Journal of Theoretical and Applied Information Technology
31st December 2015. Vol.82. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A FUZZY LOGIC APPROACH TO MANAGE UNCERTAINTY AND IMPROVE THE PREDICTION ACCURACY IN STUDENT MODEL DESIGN

[1]SALISU MUHAMMAD SANI, [2]TEH NORANIS MOHD ARIS, [3]NORWATI MUSTAPHA
[4]MD NASIR SULAIMAN

Department of Computer Science, Faculty of Computer Science and information Technology, University Putra Malaysia, 43400, Serdang, Selangor Malaysia

E-mail: [1]mssaleesu@gmail.com , [2]nuranis@upm.edu.my, [3]norwati@upm.edu.my, [4]nasir@upm.edu.my

ABSTARCT

The intelligent tutoring systems (ITSs) are special classes of e-learning systems developed using artificial intelligent (AI) techniques to provide adaptive and personalized tutoring based on the individuality of each student. For an intelligent tutoring system to provide an interactive and adaptive assistance to students, it is important that the system knows something about the current knowledge state of each student and what learning goal he/she is trying to achieve. In other words, the ITS needs to perform two important tasks, to investigate and find out what knowledge the student has and at the same time make a plan to identify what learning objective the student intends to achieve at the end of a learning session. Both of these processes are modeling tasks that involve high level of uncertainty especially in situations where students are made to follow different reasoning paths and are not allowed to express the outcome of those reasoning in an explicit manner. The main goal of this paper is to employ the use Fuzzy logic technique as an effective and sound computational intelligence formalism to handle reasoning under uncertainty which is one major issue of great concern in student model design.

Keywords: *Intelligent tutoring systems, Fuzzy Logic, Student Model, Uncertainty*

1. INTRODUCTION

The intelligent tutoring systems (ITSs) are generations of Computer based educational systems developed using artificial intelligence techniques (AI), comprising of Bayesian Networks, Neural Networks, Fuzzy Logic, Genetic Algorithm, Ontology, Data Mining etc., that share the same educational goal - providing a teaching strategy that support learning [1, 2]. The emergence of the intelligent tutoring systems in the last four decades has significantly changes the content and practice of teaching and learning in today's educational environments [3].

The most significant of this change is redefining the concept of education far from been just a traditional school setting and has increased the number of participants seeking knowledge from children to almost all adults from various age groups [4]. One of the key features that make intelligent tutoring systems differ from traditional e-learning systems is their ability t observe students actions and draws some useful conclusions from those actions in order to maintain a model of the student [5].The goal of any ITSs is to provide students with interactive assistance with the aim of helping them to achieve maximum learning gain, but before an ITS could do so, it needs to finds out what knowledge the student holds currently and to what extent does he/she intends to move it to the next level. In other words, the ITS need to perform an assessment on the part of the students in order to enable the system plan on how to assist them achieve the desired objectives. Both of these processes are modeling tasks that involve high level of uncertainty especially in situations where the students are not allowed to express their reasoning explicitly. Like most recommender models for non ITSs systems, the student model is a vital component of an intelligent tutoring system that enables the system to observe the interactions it has with the learners and adapt to their needs based on their individuality. Unlike non ITSs systems, the goal of an ITS is to ensure that students learn a target

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

instructional objective at the end of a learning session and this also contributes to a great deal of uncertainty to student modelling because it amounts to making an inference out of the student's actions to determine how well the student understood the target concepts and pass a meaningful decision about the student, this is known as knowledge tracing or assessment [6].

2 RELATED LITERATURES

2.1 Overview of an ITS

One of the key features that make intelligent tutoring systems differ from traditional e-learning systems is their ability observe students actions and draws some useful conclusions from those actions in order to maintain a model of the student cognitive state (the student model)[5]. The goal of any ITSs is to provide students with interactive assistance with the aim of helping the students to achieve maximum learning gain, but before an ITS could do so, it needs to finds out what knowledge the student has already acquired and to what extent does he/she aim to move that knowledge to the next level. In other words, the ITS need to perform what is known as assessment and plan recognition on the part of the student to find out what knowledge he/she has already has and what goal the student is also trying to achieve. Both of these processes are modeling tasks that involve high level of uncertainty especially in situations where the students are allowed to move through various lines of reasoning without being made to explicitly express their reasoning. Like most recommender models for non ITSs systems, the student model is a vital component of an intelligent tutoring system that enable the ITS to observe the interactions it has with the learners and adapt to the needs of those learners. Unlike non ITSs systems, the goal of an ITS is to ensure that students learn a target instructional objective at the end of a learning session and this also contributes to a great deal of uncertainty to student modelling because it amounts to making an inference out of the student's actions to determine how well the student understood the target concepts and pass a meaningful decision about the student, this is known as knowledge tracing or assessment [6].

2.2 Architecture of an ITS

Originally, the general concensus in the literature of the ITS was that, the tutoring system was made up of three basic components; the domain module, the tutor module and the student module [7,8]. These trinity component architechture was extended to be a four component architecture with addition of "the user interface module" [9,10,11].

