context.
 - Finding—the learner constructs new knowledge based on the content.
 CDT determines four primary presentation forms:
 - Rules, an expository presentation of general concepts.
 - Examples, an expository presentation of cases related to a concept.
 - Recall, the inquisition of general concepts.
 - Practice, the inquisition of cases.

It also includes some secondary presentation forms, such as prerequisites (other concepts required to have been studied before), objectives (the learning goals intended to be attained), help (useful material for the better understanding of the concept), mnemonics (memory aids for remembering the concept) and feedback (analysis of the interaction between learning goals and components of the concept).

To decide the level of performance required for an area of content, the matrix is set up. In CDT, it can be concluded for each of the groups in the matrix that there is a mixture of primary and secondary presentation types that will offer the most effective and efficient acquisition of available skills and information. CDT specifies that when it includes both the required primary and secondary types, instruction is more effective. A complete instruction session must then consist of a goal, accompanied by a mixture of rules, examples, recall, practice, feedback, helps, and mnemonics relevant to the subject and learning task.

The theory is mainly intended for use by groups of students. Several components are given so that a wide range of learners can participate, but learners need only the components that function best with them to accomplish the instructional objectives (Table 1).

Table 1. Applying CDT to Java learning.

	Fact	Concept	Procedure	Principle
Use	-	Identify or classify Java objects, methods, etc.	Demonstrate programming procedures	Explain why a program is running or predict the output of the program
Find	-	State or define terms, e.g., class, object, etc.	State steps	State relationship inside a program
Remember	Recall or reorganize parts of the program	Recall or reorganize definitions	Recall or reorganize steps to build a program	Recall or reorganize principles of a program

3. Adaptation of Learning Units

Adapting learning material to students' needs is crucial for providing a more effective and efficient learning process. As such, the system, developed in this study, aims to improve student performance and engagement by selecting the most proper learning units based on student characteristics using content-based filtering and delivering them according to the learning goals set by employing WSM (Algorithm 1).

The learner characteristics used in the developed algorithm concern learner cognitive level and skills, which are:

- Knowledge level (SP.KLe), emerged from the total score of student tests on a 100-point scale.
- Prior knowledge (SP.PKLe) on related domain concepts on a 100-point scale, as a result of the pre-test given to students at the beginning of the course.
- Degree of misconceptions (SP.DoM), namely a degree ranging from 0 to 1 indicating the type of mistakes the student usually does in the tests. Since the course taught is a programming language, the possible mistakes that can be done are syntactic or logical. The nearer the degree of SP.DoM is to zero, the more often a student does syntactic errors; whereas, the nearer this degree is to 1, the more often the student makes logical mistakes.
- Student performance on a 100-point scale in CDT levels, arisen from the answers on question items of the tests given; namely, student performance on the following levels:
 - Use-Concept (SP.UCon)
 - Use-Procedure (SP.UPro)
 - Use-Principle (SP.UPri)
 - Find-Concept (SP.FCon)
 - Find-Procedure (SP.FPro)
 - Find-Principle (SP.FPri)
 - Remember-Fact (SP.ReFa)
 - Remember-Concept (SP.ReCon)
 - Remember-Procedure (SP.RePro)
 - Remember-Principle (SP.RePri)

The values of the above variables, except SP.PKLe, are defined dynamically by the adaptive educational system at each interaction of the learner with the system.

Regarding the learning units, the instructors have to define the following metadata when they insert them into the repository, essential for the content adaptation:

- The knowledge level that the learning units concern (LU.KL), stated as a score on a 100-point scale.
- The learner's previous knowledge (LU.PKLe), expressed on a 100-point scale, which is a prerequisite for studying the learning unit.
- The degree of misconceptions (LU.DoM), a number between 0 to 1, indicating whether the particular learning unit is suitable for learners that make syntax (near to 0) or logic mistakes (near to 1).
- The degree, stated as performance on a 100-point scale, that the particular learning unit is suitable for each CDT level. The CDT levels are the same with those described in student performance; as such, the following names are given for the CDT levels referred to learning units: LU.UCon, LU.UPro, LU.UPri, LU.FCon, LU.FPro, LU.FPri, LU.ReFa, LU.ReCon, LU.RePro, LU.RePri.

