

Fuzzy Student Model in InterMediActor Platform

Alenka Kavcic

*Faculty of Computer and Information Science, University of Ljubljana,
Tržaška 25, 1000 Ljubljana, Slovenia*

Rafael Pedraza-Jiménez, Harold Molina-Bulla, Francisco J. Valverde Albacete,
Jesús Cid-Sueiro, Angel Navia-Vázquez

*Department of Signal Processing and Communications, University Carlos III of Madrid,
Avda. de la Universidad 30, 28911 Leganés, Spain*

E-mail: alenka.kavcic@fri.uni-lj.si, {rpedraza, hmolina, fva, jcid, navia}@tsc.uc3m.es

Abstract. *The paper deals with personalization of navigation in the educational content, introduced in a competence-based instructional design system InterMediActor. The system constructs an individualized navigation graph for each student and thus suggests the learning objectives the student is most prepared to attain. The navigation tools rely on the graph of dependencies between competences, and the student model. We use fuzzy set theory for dealing with uncertainty in the assessment of students: the marks of assessment tests are transformed into linguistic terms, which are assigned to linguistic variables. Fuzzy IF-THEN rules are applied to obtain the appropriate categories of competences in the navigation graph.*

Keywords. Educational systems, fuzzy inference, personalized navigation, user model

1. Introduction

Personalization of instruction in educational systems can significantly improve the teaching process. It has been shown that the best method of teaching is individualized tutoring [2]. Individualized instruction considers the needs of the students: the students can control the pace at which they progress through the course material and the materials they use are suited to their cognitive skills and learning styles [2].

The ability to adapt the instruction to the needs of an individual user is important in today's educational systems. Different techniques can be used to offer system's adaptation, navigation support being one of them. Curriculum sequencing or direct guidance is one of the oldest techniques, widely used in intelligent tutoring systems. This technique helps

each student to find an individualized optimal path through the learning material [1]. The personalization of navigation is done through the student model, which collects vital information on each individual student.

To assure personalized navigation through the course content was also our main objective. We set up a navigation assistance in the e-learning platform InterMediActor [8], which relies on a prediction model built upon the information gathered from the student's previous interaction with the system. The navigation graph, which is constructed for each individual student, is updated as the student proceeds with the course.

The paper is focused on fuzzy user modelling in a competence-based instructional system. First we present a computer-based distance education platform InterMediActor, for which the fuzzy student model is developed. Then we describe the fuzzy inference mechanism, which is the core of our model. It employs linguistic variables and linguistic rules for deducing information that is relevant to a particular student.

2. InterMediActor platform

InterMediActor [6, 8] is an instructional design system, which provides an environment for instructional content design, production, and reuse, as well as student evaluation. It relies on the concept of a competence, which is the educational equivalent of a grounded learning objective aiming at some learning outcome. The instructional model in InterMediActor involves the decomposition of learning objectives into a hierarchical structure of corresponding competences. The proposed method for developing educational content has two phases:

- top-down analysis of learning objectives, and
- bottom-up synthesis of competences.

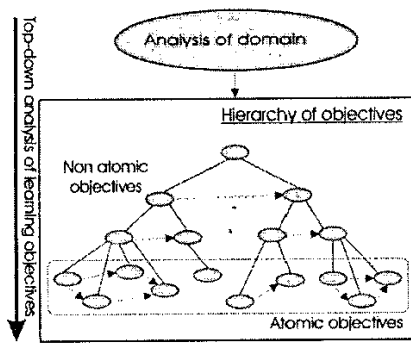


Figure 1. Top-down analysis of learning objectives

Top-down analysis, which is conceptually depicted in Fig. 1, proceeds by partitioning and refining over-arching, all-encompassing learning objectives into more detailed and concrete ones. The process ends when atomic objectives are identified. The result of this analysis is a heterogeneous graph of dependences between learning objectives with two types of relationship: *part-of relationship* between one objective and its sub-objectives, and *depends-on relationship* between objectives that are parts-of the same (more general) objective. The former is described as a tree-like hierarchy (hierarchy graph) and the latter as a finite graph (graph of dependencies).

