Adaptive E-Learning System using Fuzzy Logic

Maha E. Attia, Mohamed A. Arteimi
Arteimi@yahoo.com, mahaattia1994@gmail.com
Electrical and Computer Engineering Department- Libyan Academy,

Abstract

The spread of ICT in education and learning has changed the way of teaching and learning, and this raised the issue of getting e-learning systems to adapt/personalize with respect to student's knowledge achievement. However, the relation between the concepts in the programming domain are not definitive and they are fuzzily related. In this Paper, we describe a practical experience for developing adaptable e-learning system using fuzzy logic to model and evaluate student's knowledge for the purpose of adaptation in the field of Computer Programming. Fuzzy logic technique modeled as a plugin on MOODLE Learning Management System at the Libyan Academy for Postgraduate Studies is developed. Conclusions highlighted some important lessons in increasing learning effectiveness and learner's satisfaction.

Index Terms— Fuzzy Logic, E-Learning, Adaptable Systems, Personalized Learning, MOODLE.

1. Introduction

Many universities all over the world have taken steps to make available online courses, seminars and conferences. The COVID-19 pandemic caused huge chaos and resulted in closing of activities and classrooms in many countries, forcing millions of students and teachers to change their academic practices. This situation forces E-Learning Paradigm to demonstrate its potential as a sustainable teaching method.

Students can have access to E-Learning systems from everywhere at any time. Therefore, E-Learning systems have to adapt/personalize the learning content and learning processes to the needs of each individual learner.

Artificial Intelligence techniques are widely used in several applications of e-learning. Fuzzy logic was chosen in this research because it is based on human reasoning rather than on rigid calculations, it allows better understanding and interpretation of the domain of teaching computer programming.

Measuring the student's knowledge is a difficult task, and usually, students tested on each of their classes on the topics and concepts for that particular class. The measure of knowledge gained for topics was used to control the learning process. The relationships between topics take several forms, such as some topics are prerequisite for other topics. Therefore, if a student did not understand the prerequisite topics, it is hard to understand the depended topics.

The representation of dependencies between domain topics of the learning material includes imprecise and uncertain information. As a result, an effective solution for handling this uncertainty is to use fuzzy logic techniques in the representation of knowledge domain and controlling the learning process. Chrysafiadi and Virvu [7] have showed that the integration of fuzzy logic into the student's model of e-learning systems can increase learner's satisfaction and performance, as well as improve adaptability of learning processes.

In this paper, we propose a fuzzy Adaptive MOODLE system (Figure-1), to guide the student in navigating the learning material in computer programming domain, according to their knowledge achievement in each lesson provided.

The outline of the paper is as follows: Section (2) provides background about adaptive learning strategies used in the literature. Section (3) discusses the proposed Fuzzy learning management system. Section (4) defines the evaluation methodology used for the proposed system. Section (5) Presents final discussion about conclusions and future work.