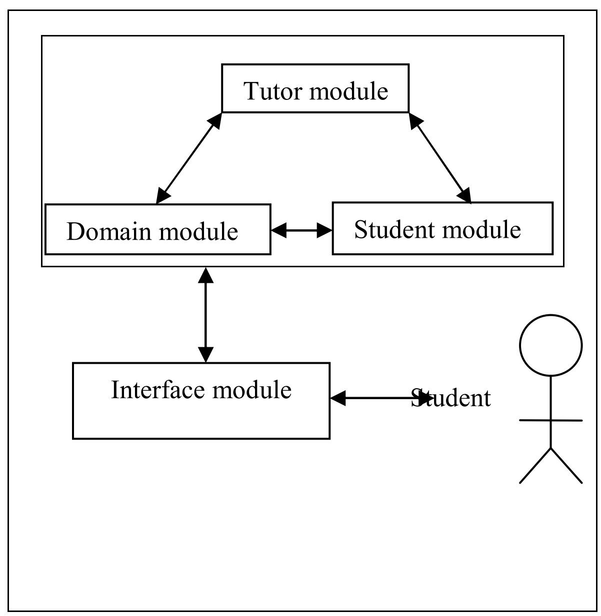

Figure 1Architecture of an ITS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.3 The Student Module

The student model is considered by many researchers as the core component of any ITS. The model is the most dynamic of all the ITS components as it represents how student's knowledge and skills are continuously experimented and updated [12]. It is designed to track and kept as much knowledge about the student's cognitive and affective states as possible as the dynamic processes of teaching and learning progresses. A well designed student model should be effective enough to be able to generate an inferred data both explicit and implicit from and about the learner, process these data to create a profile of the student in terms of his/her knowledge, individuality and learning style, and the model must also be able to manage those data to perform some basic diagnosis in terms of both student's knowledge representation and optimization of pedagogical decisions about the student. A major issue of great concern that arises in student modeling is uncertainty. The intelligent tutoring system build a complete profile of student (student model) based on the observation it makes about the student during the teaching and learning processes, this allow the ITS to pass effective

decision on students based on the outcome of their actions while interacting with the system. When a student model, which is the most vital component of an ITS is so “poor” to the extent that it does not provides a clear picture of the students to fully describe them in terms of their learning styles, characteristics or profiles, then all the decisions of other components of the ITS that depends on the student model such as the tutor or domain models are going to be of poor quality also [1]. One of the powerful and general techniques of decision making that is widely used in different areas of artificial intelligence to handle reasoning under uncertainty is the Fuzzy logic. The idea of Fuzzy logic was discovered by [13]. Since its discovery, the technique has become one of the most widely used computational intelligence approaches that are used to manage uncertainty in most areas artificial intelligence (AI) applications.

2.4 Overview of Fuzzy Logic Technique

Building an intelligent tutoring system is a challenging task; the process involves a number of different aspects such as the teaching, learning, adaptation and control under uncertainty. Fuzzy logic has been an important tool for people working in the field of ITS in particular and Artificial Intelligent (AI) areas in general. Over the past decades, we have witnessed the rapid growth of utilizing Fuzzy logic technique both in theory and applications in various fields of artificial intelligence [14]. Unlike many computational techniques, Fuzzy logic algorithms provide a tool to deal with uncertainty, random noises with relatively modest computational effort. Our world today is full of uncertainty. We go to work every day for example and we experienced different whether conditions and traffic patterns. Every time we park our vehicles at the same parking area, but we may not be sure to park those vehicles at exactly the same spot as we did a day before or even some few hours ago. Fuzzy logic algorithms offer many attractive features and they have been gaining popularity for solving many real engineering application problems especially the problems of systems which are highly non-linear, which involves many parameters and many of these parameters are

changing or drifting in time. In this paper, we describe how we use Fuzzy logic technique as the underlying framework to model both the student and tutor components of an ITS.

2.4.1 Architecture of a Fuzzy logic system

The general architecture of a Fuzzy logic system comprises of three fundamental components namely, Fuzzifier, Inference engine and defuzzifier.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