Algorithm 1. Process for intelligent selection and delivery of learning units.

h: the number of features (student characteristics/metadata), namely 13.

S: students enrolled into the course.

S(i): the vector with the values of the 13 characteristics of student i.

LU: the set of learning units that are included into the repository.

LU(j): the vector with the values of the 13 metadata of learning unit j.

$RU \subseteq LU$: the set of recommended learning units based on student characteristics.

RU(k): the vector with the values of the 13 metadata of recommended learning unit k.

n: the number of learning units in RU.

D: the set of calculated distances for each learning unit, used in content-based filtering.

w: the vector with the 13 relative weights of importance of the criterion that is associated with the corresponding metadata.

1. Experts define the values of n and w .
2. The student i searches for learning material.
3. The system applies the content-based filtering method in LU.
 - a. Calculate distance metric based on student characteristics and learning units metadata.

$$D(i, j) = \sqrt{\sum_{c=1}^h (S_i(c) - LU_j(c))^2} \quad (1)$$
 - b. Sort D in ascending order.
 - c. Set in RU the top n learning units of D .
4. The system applies the WSM method in RU .
 - (a) Normalize the values of learning units according to beneficial and non-beneficial attributes. For beneficial attributes, where g : LU.KL, LU.PKLe and CDT levels:

$$RU_k(g) = RU_k(g) / \max(RU(g)) \quad (2)$$

For non-beneficial attributes, where g : LU.DoM:

$$RU_k(g) = \min(RU(g)) / RU_k(g) \quad (3)$$
 - (b) Calculate the weighted scores.

$$RU(k) = \sum_{c=1}^h w(c) * RU(k) \quad (4)$$
 - (c) Sort RU in descending order based on the WSM scores (if two or more learning units have the same WSM score, then they are sorted randomly).
5. Return the content of RU to the student.

4. Example of Operation

For a better understanding of the algorithm, a representative example is provided in Table 2. In this case, the experts set the WSM weights giving higher values in student knowledge level and misconceptions, as well as in complex/advanced dimensions of CDT. The reason why the weights were defined in this way is that the learning goal is for students to achieve higher-order cognitive skills. The student S_1 , selected for this case, has intermediate performance and tends to make serious mistakes, as his/her degree of misconception is high. Moreover, s/he has lower performance in complex/advanced dimensions of CDT. In this example of operation, a sample of eight learning units referred to several cognitive levels is chosen; from them, the 5 more suitable for the student, regarding his/her profile and the learning goals set, are delivered.

Table 2. Sample Data: Feature Vectors of WSM Weights, A Student & 8 Learning Units.

	Feature Vector *													
W	0.15	0.09	0.15	0.05	0.05	0.10	0.05	0.05	0.09	0.05	0.05	0.06	0.06	
S_1	67	60	0.78	65	63	58	68	64	61	71	67	66	59	
LU ₁	90	85	0.10	90	85	80	90	85	80	90	90	85	85	
LU ₂	80	80	0.30	70	70	70	80	80	80	80	80	80	80	
LU ₃	70	60	0.50	70	65	60	70	65	60	70	65	65	60	
LU ₄	75	65	0.40	70	70	65	75	75	70	80	75	70	70	
LU ₅	65	60	0.70	60	60	60	60	60	60	65	65	65	65	
LU ₆	55	45	0.85	50	50	50	55	55	50	55	55	55	50	
LU ₇	70	50	0.70	75	70	65	75	70	65	75	70	70	65	
LU ₈	65	55	0.60	65	65	60	70	65	60	75	70	65	65	

* {KLe, PKLe, DoM, UCon, UPro, UPri, FCon, FPro, FPri, ReFa, ReCon, RePro, RePri}.