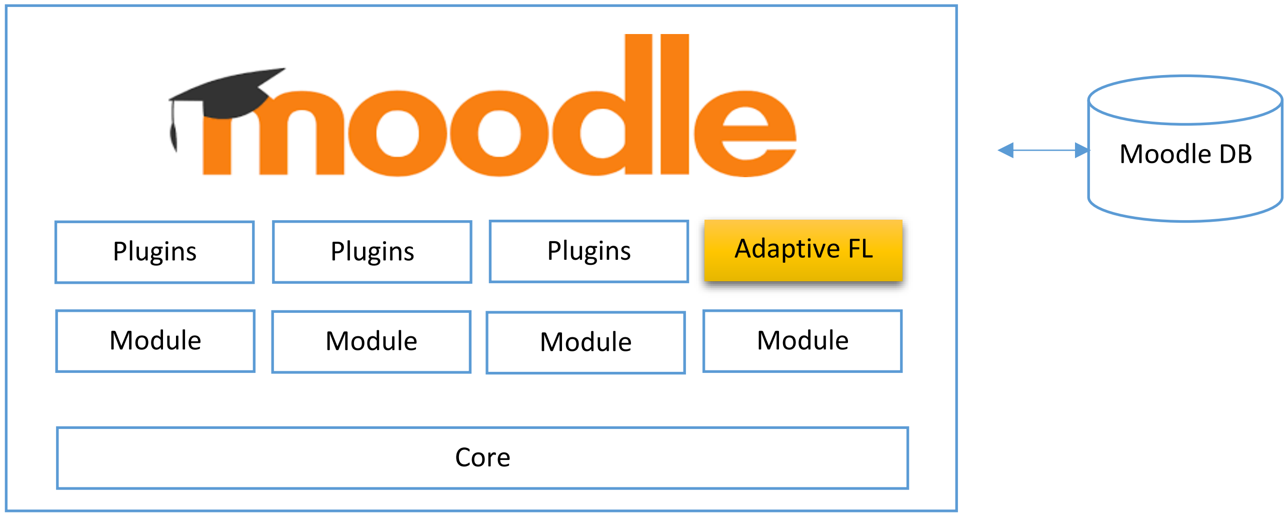


Figure 1: Architecture of Moodle with Adaptive FL

2. Adaptive E-Learning

E-learning systems have become more and more popular and many adaptive e-learning environments proposed to offer learners customized courses in accordance with their aptitudes and learning results. The aim of adaptive e-learning systems (ALS) is to customize the educational logic developed in their courses, in an attempt to be more flexible and adaptive by building a model of goals, preferences and knowledge of each individual learner and using this model throughout the interaction with the learner in order to adapt to the needs of that learner. Architecture of adaptive learning systems decomposed into three main models that are required to create or automatically generate customized courses to a learner. Learner Model (LM) describes the learner's profile such as his knowledge, his characteristics and his preferences, Adaptation Model (AM) defines the concept selection rules used for selecting appropriate concepts from the domain model, and Domain Model (DM) composed of a set of small domain knowledge elements (DKE). Each DKE represents an elementary fragment of knowledge for the given domain[1]. As an example, Sucheta V. Kolekar, et al. [3] presented an approach to identify the learning styles for adaptation as specified by Felder-Silverman Learning Style Model (FSLSM). An e-learning application developed using Moodle framework with the functionality to capture the usage data of learners. The usage data was used to cluster the learners as per learning categories of FSLSM. The customization was provided on the portal by generating the adaptive user interface for each learner based on learning style of FSLSM. The adaptation of system is validated using statistical analysis and the impact of adaptation on the learning has been identified, Also María Lucila Morales-Rodríguez, et al. [4] proposed a design of an Intelligent Agent for Personalization of Moodle Contents to create an Intelligent Learning Management System (ILMS applied to the LMS Moodle) by applying the ITS architecture. They used an intelligent agent for choosing a teaching strategy to select the content that will be displayed to the student. They focused on the learning styles of different students according to an analysis of the VARK model. However, it was limited by not supporting student aptitudes who differ based on the VARK model. In addition MOIZ UDDIN AHMED, et al. [5] proposed a model of adaptive e-learning for an ODL environment. They proposed a model with three important components: content model, learner model and adaptive model. The content model defines and stores the organization of chapters and topics in a logical sequence. The learner model stores information about survey and quizzes data and keep the track of the learning performance. The survey was designed to investigate the personal profiles and the preference of students about their favorite content types. The adaptive model managed content presentation and its navigation derived by the learning algorithm. The learning algorithm took information from the learner model, analyzed it and adjusts the display of the contents by retrieving it from the content model. Also Birol Ciloglugil and Mustafa Murat Inceoglu [6] proposed an adaptive e-learning environment architecture that supports personalization by utilizing Agents and Artifacts (A&A) Metamodel. A&A Metamodel focuses on environment modeling in multi agent system (MAS) design and models entities in agent’s environment with artifacts as first-class entities like the agents. From the perspective of MAS based e-learning systems, learner models and learning resources are part of the environment of the agents and agents interact with them constantly. Thus, they proposed an e-learning architecture that focuses on environment abstraction and models access to different learner models and learning resources with artifacts to support personalization.

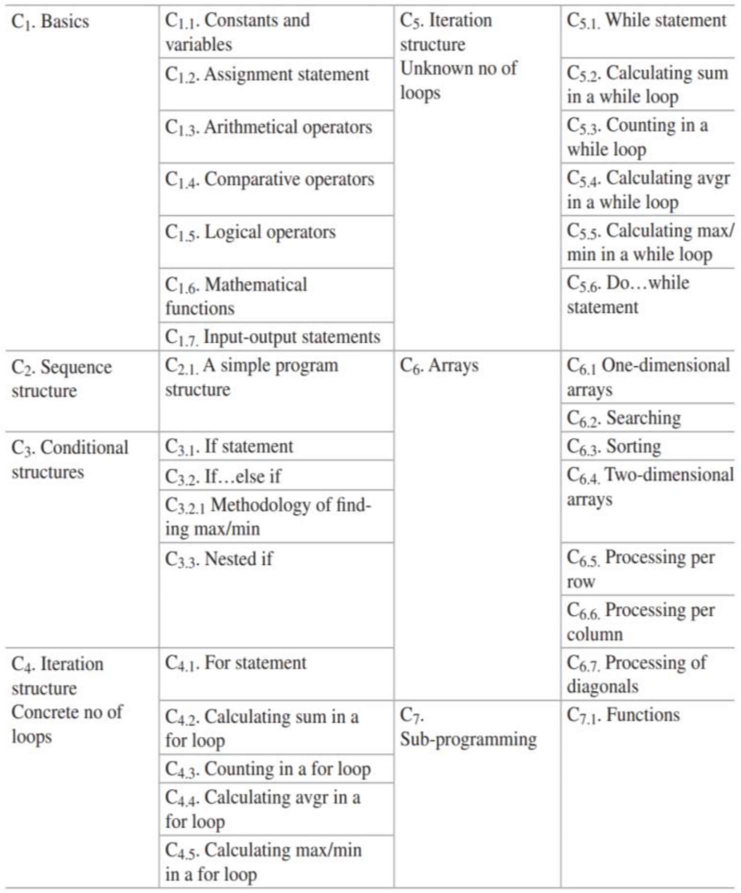
3. Proposed System

Fuzzy logic can manage uncertainty in situations when data is incomplete as well as individual subjectivity with respect to the domain particulars. In teaching computer programing language, the topics covered have an inherent relationship, meaning that some topics are prerequisite for other topics. Domain experts can define this relation with certain degree of strength, but this is subjective, and therefore fuzzy logic is very suitable for modeling course material.

The problem solving techniques used in this research employ concept maps for managing learning material relationships in the course, and student's knowledge (performance) is evaluated using fuzzy logic by converting student's grade to membership functions as discussed in Section 3.1.

One important module of an adaptive e-learning system is the domain model, which contains a description of the knowledge or behaviors that represent expertise in the subject-matter domain of the system. To enable communication between system and learner at content level, a Fuzzy Related-Concepts Network displays the knowledge relationships that exist between the learning material's domain concepts. As a result, it depicts the learning material's structure as well as the concepts' knowledge dependencies. It depicts the notion that when the knowledge level of a connected topic improves, the knowledge level of a domain concept increases, and when the knowledge level of a dependent topic is not adequate, the knowledge level of a domain concept decreases.

The knowledge domain of the programming tutoring system is the programming language ‘C Sharp’, which was used in teaching learners the principles and structures of this programming language. The learning material includes expressions, operations and statements, algorithms like calculating sums, averages and maximums or minimums. As a result, the learning material is broken down into domain concepts such as variable and constant declarations, expressions and operators, input and output expressions, and the sequential execution of a program, the if, if-then and if-else if statements, the iteration statements (for loop, while loop, do…while loop), sorting and searching algorithms, arrays, functions (Table 1). The Fuzzy Related-Concepts Network used to perform this. Figure 2, adopted from 7, shows the FR-CN for the knowledge domain of the programming language ‘C Sharp’.