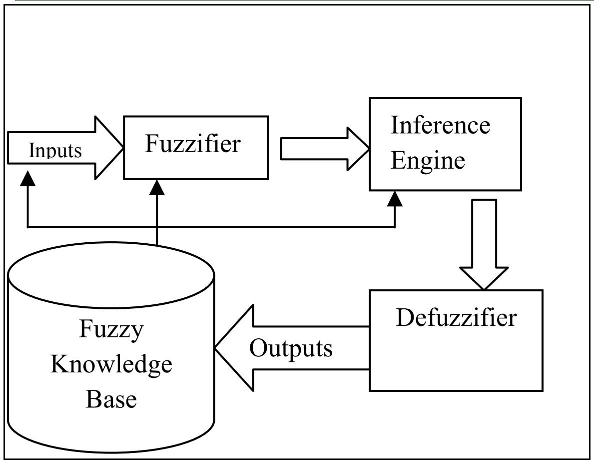

Figure 2: Architecture of a Fuzzy Logic System

2.4.2 Fuzzifier

At the Fuzzifiaction stage, the fuzzifier converts the input variables also known as the crisp inputs to a linguistic variables or fuzzy terms using the membership functions stored in the fuzzy knowledge base. The crisp inputs are mapped into the membership functions on the antecedent part defined by Fuzzy rules to obtain the corresponding fuzzy terms or linguistic variables and a corresponding degree of membership for each linguistic variable. The use of membership function in fuzzification process has created more alternatives to assign membership values to fuzzy terms than there are to assign probability density values to random variables [15]. Depending on their shapes, membership functions can take different form of representations but they all serve a common objective, transforming the crisp inputs to their equivalent fuzzy variables and corresponding membership grades. The most commonly used membership functons in fuzzification processes are Trapezoidal, Triangular, Bell curves, Gaussian and Sigmoidal membership functions.

2.4.3 Inference engine

The kernel of decision making process in Fuzzy logic system (FLS) is the Fuzzy inference engine (FIE). The FIE has the capability of simulating human decision making by performing approximate reasoning to achieve a desired control strategy. In practice, the main goal of a Fuzzy inference is to use all available knowledge to make deductive reasoning which in turn allow the system to infer conclusions based on those available body of facts and knowledge.

2.4.5 Defuzzification

The defuzzification stage is the final stage in the FLS circle. It is the process of converting fuzzy variables and their corresponding membership degrees obtained from the output of an inference engine to quantifiable crisp values. The advantage of using the defuzzifier is to provide a non-fuzzy decision or control from the inferred results of the inference engine. There are various techniques that are employed to interpret membership degrees into a specific decision or real values [16]. These includes the centroid method [17], the centroid method is sometimes referred to as center of gravity or centre of area method; the weighted average method, this method operates by computing the weight of each function by its corresponding membership value and the technique works better for symmetrical output membership functions; the mean - max membership technique; the Max-membership principle also known as the height method; the centre of sums, , the advantage of the centre of sums technique is that it always appear to be more faster than many of the defuzzification techniques; the centre of largest area defuzzification technique, it is widely used for handling Fuzzy sets with convex sub regions; First or last of maxima technique, this approach determine the smallest value of the domain with maximized membership degree.

3 THE FUZZY STUDENT MODEL DESIGN

The main objective of this paper is to use Fuzzy approximate reasoning technique to design, implement and test the prediction accuracy of a Fuzzy student models (FSM) and compare the results of this predictions with a Bayesian student model [1]. A Fuzzy student model is a model that is created using fuzzification engine to describe or provides a representation of each student in terms of a relative fuzzy term(s) and corresponding degree(s) of membership. Three Fuzzy logic systems FLS1, FLS2 and FLS3 are designed with three distinct membership function representations. The fuzzification stage in each of the three FLSs is used to generate three Fuzzy student models FSM1, FSM2 and FSM3 respectively. The membership functions of FLS1, FLS2 and FLS3 are designed using six suitable Fuzzy terms or linguistic variables “poor”, “weak”, “average”, “good”, “very good”

Journal of Theoretical and Applied Information Technology
31st December 2015. Vol.82. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and “excellent” [13]. To implement and test these approximate student model designs, the fuzzifiers of FSL1, FSL2 and FSL3 make use of the crisp inputs, the function Fv(Dc) in the student_model_data_1 (table 1). The student_model_data_1 is an instance of the original student model generated by an adaptive AC-Ware Tutor system (Grubisic, 2012). The adaptive AC-Ware Tutor system has the ability to interact and administer a knowledge test to students on 73 domain concepts that are defined in the domain of “Computer as a system”. The student's score, which is a function (Fv) on each of the 73 domain concepts (Dc), are recorded as instances of student’s knowledge test results by the AC-Ware system. The student’s scores Fv(Dc) (Table 1) are expressed in values ranging between the intervals 0 and 1. Mapping each student’s score in the function Fv into the membership function for each fuzzifier of the three FLS models, the three Fuzzy student models FSM1, FSM2 and FSM3 are obtained. Each of the three FSM is a profile expressed in Fuzzy term(s) and corresponding degree of memberships.

Table 1: Part of Student_Model_Data_1

Concept (Dc)	Fv(Dc)
1.44MB	0.125
Application software	0.375
Arithmetic operation	0
Arithmetic-logic unit	0.375
Assembler	0
Basic	0
C	0.25
Central unit	0.5
Central processing unit	0.5
Data entry	0.875
Device for communication	0.916
Disjunction	0.083
Diskette	0.25
DOS	0
Fortran	0.125
I gate	0
Input unit	0.4167
Information	0.125
Instruction	0.4375
Interpreter	0
Language translators	0
Logic gate	0.0416
Mass memory	0.5
Modem	0.5
Model of Computer system	0.083
Mouse	0
Negation	0.625
Network Card	0.375
Output unit	0.2917
Operating system	0
Pascal	0.125
Programming language	0
Software support	0.6875

3.1 Fuzzy Student Model 1

To generate the first fuzzy student model FSM1, we run all the crisp inputs from the instance of student model, the student_model_data_1 into the FLS1 fuzzifier (Figure 3). This enebles us to map all the crisp values in the data and transform each value to its corresponding fuzzy term and membership function, the result of this transformation is the FSM1. Figure 3 is the representation of membership function for FLS1 fuzzifier containing all six variable terms “poor”, “weak”, “average”, “good”, “very good” and “excellent”, in which six membership functions are created. Mapping each crisp value from table 1 into the FLS1 fuzzifier, we generate the first Fuzzy student model, the FSM1 (Table 2) as well as it’s structure (Table 3).