In the first stage, content-based filtering is applied to sample data based on the formula shown in step 3a of the algorithm. As such, for each value of the feature vector of each learning unit, the distance from the corresponding value of the student’s feature vector is calculated, and the total distance emerges from the square root of their sum (Table 3). According to the distance values, the order of the 5 learning units selected, as being closer to student profile, is LU₃, LU₈, LU₅, LU₇, and LU₄ (Table 3).

Table 3. Applying content-based filtering in sample data.

	Calculation of $(S_i(c) - LU_j(c))^2$													Sum	Distance	Order
LU ₁	529	625	0.46	625	484	484	484	441	361	361	529	361	676	5960.46	77.21	8
LU ₂	169	400	0.23	25	49	144	144	256	361	81	169	196	441	2435.23	49.35	7
LU ₃	9	0	0.08	25	4	4	4	1	1	1	4	1	1	55.08	7.42	1
LU ₄	64	25	0.14	25	49	49	49	121	81	81	64	16	121	745.14	27.29	5
LU ₅	4	0	0.006	25	9	4	64	16	1	36	4	1	36	200.006	14.14	3
LU ₆	144	225	0.005	225	169	64	169	81	121	256	144	121	81	1800.005	42.43	6
LU ₇	9	100	0.006	100	49	49	49	36	16	16	9	16	36	485.006	22.02	4
LU ₈	4	25	0.03	0	4	4	4	1	1	16	9	1	36	105.03	10.25	2

In the next stage, WSM is employed in the selected five learning units in order to be sorted based on the learning goals defined by the WSM weights. Firstly, the values of the feature vector of each learning unit in Table 2 should be normalized following the rule described in step 4a of the algorithm. As such, the maximum value is detected for the beneficial attributes and the minimum value for the non-beneficial attributes (Table 4). It should be noted that the non-beneficial attributes are those in which minimum values are desired. Hence, the only non-beneficial attribute in the presented approach is the DoM (3rd column in the feature vector).

Table 4. Indicating the max value for beneficial features and min value for non-beneficial features.

	Feature Vector *													
LU ₃	70	60	0.50	70	65	60	70	65	60	70	65	65	60	60
LU ₈	65	55	0.60	65	65	60	70	65	60	75	70	65	65	65
LU ₅	65	60	0.70	60	60	60	60	60	60	65	65	65	65	65
LU ₇	70	50	0.70	75	70	65	75	70	65	75	70	70	65	65
LU ₄	75	65	0.40	70	70	65	75	75	70	80	75	70	70	70
	(max)	(max)	(min)					(max)	(max)	(max)	(max)		(max)	(max)

* {KLe, PKLe, DoM, UCon, UPro, UPri, FCon, FPro, FPri, ReFa, ReCon, RePro, RePri}.

Afterward, the values are normalized using the formulation x/x_{max} for beneficial attributes, and x_{min}/x for non-beneficial attributes (Table 5).

Table 5. Normalization of values.

	Feature Vector *													
LU ₃	70/75	60/65	0.40/0.50	70/75	65/70	60/65	70/75	65/75	60/70	70/80	65/75	65/70	60/70	60/70
LU ₈	65/75	55/65	0.40/0.60	65/75	65/70	60/65	70/75	65/75	60/70	75/80	70/75	65/70	65/70	65/70
LU ₅	65/75	60/65	0.40/0.70	60/75	60/70	60/65	60/75	60/75	60/70	65/80	65/75	65/70	65/70	65/70
LU ₇	70/75	50/65	0.40/0.70	75/75	70/70	65/65	75/75	70/75	65/70	75/80	70/75	70/70	65/70	65/70
LU ₄	75/75	65/65	0.40/0.40	70/75	70/70	65/65	75/75	75/75	70/70	80/80	75/75	70/70	70/70	70/70

* {KLe, PKLe, DoM, UCon, UPro, UPri, FCon, FPro, FPri, ReFa, ReCon, RePro, RePri}.