C_1. Basics	$C_{1.1}$. Constants and variables	C_5. Iteration structure Unknown no of loops	$C_{5.1}$. While statement
	$C_{1.2}$. Assignment statement		$C_{5.2}$. Calculating sum in a while loop
	$C_{1.3}$. Arithmetical operators		$C_{5.3}$. Counting in a while loop
	$C_{1.4}$. Comparative operators		$C_{5.4}$. Calculating avgr in a while loop
	$C_{1.5}$. Logical operators		$C_{5.5}$. Calculating max/min in a while loop
	$C_{1.6}$. Mathematical functions		$C_{5.6}$. Do…while statement
	$C_{1.7}$. Input-output statements		
C_2. Sequence structure	$C_{2.1}$. A simple program structure	C_6. Arrays	$C_{6.1}$ One-dimensional arrays
			$C_{6.2}$. Searching
C_3. Conditional structures	$C_{3.1}$. If statement		$C_{6.3}$. Sorting
	$C_{3.2}$. If…else if		$C_{6.4}$. Two-dimensional arrays
	$C_{3.2.1}$ Methodology of finding max/min		
	$C_{3.3}$. Nested if		$C_{6.5}$. Processing per row
			$C_{6.6}$. Processing per column
C_4. Iteration structure Concrete no of loops	$C_{4.1}$. For statement		$C_{6.7}$. Processing of diagonals
	$C_{4.2}$. Calculating sum in a for loop	C_7. Sub-programming	$C_{7.1}$. Functions
	$C_{4.3}$. Counting in a for loop		
	$C_{4.4}$. Calculating avgr in a for loop		
	$C_{4.5}$. Calculating max/min in a for loop		

Figure:2 Knowledge domain of programming language 'C Sharp, Adopted from [7]

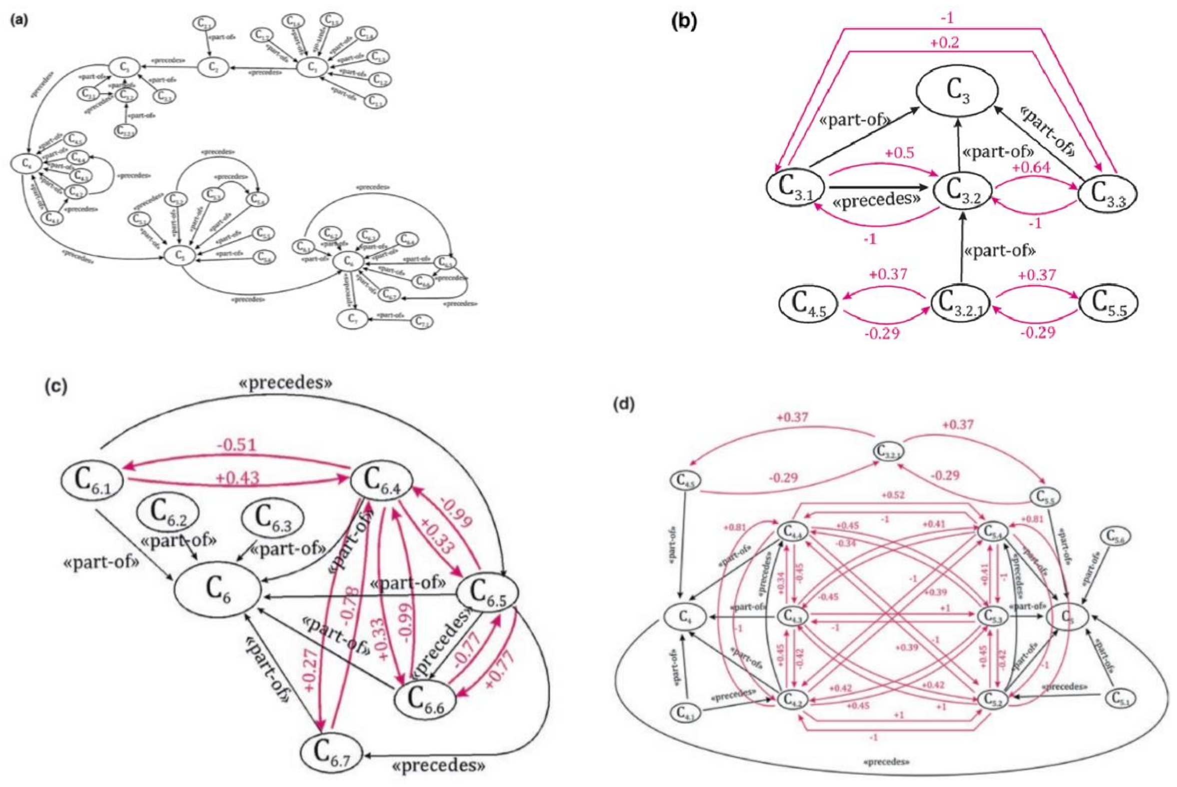


Figure 3: FR-CN of the computer language C Sharp's knowledge domain.

(It is composed of four Figures, adopted from [7]**)**

(a) The FR-CN's "precedence" and "part-of" relations; (b) The knowledge dependence relations for the section 3 domain concepts; (c) The knowledge dependence relations for the section 6 domain concepts; (d) The knowledge dependence relations for the sections 4 and 5 domain concepts

The value 1 on the FR-CN's directed arc connecting two dependent domain ideas indicates that if a learner understands one domain concept, he or she may also understand a related domain concept to the same degree. For example, if a student has been tested and found to know both the "for" and "while" loops, and knows how to calculate sum in a "for" loop, he or she will also know how to calculate sum in a "while" loop, because the process is the same. Experts in programming have described the learning material's domain ideas as their relationships ("precedence", "part-of", " knowledge dependency"). Fuzzy Cognitive Mapping is a technique for representing knowledge of systems with a lot of uncertainty and complicated processes that combines fuzzy logic and cognitive mapping.