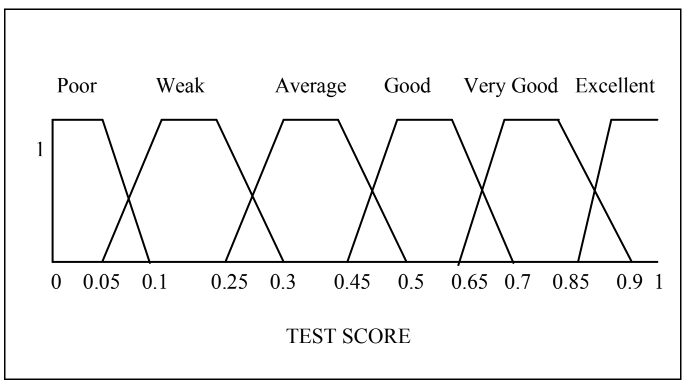

Figure3: FLS1 fuzzifier

Table 2: Part of the Fuzzy Student Model 1

Concept (Dc)	Fuzzy Term	Membership Deree
App Sofstware	Average	1.0
Arith Operation	Poor	1.0
Arith Log Unit	Average	1.0
Assembler	Poor	1.0
Basic	Poor	1.0
C	Weak	1.0
Central Unit	Good	1.0
Central Proc Unit	Good	1.0
Device for Comm	Excellent	1.0

Journal of Theoretical and Applied Information Technology
31st December 2015. Vol.82. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Disjunction	Poor/Weak	0.3/0.6
Dos	Poor	1.0
Fortran	Weak	1.0
Input unit	Poor	1.0

Table 3: Structure of FSM1

Fuzzy Term	Frequency	Percentage (%)
Poor	31	42.5
Weak	18	24.7
Average	9	12.3
Good	10	13.7
Very Good	2	2.7
Excellent	3	4.1

3.2 Fuzzy Student Model 2

To generate the second fuzzy student model FSM2, we run all the crisp inputs from the instance of student_model_data_1 into the FLS2 fuzzifier funtion (Figure 4). This enebles us to map all the crisp values in the data and transform each value to its corresponding fuzzy term and membership degree, the result of this transformation is the FSM2. Figure 4 is the representation of the six membership functions for FLS2 fuzzifier containing all six variable terms "poor", "weak", "average", "good" and "very good" and "Excellent". Mapping each crisp value from table 1 into FLS2 fuzzifier, we generate the second fuzzy student model, the FSM2 (Table 4) as well as its structure (Table 5).

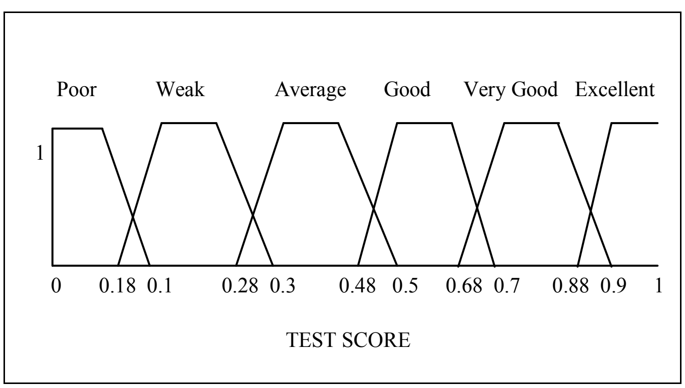

Figure 4: FLS2 Fuzzifier

Table 4: Part of the Fuzzy Student Model 2

Concept (Dc)	Fuzzy Term	Membership Deree
App Software	Average	1.0
Arith Operation	Poor	1.0
Arith Log Unit	Average	1.0
Assembler	Poor	1.0
Basic	Poor	1.0
C	Weak	1.0
Central Unit	Average	1.0
Central Proc Unit	Average	1.0
Device for comm	Very Good/Excellent	0.7/0.2
Disjunction	Poor	1.0
Dos	Poor	1.0
Fortran	Poor/Weak	0.5/0.4
Input unit	Average	1.0

Table 5: Structure of FSM2

Fuzzy Term	Frequency	Percentage (%)
Poor	38	52
Weak	12	16.4
Average	16	22
Good	3	4.1
Very Good	3	4.1
Excellent	1	1.4

3.3 Fuzzy Student Model 3

To generate the third fuzzy student model FSM3, we run all the crisp inputs from the student_model_data_1 into the FLS3 fuzzifier funtion (Figure 5). This enebles us to map all the crisp values in the data and transform each value to its corresponding fuzzy term and degree of membership, the result of this transformation is the FSM3 model. Figure 5 is the representation of six membership functions for FLS3 fuzzifier containing six variable terms "poor", "Weak" "average", "good", "very good" and "Excellent". Mapping each crisp value from table 1 enable us to generate the third Fuzzy student model, the FSM3 model (Table 6) as well as it's structure (Table 7).