Finally, the WSM scores are calculated based on the formula presented in step 4b of the algorithm, i.e., each normalized value of Table 5 is multiplied with the respective weight of Table 2, and their sum corresponds to the WSM score (Table 6). Based on this score, the rank of the five learning units is redefined (Table 6). As a result, the learning units are delivered in the following order: LU₄, LU₃, LU₇, LU₈, and LU₅.

Table 6. Applying WSM in sample data.

	Feature Vector *													WSM	Rank
LU ₃	0.933	0.923	0.800	0.933	0.929	0.923	0.933	0.867	0.857	0.875	0.867	0.929	0.857	0.8898	2
	×	×	×	×	×	×	×	×	×	×	×	×	×		
	0.15	0.09	0.15	0.05	0.05	0.10	0.05	0.05	0.09	0.05	0.05	0.06	0.06		
LU ₈	0.867	0.864	0.667	0.867	0.929	0.923	0.933	0.867	0.857	0.938	0.933	0.929	0.929	0.8603	4
	×	×	×	×	×	×	×	×	×	×	×	×	×		
	0.15	0.09	0.15	0.05	0.05	0.10	0.05	0.05	0.09	0.05	0.05	0.06	0.06		
LU ₅	0.867	0.923	0.571	0.800	0.857	0.923	0.800	0.800	0.857	0.812	0.867	0.929	0.929	0.8265	5
	×	×	×	×	×	×	×	×	×	×	×	×	×		
	0.15	0.09	0.15	0.05	0.05	0.10	0.05	0.05	0.09	0.05	0.05	0.06	0.06		
LU ₇	0.933	0.769	0.571	1 ×	1 ×	1 ×	1 ×	0.933	0.929	0.938	0.933	1 ×	0.929	0.8844	3
	×	×	×	0.05	0.05	0.10	0.05	×	×	×	×	0.06	×		
	0.15	0.09	0.15	0.05	0.05	0.10	0.05	0.05	0.09	0.05	0.05	0.06	0.06		
LU ₄	1 ×	1 ×	1 ×	0.933	1 ×	1 ×	1 ×	1 ×	1 ×	1 ×	1 ×	1 ×	1 ×	0.9967	1
	0.15	0.09	0.15	×	0.05	0.10	0.05	0.05	0.09	0.05	0.05	0.06	0.06		
				0.05											

* {KLe, PKLe, DoM, UCon, UPro, UPri, FCon, FPro, FPri, ReFa, ReCon, RePro, RePri}.

5. Evaluation Procedure

The phase of evaluation plays a crucial role in the life cycle of educational software; this phase identifies if the system meets the initial requirements and objectives. The phase of the evaluation is oriented to the assessment of the delivery of learning units to students.

5.1. Materials and Methods

The evaluation was conducted in a computer science department of a public university in the capital city of the country. The population consists of 120 undergraduate students attending the mandatory course of the programming language Java. All the students are in the first year of their studies and, as such, they have approximately the same age and educational level. The whole process of the evaluation lasted a semester, during the COVID-19 lockdown. It needs to be noted that the course was delivered in a fully online and synchronous way, while the students used the system all this time to support their studies. The students showed great interest in the system and were supported by the two course lecturers when needed.

The students were divided into three equal groups (Groups A, B, C) by the lecturers (40 students in each group). Towards keeping a high level of quality in the evaluation process, the lecturers divided students into groups very carefully respecting their characteristics (Table 7).

Table 7. Students’ characteristics.

Features	Group A	Group B	Group C
Average age	17.9	18.2	18.1
Sex	18 females 22 males	19 females 21 males	20 females 20 males
Demographics	Equivalent number of urban students and those of rural descent.		
Technology knowledge	Advanced experience in the use of technology.		
Previous knowledge	All students passed the national exams with similar grades in order to be admitted to the university.		
Motivation	All students attended the course of Java programming and expected a high grade to be attained.		