3.1 Operation of Adaptive e-learning system by fuzzy logic

MOODLE (Modular Object Oriented Dynamic Learning Environment) is a learning platform that is famous for its use, and active community available for support, as well as a number of available features. MOODLE system offers several resources such as: book, lesson, file, page, URL, etc. and activities such as: chat, quiz, forum, etc. In MOODLE, it is possible to extend the functionalities using plugins. MOODLE provides students with access to a range of educational material, interact with teachers and with their colleagues, also teachers can manage learning and learning process through access control available on MOODLE depending on date, grade, group, grouping, and user profile. In this research, we developed another approach for providing access control for learning through fuzzy logic, implemented as a plugin.

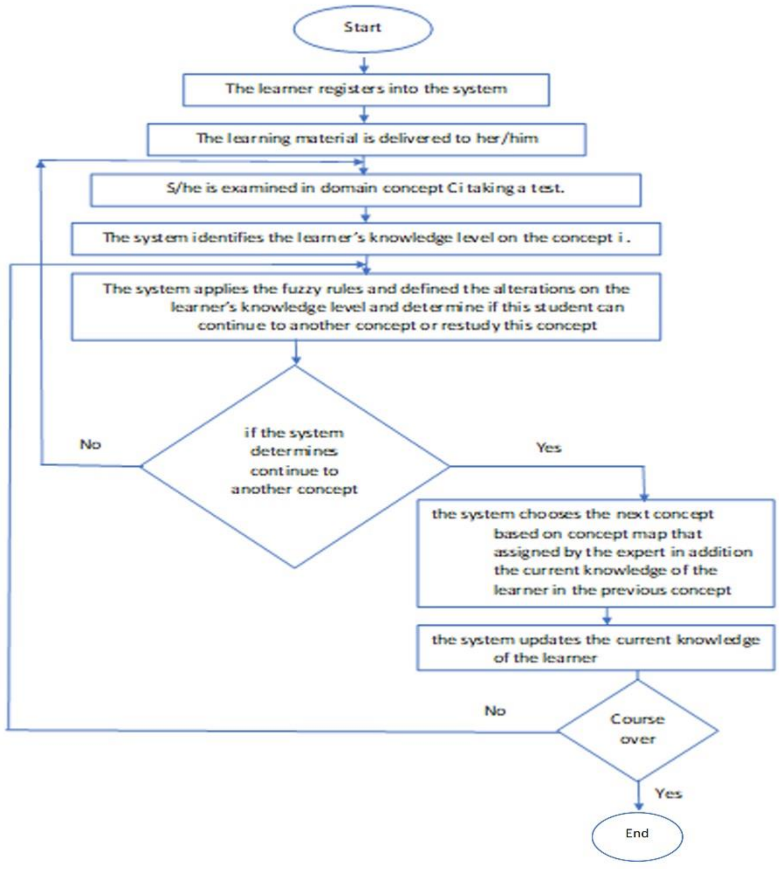

Start
The learner registers into the system
The learning material is delivered to her/him
S/he is examined in domain concept Ci taking a test.
The system identifies the learner's knowledge level on the concept i .
The system applies the fuzzy rules and defined the alterations on the learner's knowledge level and determine if this student can continue to another concept or restudy this concept
if the system determines continue to another concept
No
Yes
the system chooses the next concept based on concept map that assigned by the expert in addition the current knowledge of the learner in the previous concept
the system updates the current knowledge of the learner
Course over
No
Yes
End

Figure 4: Flowchart for the Proposed FL- MOODLE System

We use fuzzy sets to represent the student's knowledge level for each lesson, the following fuzzy sets represent student's knowledge level for a lesson (x indicates the student's degree of success in a lesson).

a- Fuzzy Sets

The following are the defined fuzzy sets and their membership functions:

- Unknown (Un): the success rate in the domain concept ranges from 0% to 50%.
- Moderate Known (MKn): the domain concept's success rate ranges from 40% to 70%.
- Known (Kn): the success rate in the domain concept ranges from 60% to 80%.
- Learned (L): the success rate in the domain concept ranges from 75 to 90%.
- Assimilated (A): the success rate in the domain concept ranges from 85 to 100%.

b- Overlay Model

The qualitative values of the fuzzy-weighted qualitative overlay model are the defined fuzzy sets. In other words, they are the values: 'unknown', 'moderate known', 'known', 'learned' and 'assimilated'. Therefore, the overlay model uses a quintet (μUn, μMKn, μKn, μL, μA), which expresses the degree in which each of the above qualitative values are active (Fig. 3.8). For example, (0, 0, 0.6, 0.3, 0.1) declares that the domain concept is 60 % 'known', 30 % 'learned' and 10 % 'assimilated'. Similarly, (0.7, 0.3, 0, 0, 0) declares that the concept is 70 % 'Unknown' and 30 % 'moderate known'.

c- Cognitive State Transitions of Learners of the Programming Tutoring System

Through the learning process, the domain concept passes through five states ('unknown', 'moderate known', 'known', 'learned', 'assimilated') during the interaction of the user with the system. Furthermore, a learner passes through several states during the learning process. S/he can learn or not, or forget etc. These states determine the progress of the learner each time; they determine the transition from one knowledge level to another. The transition from one knowledge level to another can reveal the state of the learner.

d- Dependency

- Low: the degree of dependency between the domain concepts is from 0 to 33 %.
- Medium: the degree of dependency between the domain concepts is from 34 to 66 %.
- High: the degree of dependency between the domain concepts is from 67 to 100 %.

The presented Adaptive FL, was implemented in a web-based educational application for teaching programming language 'C Sharp'. The presented system provides adaptation of the instructional material, taking into account the individuality of learners in terms of background, skills and pace of learning. The particular student model includes a fuzzy-weighted qualitative overlay model; a rule-based fuzzy system, and a dependency model. During the learning process, the system assists learners in saving time and effort. Because the student model allows each learner to complete the course at their own pace, they can choose which concepts to deliver, which concepts to revise, and which concepts they already know and do not need to reread.

e- Fuzzy Rules

In the programming domain, for any two concepts C_i and C_j where C_i is taught before C_j, the knowledge level of the learner can change according to predefined rules adopted from Chrysafiadi [7]. There are (8) rules defining relationships of knowledge dependency, and they are explained in detail in [8].