Journal of Theoretical and Applied Information Technology
31st December 2015. Vol.82. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

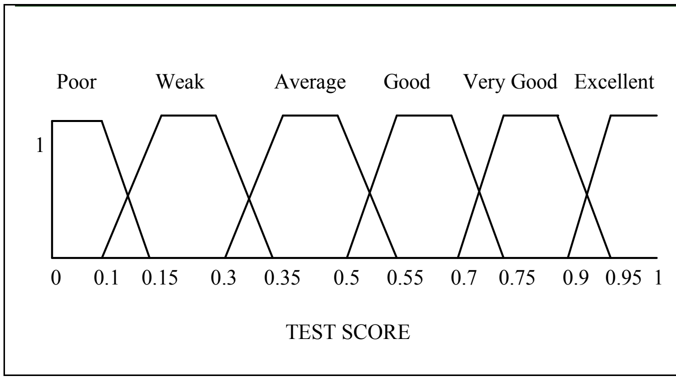

Figure 5: FLS3 Fuzzifier

Table 6: Part Of The Fuzzy Student Model 3

Concept (Dc)	Fuzzy Term	Membership Deree
App Sofstware	Average	1.0
Arith Operation	Poor	1.0
Arith Log Unit	Average	1.0
Assembler	Poor	1.0
Basic	Poor	1.0
C	Weak	1.0
Central Unit	Good	1.0
Central Proc Unit	Good	1.0
Device for comm	Excellent	1.0
Disjunction	Poor/Weak	0.8/0.2
Dos	Poor	1.0
Fortran	Weak	1.0
Input unit	Average	1.0

Table 7: Structure of FSM3

Fuzzy Term	Frequency	Percentage
Poor	33	45.2
Weak	16	22
Average	9	12.3
Good	11	15
Very Good	3	4.1
Excellent	1	1.4

3.4 Testing the Prediction Accuracy

We have successfully used the student_model_data_1 in three different fuzzifiers for the three fuzzy logic systems FLS1, FLS2 and FLS3 to generate three fuzzy student models FSM1, FSM2 and FSM3 respectively. In order to identify which of these FSMs provides a better and more accurate result in its prediction, we use the next stage of Fuzzy logic system, the Fuzzy inference mechanism to set and test the prediction accuracy of the fuzzy student models.

3.4.1 Setting the pieces of evidences

The values of the functions Fv(Dc) (table 1) are not only meant for determining the fuzzy student models, they can as well be used to set the pieces of evidences that will enable us to test the prediction accuracy of each Fuzzy student model. In each of the three Fuzzy student models FSM1, FSM2 and FSM3, we will observe three different ways of setting the pieces of evidences by heuristic selection of nine different values of the function Fv(Dc) to be used as a threshold and this enable us to generate nine different Fuzzy models. And since the arbitrary threshold values are heuristically determined, then each one of them is analysed in order to identify which among them is best for setting up the pieces of evidence. The observed gained predictions from these pieces of evidences are then compared with the Bayesian student model [7].

3.4.2 Test 1 FSM1

Let Dc be any concept in the domain knowledge and let the function $F_v(Dc)$ represent the value of knowledge test score on each domain concept. If $Fv(Dc) \geq 0.85$, then we set the evidence on the domain concept Dc as "pass" because by mapping the value 0.85 into FLS1 fuzzifier, we got a fuzzy grade "Excellent" . Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.85 i.e. Ʉ $F_v(Dc)$, if $1-F_v(Dc) \geq 0.85$ then it is enough

for us to set the evidence on that domain concept as "fail".

Journal of Theoretical and Applied Information Technology
31st December 2015. Vol.82. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

For example, in the instance data, the student_model_data_1, there exist a domain concept called "Pascal" whose value of the function $F_v(Dc)$ = 0.0416 i.e. $F_v(Pascal)$ = 0.0416. It is obvious that the fuzzy compliment of this concept $\mu'(F_v(Pascal))$ = 1- $F_v(Pascal)$ = 1-0.125 = 0.875 which is greater than 0.85. This allows us to set the evidence on the concept "Pascal" to a "failed" status and based on this example all three concepts with a linguistic term "Excellent" are pieces of evidences, therefore we set the failed evidence on 3 domain concepts (4% of all three domain concepts are pieces of evidences).

3.4.3 Test 2 for FSM1

Let Dc be any domain concept, if $Fv(Dc) \geq 0.8$, then we set the evidence on the domain concept Dc as "pass" because by mapping the value 0.8 into FLS1 fuzzifier, we got a fuzzy grade "Very Good" . Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.8 i.e. Ʉ Fv(Dc), if $1-F_v(Dc) \geq 0.8$ then we are certain to set the evidence on that domain concept as "fail".