During the evaluation process, learners in Group A (experimental group) have used a system that integrated the presented approach, namely the delivery of adaptive learning material using CDT and WSM, while students in the control groups (Groups B, C) have used

two conventional versions of the aforementioned system having the same user interface. Specifically, Group B has used a system that incorporated only the CDT theory for learning material delivery and Group C used a system that incorporated only WSM for learning material delivery.

One-way ANOVA was used to measure the satisfaction of learners of three groups concerning the delivery of adaptive learning material. Two control groups were used in order to investigate how successful the delivery of adaptive learning material is. There will be a comparison between the intelligence of the presented method (Group A) and the models of CDT (incorporated in the system used by Group B) and WSM (incorporated in the system used by Group C), individually. As such, at the end of the evaluation phase, the three groups were asked to reply to the following questions based on the Likert scale, namely from 'Very much' (10) to 'Not at all' (0):

- Was the learning material based on the level of your knowledge? (Q1)
- Was the learning material based on the level of your CDT level? (Q2)
- Did the learning material help you advance your performance? (Q3)

The ANOVA test was applied to students' answers regarding their satisfaction with the proposed learning material; hence, the null hypothesis is that there is no difference in the delivery of adaptive learning material between the experimental and control groups.

5.2. Results and Discussion

The results reveal that the null hypothesis is rejected for all three questions, since $F_{Q1} = 28.56 > F\text{-crit}_{Q1} = 3.07$, $F_{Q2} = 30.60 > F\text{-crit}_{Q2} = 3.07$ and $F_{Q3} = 28.08 > F\text{-crit}_{Q3} = 3.07$ (Tables 8 and 9). This fact shows that the means of the three groups are not all equal. However, in order to determine the group that surpasses the others, the calculation of the Least Significance Difference (LSD) is needed. In our case, the LSD gets the value of 0.56. Comparing LSD with the difference of the means of Group A-Group B and Group A-Group C for each question, the results show that the differences are greater than it. In particular, in Q1, the difference of the means of Group A and Group B is 1.475; while this of Group A and Group C is 2.35 (Q1: $\text{diff}_{AB} = 1.475 > 0.56$, $\text{diff}_{AC} = 2.35 > 0.56$). Hence, the presented system provides a more appropriate method for delivering learning units according to student knowledge than the other systems. Regarding Q2, the difference of the means of Group A and Group B is 0.8; while this of Group A and Group C is 2.32 (Q2: $\text{diff}_{AB} = 0.8 > 0.56$, $\text{diff}_{AC} = 2.32 > 0.56$). This fact indicates that the learning material provided by the presented system corresponds better to student CDT level compared to the simple approaches. Finally, concerning Q3, the difference of the means of Group A and Group B is 1.23; while this of Group A and Group C is 2.12 (Q3: $\text{diff}_{AB} = 1.23 > 0.56$, $\text{diff}_{AC} = 2.12 > 0.56$). As such, the presented system has a more positive effect on student performance than the other two systems.

Table 8. ANOVA analysis of students' feedback—Part I.

Quest.	Group	Count	Sum	Mean	Variance
Q1	Group A	40	331	8.275	1.49
	Group B	40	272	6.8	1.81
	Group C	40	237	5.925	2.64
Q2	Group A	40	332	8.3	1.70
	Group B	40	300	7.5	1.79
	Group C	40	239	5.98	1.97
Q3	Group A	40	338	8.45	1.13
	Group B	40	289	7.22	1.61
	Group C	40	253	6.33	2.12

Table 9. ANOVA analysis of students' feedback—Part II.

Quest.	Source of Variation	SS	df	MS	F	P	F-Crit
Q1	Between Groups	112.85	2	56.43	28.56	7.93×10^{-11}	3.07
	Within Groups	231.15	117	1.98			
	Total	344	119				
Q2	Between Groups	111.62	2	55.81	30.60	2.04×10^{-11}	3.07
	Within Groups	213.38	117	1.82			
	Total	325	119				
Q3	Between Groups	91.02	2	45.51	28.08	1.1×10^{-10}	3.07
	Within Groups	189.65	117	1.62			
	Total	280.67	119				

To sum up, the recommendation of learning material incorporating content-based filtering and WSM outperforms the simple approaches where only one method is used. More specifically, the content-based filtering ensures the selection of the most suitable material for student profile, while the WSM reorders the selected learning units to better fit the learning goals set. The findings support the effectiveness and efficiency of the presented system regarding the successful delivery of content based on students' knowledge level and CDT level, as well as regarding the improvement of learning outcomes.