4. Evaluation of Adaptive FL

The evaluation of adaptive e-learning systems is a difficult task due to the complexity of such systems, for this reason there are many evaluation methods available in literature, however there is no standard agreed measurement framework for assessing the value and effectiveness of the adaptation yielded by adaptive systems. The most common practice of evaluation is through experiments. For this reason, the presented evaluation process is performed through applying the evaluation framework PERSIVA[7], which include both questionnaires and observations through experiments. Applying the particular evaluation framework, either the educational impact (i.e., performance, satisfaction, change of learners' attitudes) or the adaptation of the personalized and/or adaptive mechanism is assessed. The presented evaluation includes the following two levels:

• Level 1: Evaluation of the educational impact. This level includes assessment of learners' performance and satisfaction.

• Level 2: Evaluation of adaptation. This level is responsible for giving answers to the following questions:

-How satisfied are the learners about the system's adaptive responses to their needs?

– How important and essential is to model the particular student's characteristics (prior knowledge, knowledge level, type of errors etc.)?

– How valid are the conclusions drawn by the student model concerning the aspects of the students' characteristics (i.e., their background)?

– How valid are the decisions for adaptivity?

Students, who used the presented programming tutoring system, were divided into three categories according to their background knowledge. There are three categories namely: arts, science fields (other than computer science) and computer science related fields.

For First group which is group A, students are studying computer science in different stages and they have a background in programming languages, and they learned more than one programming language. Group B consists of students studying in different fields other than computer science, examples of such backgrounds are geophysics, mathematics, geography and they do not have any background about programming languages. Both groups consisted of 34 students. The system allows each student, in principle, to take first lesson only, and then follows a sequence of providing lessons depending on his background and level of knowledge in previous lessons, in addition to considering the relationships between lessons and their dependence on each other. Both groups used a similar educational system. Both systems had the same knowledge domain, but students of group B, had no previous experience in programming. Students, who used the system, had to read all the concepts one time at least. Furthermore, according to the learner's degree of success on tests, the system advised if s/he had to return to a previous concept or if s/he had to be transited to the next section of the learning material. The learners of both groups used the corresponding systems without attending any complementary course on programming, over a period of one month. The Learners' general satisfaction about the adaptive e-learning system is very good, the average learners' overall rating of satisfaction is 4.00. The results are depicted in Figure 5.

As for the performance of the learners, a classical pre-test and post-test methodology could not be sufficient for the aimed evaluation. The value of factor P, is derived through dividing the learner's degree of success in a particular domain concept with the times that s/he needs to read the particular concept, was measured for group A and group B. Then, the two average values compared to assess the impact of the presented fuzzy student model on the student's performance and progress.

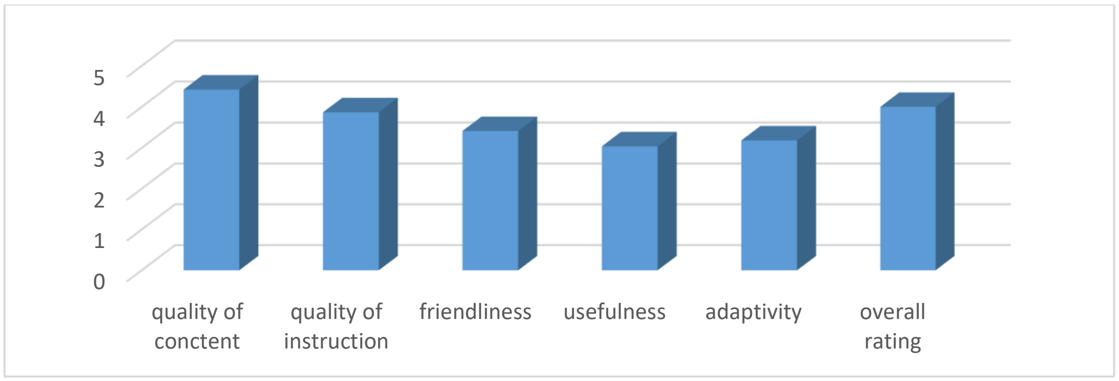


Figure 5: Learners' general satisfaction about the programming tutoring system

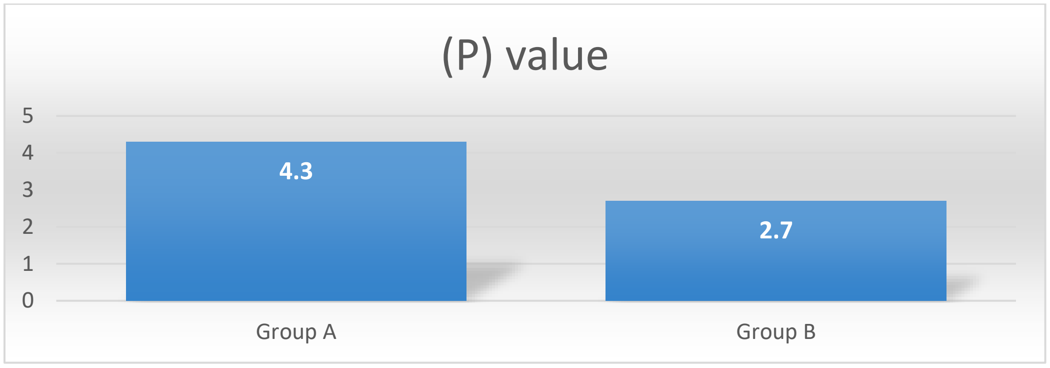


Figure 6: Learners' general performance

It is the dynamic of adaptation in the sequence of lessons according to the knowledge level of the user in this sense, the evaluation's aim is to evaluate how the student is taught and whether s/he learned successfully rather than just to see if s/he learned successfully.