For example, in the instance data, the student_model_data_1, there exist a domain concept called "Capacity" whose value of the function $F_v(Dc)$ = 0.125 i.e. Fv(Capacity) = 0.125. It is obvious that the fuzzy compliment of this concept $\mu'(Fv(Capacity))$ = 1- Fv(Capacity) = 1 - 0.125 = 0.875 which is greater than 0.8. This allows us to set the evidence on the concept "Capacity" to a "failed" status and based on this example all 5 concepts are pieces of evidences, therefore we set the failed evidence on 5 domain concepts (7% of all five domain concepts are pieces of evidences).

3.4.4 Test 3 FMS1

Let Dc be any domain concept, if $Fv(Dc) \geq 0.6$, then we set the evidence on the domain concept Dc as "pass" because by mapping this selected value 0.6 into FLS1 fuzzifier, we got a fuzzy grade "Good" . Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.6 i.e. Ʉ $F_v(Dc)$, if $1-F_v(Dc) \geq 0.6$ then we are certain to

set the evidence on that domain concept as "fail".For example, in the instance data, the student_model_data_1, there exist a domain concept called "Data Transfer" whose value of the function $F_v(Dc)$ = 0.375 i.e. F_v(Data Transfer) = 0.375. It is obvious that the fuzzy compliment of this concept $\mu'(f_v(\text{Data transfer}))$ = 1-0.375 = 0.625, which is greater than 0.6. This allows us to set the evidence on the concept "Data transfer" to a "failed" status and based on this example all 15 concepts are pieces of evidences, we set the failed evidence on all 15 domain concepts (21% of all fifteen domain concepts are pieces of evidences).

3.4.5 Test 1 for FSM2

Let Dc be any concept in the domain knowledge and let the function Fv(Dc) represent the value of knowledge test score on each domain concept. If $Fv(Dc) \geq 0.9$, then we set the evidence on the domain concept Dc as "pass" because by mapping the value 0.9 into FLS1 fuzzifier, we got a fuzzy grade "Excellent". Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.9 i.e. Ʉ Fv(Dc), if $1-Fv(Dc) \geq 0.9$ then it is enough for us to set the evidence on that domain concept as "fail".

For example, in the instance data, the student_model_data_1, there exist a domain concept called "Disjunction" whose value of the function Fv(Dc) = 0.083 i.e. Fv(Disjunction) = 0.083. It is obvious that the fuzzy compliment of this concept $\mu'(Fv(Disjunction))$ = 1- Fv(Disjunction) = 1 - 0.083 = 0.917 which is greater than 0.9. This allows us to set the evidence on the concept Disjunction to a "failed" status and based on this example the one concept with a linguistic term "Excellent" is our piece of evidences we set the failed evidence on 1 domain

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

concepts (1% of one domain concept are pieces of evidences).

3.4.6 Test 2 FSM2

Let Dc be any concept in the domain knowledge and let the function Fv(Dc) represent the value of knowledge test score on each domain concept. If $F_v(Dc) \geq 0.7$, then we set the evidence on the domain concept Dc as "pass" because by mapping the selected value 0.7 into FLS2 fuzzifier, we got a fuzzy grade "Very Good". Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.7 i.e. $\forall\, F_v(Dc)$, if $1\text{-}F_v(Dc) \geq 0.7$ then we are certain to set the evidence on that domain concept as "fail".

For example, in the instance data, the student_model_1, there exist a domain concept called "Hard Disk" whose value of the function $F_v(Dc) = 0.25$ i.e. $F_v(\text{Hard Disk}) = 0.25$. It is obvious that the fuzzy compliment of this concept $\mu'(X_v(\text{Hard Disk})) = 1\text{-}\ F_v(\text{Hard Disk}) = 1\text{-}0.25 = 0.75$ which is greater than 0.7. This allows us to set the evidence on the concept "Hard Disk" to "failed" and based on this example, all 7 domain concepts are pieces of evidence and therefore we set the failed evidence on this 7 domain concepts (10% of all seven domain concepts are evidences).

3.4.7 Test 3 FSM2

Let Dc be any concept in the domain knowledge and let the function $F_v(Dc)$ represent the value of knowledge test score on each domain concept. If $F_v(Dc) \geq 0.65$, then we set the evidence on the domain concept Dc as "passed" because by mapping the value 0.65 into FLS2 fuzzifier, we got a fuzzy grade "Good" . Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.65 i.e. $\forall\, F_v(Dc)$, if $1\text{-}F_v(Dc) \geq 0.65$ then we are certain to set the evidence on that domain concept as "failed".

For example, in the instance data, the student_model_data_1, there exist a domain concept called "Computer system" whose value of the function $F_v(Dc) = 0.33$ i.e. $F_v(\text{Computer system}) = 0.33$. It is obvious that the fuzzy compliment of this concept $\mu'(F_v(\text{Computer system})) = 1\text{-}\ F_v(\text{Computer system}) = 1\text{-}0.33 = 0.67$ which is greater than 0.65. This allows us to set the evidence on the concept "Computer system" to a "failed" status and based on this example, all 15 domain concepts are pieces of evidences concepts (15% of all seven domain concepts are evidences).