6. Conclusions and Future Work

In this paper, an innovative approach for the delivery of adequate learning material to students is presented. To achieve this, the approach encompasses the combination of an instructional design theory and intelligent techniques. More specifically, CDT is used for organizing the learning material. It functions in conjunction with the content-based filtering and MCDA, and specifically WSM, and as such, adequate learning material is delivered to learners.

As a testbed for this research, a prototype intelligent tutoring system has been developed. The system was used for a semester for the teaching of Java programming to university students during the COVID-19 lockdown. The system has been evaluated with promising results. Specifically, it was compared to two conventional versions, incorporating merely instructional models or intelligent tools, and it was found that it surpasses them in terms of satisfaction and knowledge acquisition.

Limitations of this work include that the WSM weights of the presented model are set by the experts according to the learning goals of the students. However, this process could be automated and the weights could be set dynamically in order to take into account the learning progress and performance of the students.

Future steps include the utilization of other intelligent techniques, like artificial neural networks, in order to investigate the effectiveness of the delivery of learning material.

Author Contributions: Conceptualization, C.T. and A.K.; methodology, C.T. and A.K.; software, C.T. and A.K.; validation, C.T. and A.K.; formal analysis, C.T. and A.K.; investigation, C.T. and A.K.; resources, C.T. and A.K.; data curation, C.T. and A.K.; writing—original draft preparation, C.T. and A.K.; writing—review and editing, C.T. and A.K.; visualization, C.T. and A.K.; supervision, C.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available on request.

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

References

1. 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](#)]
2. Dias, S.B.; Hadjileontiadiou, S.J.; Diniz, J.; Hadjileontiadis, L.J. DeepLMS: A deep learning predictive model for supporting online learning in the Covid-19 era. *Sci. Rep.* **2020**, *10*, 1–17. [[CrossRef](#)]