Case 1: Learner 1 was from group B (with no background about programming language), this learner had learned section 1 (domain concepts 1.1 to 1.7) and section 2 (domain concept 2.1) and he was taught the domain concepts of section 3 (domain concepts 3.1 to 3.3) (Interaction I of Table 1). He read concept C3.1. Then, he was examined in the particular domain concept and succeeded 70%. According to the above, the value of the defined membership functions for concept C3.1 become $\mu Un = 0$, $\mu MKn = 0$, $\mu Kn = 1$, $\mu L = 0$ and $\mu A = 0$. According to the FR-CN (Figure 3) the concept C3.1 affects the following concepts C3.2 and C3.3 with "strength of impact" 0.5 and 0.2 correspondingly. Consequently, applying the fuzzy rule R2 (b) and (c), KL (C3.2) becomes 50 % 'Known'. The rest 50 % of the particular concept remains 'Unknown' (Interaction II of Table 1). Similarly, applying the same rules, KL (C3.3) becomes 20 % 'Known'. The rest 80 % of the particular concept remains 'Unknown' (Interaction II of Table 1). Therefore, although concepts C3.2 and C3.3 are not completely unknown to this learner, so the system didn't advise him to skip them, Thus, it became necessary for this learner to study it and increase his knowledge level for this section (section 3) (domain concepts 3.1 to 3.3) in order to be able to complete the rest of the course.

Table 1: Learner (A) progress

Domain concepts	Learner's knowledge	
	Interaction I (μUn, μMKn, μKn, μL, μA)	Interaction II (μUn, μMKn, μKn, μL, μA)
1.1 Constants and variables	(0,0,0,1,0)	(0,0,0,1,0)
1.2 Assignment statement	(0,0,0,1,0)	(0,0,0,1,0)
1.3 Arithmetic operators	(0,0,0,1,0)	(0,0,0,1,0)
1.4 Comparative operators	(0,0,0,1,0)	(0,0,0,1,0)
1.5 Logical operators	(0,0,0,1,0)	(0,0,0,1,0)
1.6. Mathematic functions	(0,0,1,0,0)	(0,0,1,0,0)
1.7 Input-output statements	(0,0,0,1,0)	(0,0,0,1,0)
2.1 A simple program's structure	(0,0,0,1,0)	(0,0,0,1,0)
3.1 If statement	*(1,0,0,0,0)*	*(0,0,1,0,0)*
3.2 If…else if	***(1,0,0,0,0)***	***(0.5,0,0.5,0,0)***
3.2.1 Finding max, min	(1,0,0,0,0)	(1,0,0,0,0)
3.3 Nested if statement	***(1,0,0,0,0)***	***(0.8,0,0.2,0,0)***
4.1 For statement	(1,0,0,0,0)	(1,0,0,0,0)
4.2 Calc. sum in a for loop	(1,0,0,0,0)	(1,0,0,0,0)
4.3 Counting in a for loop	(1,0,0,0,0)	(1,0,0,0,0)
4.4 Calc. avrg in a for loop	(1,0,0,0,0)	(1,0,0,0,0)
4.5 Calc. max/min in a for loop	(1,0,0,0,0)	(1,0,0,0,0)
5.1 While statement	(1,0,0,0,0)	(1,0,0,0,0)
5.2 Calc. sum in a while loop	(1,0,0,0,0)	(1,0,0,0,0)
5.3 Counting in a while loop	(1,0,0,0,0)	(1,0,0,0,0)
5.4 Calc. avrg in a while loop	(1,0,0,0,0)	(1,0,0,0,0)
5.5 Calc. max/min in a while loop	(1,0,0,0,0)	(1,0,0,0,0)
5.6 Do…until	(1,0,0,0,0)	(1,0,0,0,0)
6.1 One-dimension arrays	(1,0,0,0,0)	(1,0,0,0,0)
6.2 Searching	(1,0,0,0,0)	(1,0,0,0,0)
6.3 Sorting	(1,0,0,0,0)	(1,0,0,0,0)
6.4 Two-dimensions arrays	(1,0,0,0,0)	(1,0,0,0,0)
6.5 Processing per rows	(1,0,0,0,0)	(1,0,0,0,0)
6.6 Processing per column	(1,0,0,0,0)	(1,0,0,0,0)
6.7 Processing of diagonals	(1,0,0,0,0)	(1,0,0,0,0)
7.1 Functions	(1,0,0,0,0)	(1,0,0,0,0)

Case 2: Learner 2 was from group A (had knowledge of more than one programming language). This learner had learned section 1 (domain concepts 1.1 to 1.7), and section 2 (domain concept 2.1) and she was taught the domain concepts of the section 3 (domain concepts 3.1 to 3.3) and the concepts 4.1, 4.5 and 5.5 (Interaction I of Table 2). She read the concept C4.2 to improve her knowledge level. Then, she was examined in the particular domain concept and succeeded 80 %. According to the above, the value of the defined membership functions for concept C4.2 become μUn = 0, μMKn = 0, μKn = 0, μL = 1 and μA = 0. According to the FR-CN (Figure 3) the concept C4.2 affects the following concepts C4.3, C4.4, C5.2, C5.3 and C5.4 with "strength of impact" 0.45, 0.81, 1, 0.45 and 0.39 correspondingly. Consequently, applying the fuzzy rule R2 (c) and (d), KL (C4.3) becomes 75 % 'known' and the rest 25 % of the particular concept remains 'Unknown' (Interaction II of Table 2). Similarly, applying the same rules, KL (C4.4) becomes

75% ‘Learned’ and 16.2 % ‘Assimilated’ (the rest 19 % of the particular remains ‘Unknown’), KL(C5.2) becomes 80 % ‘Learned’. For that the system advised her to skip it, but the lesson is available to her, where the student has the freedom to choose whether she wants to review it or not.