3.4.8 Test 1 FSM3

Let Dc be any concept in the domain knowledge and let the function $F_v(Dc)$ represent the value of knowledge test score on each domain concept. If $F_v(Dc) \geq 0.95$, then we set the evidence on the domain concept Dc as "pass" because by mapping the value 0.95 into FLS3 fuzzifier, we got a fuzzy grade "Excellent" . Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.95 i.e. $\forall\, F_v(Dc)$, if $1\text{-}F_v(Dc) \geq 0.95$ then we are certain to set the evidence on that domain concept as "fail".

For example, in the instance data, the student_model_data_1, there exist a domain concept called "Logic gate" whose value of the function $F_v(Dc) = 0.0416$ i.e. $F_v(\text{Logic gate}) = 0.0416$. It is obvious that the fuzzy compliment of this concept $\mu'(F_v(\text{Logic gate})) = 1\text{-}\ F_v(\text{Logic gate}) = 1\text{-}0.0416 = 0.9584$ which is greater than 0.95. This allows us to set the evidence on the concept "Logic gate" to "failed" status and based on this example, the one concept with a linguistic term “Excellent” is our piece of evidence, therefore we set the failed evidences on 1 domain concepts (1% of one domain concepts are pieces of evidences).

3.8.9 Test 2 FSM3

Let Dc be any concept in the domain knowledge and let the function $F_v(Dc)$ represent the value of knowledge test score on each domain concept. If $F_v(Dc) \geq 0.75$, then we set the evidence on the domain concept Dc as "pass" because by mapping the value 0.75 into FLS3 fuzzifier, we got a fuzzy grade "Very Good" . Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.75 i.e. $\forall\, F_v(Dc)$, if $1\text{-}F_v(Dc) \geq 0.75$ then we are certain to set the evidence on that domain concept as "failed".

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

For example, in the instance data, the student_model_data_1, there exist a domain concept called "Data storage" whose value of the function $F_v(Dc) = 0.25$ i.e. F_v(Data storage) = 0.25. It is obvious that the fuzzy compliment of this concept $\mu'(X_v$(Data storage)) = 1- F_v(Data storage) = 1-0.25 = 0.75 which is equal to 0.75. This allows us to set the evidence on the concept Arithmetic logic unit to "failed" and based on this example, all 4 domain concepts are pieces of evidences (6% of all 4 domain concepts are evidences).

3.4.10 Test 3 FSM3

Let Dc be any concept in the domain knowledge and let the function $F_v(Dc)$ represent the value of knowledge test score on each domain concept. If $F_v(Dc) \geq 0.55$, then we set the evidence on the domain concept Dc as "passed" because by mapping the value 0.55 into FLS3 fuzzifier, we got a fuzzy grade "Good" . Similarly, if the fuzzy compliment of any domain concept in the student_model_data_1 is greater or equal 0.55 i.e. Ʉ $F_v(Dc)$, if $1-F_v(Dc) \geq 0.55$ then we are certain to set the evidence on that domain concept as "failed".

For example, in the instance data, the student_model_data, there exist a domain concept called "Network Card" whose value of the function $F_v(Dc) = 0.375$ i.e. F_v(Network Card) = 0.375. It is obvious that the fuzzy compliment of this concept $\mu'(F_v$(Computer system)) = 1- F_v(Computer system) = 1-0.375 = 0.625 which is greater than 0.55. This allows us to set the evidence on the concept "Computer system" to a "failed" status and based on this example, all 15 domain concepts are pieces of evidences concepts (21% of all 15 domain concepts are evidences).

3.5 Result of the Prediction Accuracy

The student_model_data_1 is the representation of the actual student knowledge from the AC-Ware Tutor system. Based on that data, it is apparent to understand which concept the student has mastered well enough. We designed nine different ways of setting pieces of evidences to test the prediction accuracies of the Fuzzy student models and that allow us to formulates nine models that we will analyze and justify which one of them best predicts the actual student's knowledge, the student_model_1. For each of the nine models, the percentage of overlap relative to the actual student knowledge as contained in student_model_1 is calculated and expressed in terms of three parameters; prediction match, prediction indication and prediction miss. If the difference in absolute value between the fuzzy compliment of the function Fv and the value of the prediction threshold differ by less than or equal to 0.1, then we conclude that we have a "prediction match". Similarly, if the difference in absolute value between the Fuzzy compliment of the function Fv and the value of the prediction threshold differ by more than 0.1 and less or equal to 0.2, then we conclude that we have a "prediction indication". If the difference in absolute value between the fuzzy compliment of the function Fv and the value of the prediction threshold differ by more than 0.1, then we conclude that we have a "prediction miss".