3. Christudas, B.C.L.; Kirubakaran, E.; Thangaiah, P.R.J. An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials. *Telemat. Inform.* **2018**, *35*, 520–533. [[CrossRef](#)]
4. El Ghouch, N.; Kouissi, M.; En-Naimi, E.M. Multi-agent adaptive learning system based on incremental hybrid case-based reasoning (IHCBR). In Proceedings of the 4th International Conference on Smart City Applications—SCA '19, Casablanca, Morocco, 2–4 October 2019; Association for Computing Machinery (ACM): New York, NY, USA, 2019; p. 50.
5. Korovin, M.; Borgest, N. Multi-agent Approach Towards Creating an Adaptive Learning Environment. In *Proceedings of the Advances in Intelligent Systems and Computing*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2018; pp. 216–224.
6. Nurjanah, D. Good and Similar Learners' Recommendation in Adaptive Learning Systems. In *Proceedings of the 8th International Conference on Computer Supported Education*; SCITEPRESS—Science and Technology Publications: Setúbal, Portugal, 2016; Volume 1, pp. 434–440.
7. Saleh, M.; Salama, R.M. Recommendations for building adaptive cognition-based e-learning. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 385–393. [[CrossRef](#)]
8. Lestari, W.; Nurjanah, D.; Selviandro, N. Adaptive presentation based on learning style and working memory capacity in adaptive learning system. In Proceedings of the 9th International Conference on Computer Supported Education—CSEDU 2017, Porto, Portugal, 21–23 April 2017; pp. 363–370.
9. Martin, J.; Dominic, M. Adaptation using machine learning for personalized elearning environment based on students preference. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 4064–4069.
10. Altaher, A.W.; Hussein, N.J. Adaptive mobile learning framework based on IRT theory. *Int. J. Adv. Trends Comput. Sci. Eng.* **2019**, *8*, 2647–2652. [[CrossRef](#)]
11. Tsarev, R.Y.; Yamskikh, T.N.; Evdokimov, I.V.; Prokopenko, A.V.; Rutskaya, K.A.; Everstova, V.N.; Zhigalov, K.Y. An Approach to Developing Adaptive Electronic Educational Course. In *Proceedings of the Advances in Intelligent Systems and Computing*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2019; pp. 332–341.
12. Alfaro, L.; Rivera, C.; Luna-Urquizo, J.; Castaneda, E.; Fialho, F. Utilization of a Neuro Fuzzy Model for the Online Detection of Learning Styles in Adaptive e-Learning Systems. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 12. [[CrossRef](#)]
13. Alsobhi, A.Y.; Alyoubi, K.H. Adaptation algorithms for selecting personalised learning experience based on learning style and dyslexia type. *Data Technol. Appl.* **2019**, *53*, 189–200. [[CrossRef](#)]
14. Li, R.; Yin, C.; Zhang, X.; David, B. Online Learning Style Modeling for Course Recommendation. In *Advances in Intelligent Systems and Computing*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2018; pp. 1035–1042.
15. Chiu, T.K.; Mok, I.A. Learner expertise and mathematics different order thinking skills in multimedia learning. *Comput. Educ.* **2017**, *107*, 147–164. [[CrossRef](#)]
16. Khamparia, A.; Pandey, B. SVM and PCA Based Learning Feature Classification Approaches for E-Learning System. *Int. J. Web-Based Learn. Teach. Technol.* **2018**, *13*, 32–45. [[CrossRef](#)]
17. Gan, W.; Sun, Y.; Ye, S.; Fan, Y.; Sun, Y. Field-Aware Knowledge Tracing Machine by Modelling Students' Dynamic Learning Procedure and Item Difficulty. In Proceedings of the 2019 International Conference on Data Mining Workshops (ICDMW), Beijing, China, 8–11 November 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 1045–1046.
18. Scheiter, K.; Schubert, C.; Schüler, A.; Schmidt, H.; Zimmermann, G.; Wassermann, B.; Krebs, M.-C.; Eder, T. Adaptive multimedia: Using gaze-contingent instructional guidance to provide personalized processing support. *Comput. Educ.* **2019**, *139*, 31–47. [[CrossRef](#)]
19. Chi, Y.-L.; Hung, C.; Chen, T.-Y. Learning adaptivity in support of flipped learning: An ontological problem-solving approach. *Expert Syst.* **2018**, *35*, e12246. [[CrossRef](#)]
20. Ewais, A.; Awad, M.; Hadia, K. Aligning Learning Materials and Assessment with Course Learning Outcomes in MOOCs Using Data Mining Techniques. In *Blockchain Technology and Innovations in Business Processes*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2020; pp. 1–25.
21. Al Duhayyim, M.; Newbury, P. Concept-based and Fuzzy Adaptive E-learning. In Proceedings of the 2018 the 3rd International Conference on Information and Education Innovations—ICIEI 2018, London, UK, 30 June–2 July 2018; ACM: New York, NY, USA, 2018; pp. 49–56.
22. Cornelisz, I.; Van Klaveren, C. Student engagement with computerized practising: Ability, task value, and difficulty perceptions. *J. Comput. Assist. Learn.* **2018**, *34*, 828–842. [[CrossRef](#)]
23. Bauer, M.; Bräuer, C.; Schuldt, J.; Krömker, H. Adaptive E-learning for Supporting Motivation in the Context of Engineering Science. In *Proceedings of the Advances in Intelligent Systems and Computing*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2019; pp. 409–422.
24. Hamid, S.S.A.; Admodisastro, N.; Ghani, A.A.A.; Kamaruddin, A.; Manshor, N. The development of adaptive learning application to facilitate students with dyslexia in learning malay language. *Int. J. Adv. Sci. Technol.* **2019**, *28*, 265–272.
25. Lin, J. Optimization of Personalized Learning Pathways Based on Competencies and Outcome. In Proceedings of the 2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT), Austin, TX, USA, 25–28 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 49–51.