Table 2: Learner (B) progress

Domain concepts	Learner's knowledge	
	Interaction I (μUn, μMKn, μKn, μL, μA)	Interaction II (μUn, μMKn, μKn, μL, μA)
1.1 Constants and variables	(0,0,0,1,0)	(0,0,0,1,0)
1.2 Assignment statement	(0,0,0,1,0)	(0,0,0,1,0)
1.3 Arithmetic operators	(0,0,0,1,0)	(0,0,0,1,0)
1.4 Comparative operators	(0,0,0,1,0)	(0,0,0,1,0)
1.5 Logical operators	(0,0,0,1,0)	(0,0,0,1,0)
1.6. Mathematic functions	(0,0,1,0,0)	(0,0,1,0,0)
1.7 Input-output statements	(0,0,0,1,0)	(0,0,0,1,0)
2.1 A simple program's structure	(0,0,0,1,0)	(0,0,0,1,0)
3.1 If statement	(0,0,0,0,1)	(0,0,0,0,1)
3.2 If…else if	(0,0,0,0,1)	(0,0,0,0,1)
3.2.1 Finding max, min	(0,0,0,0,1)	(0,0,0,0,1)
3.3 Nested if statement	(0,0,0,0.1,0.9)	(0,0,0,0.1,0.9)
4.1 For statement	(0,0,0,0,1)	(0,0,0,0,1)
4.2 Calc. sum in a for loop	*(1,0,0,0,0)*	*(0,0,0,1,0)*
4.3 Counting in a for loop	***(1,0,0,0,0)***	***(0.55,0,0,0.45,0)***
4.4 Calc. avrg in a for loop	***(1,0,0,0,0)***	***(0.19,0,0,0.81,0)***
4.5 Calc. max/min in a for loop	(0,0,0,0.3,0.9)	(0,0,0,0.3,0.9)
5.1 While statement	(1,0,0,0,0)	(1,0,0,0,0)
5.2 Calc. sum in a while loop	***(1,0,0,0,0)***	***(0,0,0,1,0)***
5.3 Counting in a while loop	***(1,0,0,0,0)***	***(0.55,0,0,0.45,0)***
5.4 Calc. avrg in a while loop	***(1,0,0,0,0)***	***(0.61,0,0,0.39,0)***
5.5 Calc. max/min in a while loop	(0,0,0,0.3,0.9)	(0,0,0,0.3,0.9)
5.6 Do…until	(1,0,0,0,0)	(1,0,0,0,0)
6.1 One-dimension arrays	(1,0,0,0,0)	(1,0,0,0,0)
6.2 Searching	(1,0,0,0,0)	(1,0,0,0,0)
6.3 Sorting	(1,0,0,0,0)	(1,0,0,0,0)
6.4 Two-dimensions arrays	(1,0,0,0,0)	(1,0,0,0,0)
6.5 Processing per rows	(1,0,0,0,0)	(1,0,0,0,0)
6.6 Processing per column	(1,0,0,0,0)	(1,0,0,0,0)
6.7 Processing of diagonals	(1,0,0,0,0)	(1,0,0,0,0)
7.1 Functions	(1,0,0,0,0)	(1,0,0,0,0)

Case 3: Learner 3 was also from group B (had not background about programming language), This learner had learned section 1 (domain concepts 1.1 to 1.7) and section 2 (domain concept 2.1) and he was taught domain concepts of section 3 (domain concepts 3.1 to 3.3), 4 (the domain concepts 4.1 to 4.5) (Interaction I of Table 3). He read concept 5 (domain concepts 5.1 and 5.2. Then, he was examined in these particular domain concept 5.2 and got 30%, According to the FR-CN (Figure 3) concept C5.2 Under the influence of preceding concepts C4.2, C4.3, C4.4, with

"strength of impact" 0.45, 0.81 and 1 correspondingly According to the above, the student's score is lower than expected for this concept, which indicates a lack of understanding or forgetting some concepts that were learned in the past. Therefore, the system advised him to re-examine the concepts on which this concept is based on to improve his knowledge level.

Table 3: Learner (C) progress

Domain concepts	Learner's knowledge	
	Interaction I (μUn, μMKn, μKn, μL, μA)	Interaction II (μUn, μMKn, μKn, μL, μA)
1.1 Constants and variables	(0,0,0,1,0)	(0,0,0,1,0)
1.2 Assignment statement	(0,0,0,1,0)	(0,0,0,1,0)
1.3 Arithmetic operators	(0,0,0,1,0)	(0,0,0,1,0)
1.4 Comparative operators	(0,0,0,1,0)	(0,0,0,1,0)
1.5 Logical operators	(0,0,0,1,0)	(0,0,0,1,0)
1.6. Mathematic functions	(0,0,1,0,0)	(0,0,1,0,0)
1.7 Input-output statements	(0,0,0,1,0)	(0,0,0,1,0)
2.1 A simple program's structure	(0,0,0,1,0)	(0,0,0,1,0)
3.1 If statement	(0,0,0.3,0.7,0)	(0,0,0.3,0.7,0)
3.2 If…else if	(0,0,0.4,0.6,0)	(0,0,0.4,0.6,0)
3.2.1 Finding max, min	(0,0,0.1,0.9,0)	(0,0,0.1,0.9,0)
3.3 Nested if statement	(0,0,0,0.6,0.4)	(0,0,0,0.6,0.4)
4.1 For statement	(0,0,0,0.5,0.5)	(0,0,0,0.5,0.5)
4.2 Calc. sum in a for loop	***(0,0,0,0.2,0.8)***	***(0,1,0,0,0)***
4.3 Counting in a for loop	***(0,0,0,0.6,0.4)***	***(0,0,1,0,0)***
4.4 Calc. avrg in a for loop	***(0,0,0,1,0)***	***(0,0,1,0,0)***
4.5 Calc. max/min in a for loop	(0,0,0,1,0)	(0,0,0,1,0)
5.1 While statement	(0,0,0,1,0)	(0,0,0,1,0)
5.2 Calc. sum in a while loop	*(0,0,0,0,0)*	*(1,0,0,0,0)*
5.3 Counting in a while loop	***(1,0,0,0,0)***	***(0,1,0,0,0)***
5.4 Calc. avrg in a while loop	***(1,0,0,0,0)***	***(0.5,0.5,0,0,0)***
5.5 Calc. max/min in a while loop	(1,0,0,0,0)	(1,0,0,0,0)
5.6 Do…until	(1,0,0,0,0)	(1,0,0,0,0)
6.1 One-dimension arrays	(1,0,0,0,0)	(1,0,0,0,0)
6.2 Searching	(1,0,0,0,0)	(1,0,0,0,0)
6.3 Sorting	(1,0,0,0,0)	(1,0,0,0,0)
6.4 Two-dimensions arrays	(1,0,0,0,0)	(1,0,0,0,0)
6.5 Processing per rows	(1,0,0,0,0)	(1,0,0,0,0)
6.6 Processing per column	(1,0,0,0,0)	(1,0,0,0,0)
6.7 Processing of diagonals	(1,0,0,0,0)	(1,0,0,0,0)
7.1 Functions	(1,0,0,0,0)	(1,0,0,0,0)

The results of the validity of the system's decision making were very good and encouraging. It has shown that the percentage of time that the learners spent to revise a previous domain concept is not at all waste of time.