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 7 Result Of Fuzzy Student Model Prediction Testing

Model	Fuzzy Student Model	Evidence settin	Match ≤ 0.1 (%)	Indication 0.1 ≤ 0.2 (%)	Miss ≤ 0.1 (%)
Model1	FSM1	Test 1	26	44	30
Model2	FSM2	Test 1	51	17	32
Model3	FSM3	Test 1	51	15	34
Model4	FSM1	Test 2	23	52	25
Model5	FSM2	Test2	25	20	55
Model6	FSM3	Test 2	18	22	60
Model7	FSM1	Test 3	22	19	59
Model8	FSM2	Test 3	27	11	62
Model9	FSM3	Test 3	20	22	5

4. CONCLUSION

In this research, we have successfully use an instance of a student model data (table 1) obtained from an adaptive AC-Ware Tutor system [19] to design, implement and produce a fuzzy student model whose prediction accuracy proved to more better and more flexible in providing grounds for effective pedagogical decision than a Bayesian student model [7]. This research also describes how fuzzy logic technique is used as a frame work and address uncertainty which is one of the most serious issue facing student model design in an ITS. A Bayesian network (BN) student model [1] has already been implemented using the same test data, the Student_Model_Data_1. The Bayesian model uses dichotomous variables approach true or false in determining the probabilities of nodes in the domain knowledge graphs (DKGs) in the Bayesian network structure. This dichotomous nature of the Bayesian Network model contributed to its lack of flexibility as the model can only allow each concept in the domain knowledge to be represented by either a true or false variable but nothing comes in between the two when creating the student profile. In contrast, the fuzzy student model allows for multi variable representation of the domain knowledge concepts using six interval scales represented by six Fuzzy terms or linguistic variables namely, "poor", "weak", "average", "good", "very good" and "excellent". This contributes to a great deal in making the Fuzzy student model approach to be more flexible handling the uncertainty issue and also making effective pedagogical decision about the student.

Looking at the results from the prediction table, it is easier for one to observe that Model2 has the most and best overlap with the actual student's knowledge produced by the adaptive AC-Ware Tutor system (Table 1). This is because the model has the highest number of prediction matches and very low percentage of prediction misses in relation to other models. It is enough therefore for us to conclude that Model2 is the best Fuzzy student model for predicting the actual student knowledge. On the other hand, if we compare the result of this research with the result of the existing model, the Bayesian student model [7], it is also clear for us to observe the significant improvement achieved with the Fuzzy student model over the existing Bayesian student model. The best prediction match for the existing model is 36% where as with the Fuzzy student model we are able to achieve a prediction match of 51%, although the two models both produces a prediction miss of 32%, it is obvious to conclude that reducing the percentage of the prediction miss in both two models remains one big challenge that open a huge window of improvement to both two models.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1] A. Grubišić, "Ontology based approach to Bayesian student model design", Expert Systems with Applications, vol. 40, 2013, pp 5363 – 5371.

[2] R. Nkambou, Advances in Intelligent Tutoring Systems, Springer-Verlag Berlin Heidelberg 2010.

[3] L. Xin, L. Soh, "A literature review on learner control strategies in Software tutoring", Journal of Science & Engineering, (2003), CSE Technical Report, Paper 71.

[4] J. Kay, Learner Control User Modeling & User-Adaptation Interaction, Kluwer Academic Publishers. Netherlands.

[5] V. J. Shute, J. Psotka, Intelligent tutoring systems: Past, Present and Future, Jonassen ed. 1996.

[6] Conati C., Gertner A, K. VanLehn, "Using Bayesian Networks to Manage Uncertainty In Student Modelin", Journal of User Modeling & User Interaction vol. 12, 2002, pp 371 – 417.

[7] A. Barr, E. Faigenbaum, The handbook of artificial intelligence: Volume I and II, Kaufmann Press: 1982.

[8] A. Bonet, Artificial Intelligence: Premise and Performance. London: Prentice Hall 1985.

[9] E. Wenger, Artificial Intelligence and tutoring systems. Kaufmann Press: 1987.

[10] H. L. Burns, C. G. Capps, Foundations of intelligent tutoring systems. Lawrence Erlbaum Associates: 1988.

[11] H. Mandl, A. Lesgold, Learning issues for intelligent tutoring systems. Spring NY: 63-79. 1988.

[12] H. S. Nwanna Intelligent tutoring systems: an Overview. Artificial Intelligent Review, vol. 4, 1990, pp 251-277.

[13] L. A. Zadeh, "Fuzzy Sets*" Journal of Information & Control, vol. 8, 1965, pp338 – 353.

[14] H. Li, M. Gupta, Fuzzy Logic and intelligent tutoring systems. Academic Publishers, Boston 1995.

[15] D. Dubois and H. Prade, 1980, Fuzzy Sets and Systems: Theory and Application, Academic Press,1980.

[16] J. Thimothy, Fuzzy Logic with Engineering Applications, Wiley Publishers, 2009.

[17] M. Sugeno "Fuzzy Identification of Systems and its Applications to Modeling and Control" IEEE Transaction on Systems, Man, and Cybernetics, Vol. SMC-15, pp 116-133, 1985.

[18] L. Goel, "Fuzzy Logic Representation of Student Model", Springer Vergal, 2011.

[19] A. Grubišić, Adaptive Student Knowledge Acquisition Model in e-learning Systems. PhD Thesis, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. 2012.