26. Nabizadeh, A.H.; Gonçalves, D.; Gama, S.; Jorge, J.; Rafsanjani, H.N. Adaptive learning path recommender approach using auxiliary learning objects. *Comput. Educ.* **2020**, *147*, 103777. [[CrossRef](#)]
27. Saito, T.; Watanobe, Y. Learning Path Recommendation System for Programming Education Based on Neural Networks. *Int. J. Distance Educ. Technol.* **2020**, *18*, 36–64. [[CrossRef](#)]
28. El Said, G.R.M. Context-aware adaptive m-learning: Implicit indicators of learning performance, perceived usefulness, and willingness to use. *ASEE Comput. Educ. (CoED) J.* **2019**, *10*. Available online: <https://coed.asee.org/wp-content/uploads/2020/08/5-Context-Aware-Adaptive-M-Learning-Implicit-Indicators-of-Learning-Performance-Perceived-Usefulness-and-Willingness-to-Use.pdf> (accessed on 5 May 2021).
29. Shawky, D.; Badawi, A. Towards a Personalized Learning Experience Using Reinforcement Learning. In *Econometrics for Financial Applications*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2018; pp. 169–187.
30. Gasparetti, F.; De Medio, C.; Limongelli, C.; Sciarrone, F.; Temperini, M. Prerequisites between learning objects: Automatic extraction based on a machine learning approach. *Telemat. Inform.* **2018**, *35*, 595–610. [[CrossRef](#)]
31. Thaker, K.; Brusilovsky, P.; He, D. Student modeling with automatic knowledge component extraction for adaptive textbooks. In *iTextbooks@AIED*; University of Pittsburgh: Pittsburgh, PA, USA, 2019; pp. 95–102.
32. Machado, M.D.O.C.; Barrère, E.; Souza, J. Solving the Adaptive Curriculum Sequencing Problem with Prey-Predator Algorithm. *Int. J. Distance Educ. Technol.* **2019**, *17*, 71–93. [[CrossRef](#)]
33. Cevik, V.; Altun, A. Roles of working memory performance and instructional strategy in complex cognitive task performance. *J. Comput. Assist. Learn.* **2016**, *32*, 594–606. [[CrossRef](#)]
34. Gavrilović, N.; Arsić, A.; Domazet, D.; Mishra, A. Algorithm for adaptive learning process and improving learners' skills in Java programming language. *Comput. Appl. Eng. Educ.* **2018**, *26*, 1362–1382. [[CrossRef](#)]
35. Subirats, L.; Fort, S.; Martín, Á.; Huion, P.; Peltonen, M.; Nousiainen, T.; Miakush, I.; Vesisenaho, M.; Sacha, G.M. Adaptive techniques in e-Learning for transnational programs. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, León, Spain, 16–18 October 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 879–884.
36. Vanitha, V.; Krishnan, P. A modified ant colony algorithm for personalized learning path construction. *J. Intell. Fuzzy Syst.* **2019**, *37*, 6785–6800. [[CrossRef](#)]
37. Farmer, E.C.; Catalano, A.J.; Halpern, A.J. Exploring Student Preference between Textbook Chapters and Adaptive Learning Lessons in an Introductory Environmental Geology Course. *TechTrends* **2020**, *64*, 150–157. [[CrossRef](#)]
38. Merrill, M.D. Component Display Theory. In *Instructional Design Theories and Models: An Overview of Their Current Status*; Reigeluth, C.M., Ed.; Psychology Press: London, UK, 1983; pp. 279–333.
39. Merrill, M.D. A Lesson Based on the Component Display Theory. In *Instructional Theories in Action: Lessons Illustrating Selected Theories and Models*; Reigeluth, C.M., Ed.; Routledge: London, UK, 1987; pp. 201–244.