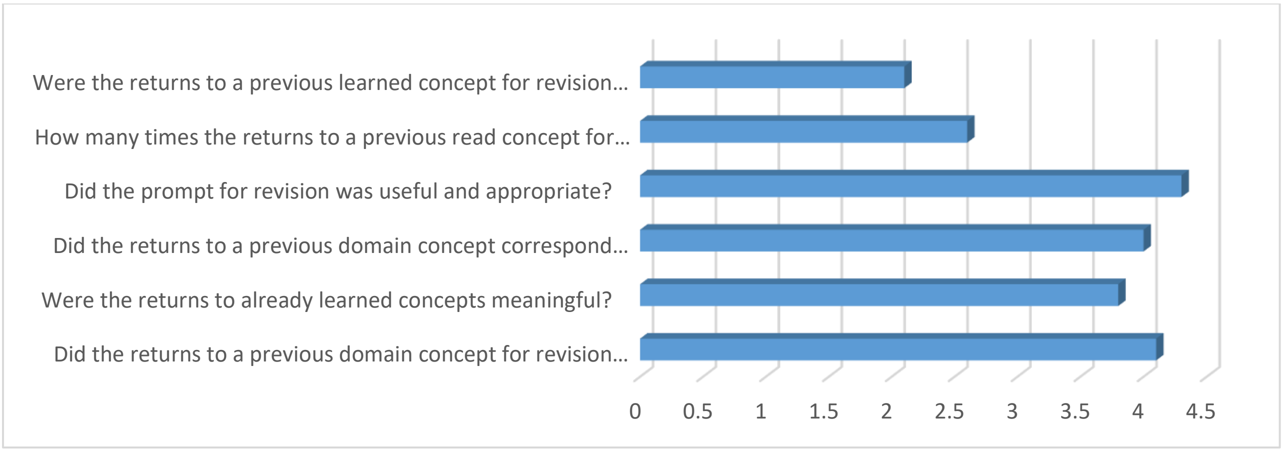


Figure 7: Learners' answers about the appropriateness or necessity of the returns to a domain concept for revision

Based on results of evaluation, the system's decisions about adaptive responses to the learner's needs are valid. The learners seem to do better in the end and absorb the learning material. The percentage of times a learner needs to read the concept of a domain that the system advises not to read is satisfactory enough to be able to lead to the conclusion that decisions, made by the system based on the student's model, are correct. The target of this system was to combine fuzzy logic techniques for offering individualized instruction and personalized support in adaptive educational systems. The presented student modeling approach adapting the delivery of the knowledge domain to the individual learner's learning needs and pace. In addition, the system keeps track of cognitive state transitions of learners with respect to their progress or no-progress. Thereby, it reveals if a student learns or not, Therefore, the system allows each individual learner to complete the e-learning course at her/his own pace, taking decisions about the concepts of the learning material that have to be delivered to them, the concepts that need revision and the concepts that are known and do not need further reading. In this way, the system helps learners to save time and effort during the course.

5. Conclusion and Future work

In this research paper, we presented the idea of utilizing Fuzzy logic as a plugin in MOODLE system, for evaluating student's performance in the domain of C sharp programming. The developed plugin determines the course path according to student's understanding of the given topic at hand with certain predefined level. If the student did not reach that level of understanding, the system does not allow the student to move forward to the depended topic. Course material provided with relationship between course topics, defined by the course teacher. Experimentation with the adaptive MOODLE System showed that integration of Fuzzy Logic significantly improves student's learning and provides satisfaction when using the system. Further study could be using the developed system on complete course at university level. We also recommend the application of this system in different types of subjects or domains as it could be useful for future investigations.

6. References

[1] K. Chrysafiadi and M. Virvou, *Advances in Personalized Web-Based Education*, vol. 78. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-12895-5.

[2] A. Aajli and K. Afdel, “Generation of an adaptive e-learning domain model based on a fuzzy logic approach,” Nov. 2016. doi: 10.1109/AICCSA.2016.7945708.

[3] S. v. Kolekar, R. M. Pai, and M. Pai M.M., “Adaptive User Interface for Moodle based E-learning System using Learning Styles,” *Procedia Computer Science*, vol. 135, 2018, doi: 10.1016/j.procs.2018.08.226.

[4] M. Lucila Morales-Rodríguez, J. Apolinar Ramírez-Saldivar, J. Patricia Sánchez Solís, and A. Hernández Ramírez, “Design of an Intelligent Agent for Personalization of Moodle Contents Optimization and Decision Support View project,” 2012. [Online]. Available: https://www.researchgate.net/publication/270904751

[5] M. Ahmed, N. A. Sangi, and A. Mahmood, “A Model of Adaptive E-Learning in an ODL Environment,” *Mehran University Research Journal of Engineering and Technology*, vol. 37, no. 2, pp. 367–382, Apr. 2018, doi: 10.22581/muet1982.1802.13.

[6] B. Ciloglugil and M. M. Inceoglu, “An Adaptive E-Learning Environment Architecture Based on Agents and Artifacts Metamodel,” Jul. 2018. doi: 10.1109/ICALT.2018.00024.

[7] K. Chrysafiadi and M. Virvou, “PeRSIVA: An empirical evaluation method of a student model of an intelligent e-learning environment for computer programming,” *Computers and Education*, vol. 68, pp. 322–333, 2013, doi: 10.1016/j.compedu.2013.05.020.

[8] M. Attia and M. Arteimi, "Adaptive E-Learning System using Fuzzy Logic", Master thesis, Libyan Academy, 2021.