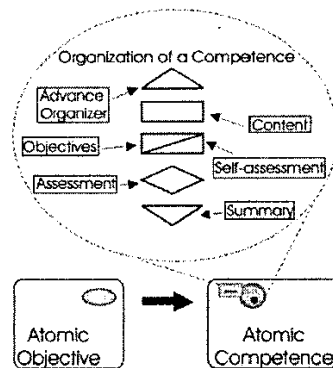


Figure 2. Transforming an atomic objective into an atomic competence

After performing the top-down analysis, a set of atomic competences is extracted out of the learning objectives (see Fig. 2). Each atomic objective is extended into one atomic competence that among other information consists of:

- *advance organizer* (introduces the topic to be learned),
- *content to be learned* (can be a complex multimedia object),
- *summary* (states what the learning outcome should be),
- *self-assessment* and *final-assessment tests* (based on the introduced content),
- *pre-requisites of the competence* (which have to be grasped before trying to understand this particular competence; determine the *depends-on* relationship between competences of the same granularity).

Once the atomic competences are identified, more complex competences can be synthesized following the *part-of* relationship in the hierarchy graph. In this bottom-up approach, atomic competences are aggregated as the content of a higher-level competence, for which the advance organizer, the summary and assessment tests are further provided. Such aggregated, second-order competences can then be used to synthesize new ones, until each learning objective in the analysis coalesces into a competence and the resulting content has the shape of a competence graph.

3. Fuzzy logic and fuzzy inference

The theory of fuzzy logic is a part of a broader fuzzy set theory [9]. It was introduced in the 1960's as a means to model the uncertainty of natural language. Fuzzy logic extends conventional Boolean logic to handle the concept of partial truth (truth values between "completely true" and "completely false"). The partial truth takes a continuous range of truth values and is determined by the membership function, which takes values from the closed interval [0,1].

Fuzzy inference (approximate reasoning) is based on fuzzy logic and resembles human reasoning in its use of approximate information and uncertainty to generate decisions. It consists of one or more fuzzy rules (implications), a set of facts, and a conclusion [4, 5].

The fuzzy production rules, which connect premises with conclusions, have the form of IF-THEN rules:

$$\text{IF } X_1 = A_1 \text{ and } X_2 = A_2 \text{ and } \dots \text{ and } X_n = A_n \quad (1) \\ \text{THEN } Y = B,$$

where X_i and Y are linguistic variables, and A_i and B are linguistic terms. The IF part of the rule

is the premise, the THEN part of the rule is the conclusion.

Fuzzy inference matches fuzzy facts against fuzzy conditions and assigns a fuzzy output set. In contrast to crisp rules, each rule is allowed to fire in a fuzzy system. Consequently, the order in which the rules execute is not important.

Fuzzy reasoning involves the following four processes [7]:

- fuzzification,
- aggregation,
- composition, and
- defuzzification.

Crisp input values are fuzzified into linguistic values and related to the linguistic variables. This process is called fuzzification. Linguistic values are words (linguistic terms) with associated degrees of membership, thus we need to specify the appropriate membership functions.

Aggregation is a process of computing the value of the rule's premise. Each condition in the IF part of the rule is assigned a degree of truth based on the degree of membership of the corresponding linguistic value. The degree of truth of the IF part is computed as either the minimum (MIN) or the product (PROD) of the degrees of truth of the conditions. This degree of support for the rule is assigned to the degree of truth of the THEN part.

The process of computing the values of the rule's conclusion is called composition. The degree of truth of each linguistic term of the output linguistic variable is calculated using either the maximum (MAX) or the sum (SUM) of the degrees of truth of the rules with the same linguistic terms in the THEN part.

Defuzzification of linguistic values of the output linguistic variable is the last step in this process. One of the common techniques used is Centre of Maximum (CoM) method, where the crisp value is computed as the best compromise for the most typical values of each linguistic value and respective degrees of membership.

4. InterMediActor and student model

When introducing the user model in the InterMediActor platform, our main objective was to provide the personalization of navigation through the course content. This was achieved by constructing the personal navigation graph for each student.

The graph is displayed with competences as nodes in different colours: grey for already learned competences, red indicates forbidden

competences (at the moment), and green implies recommended competences (in range from dark to light green, indicating more or less recommended competence). The arcs in the navigation graph show all possible paths this student can take from the currently studied competence to all (more/less) recommended ones (green nodes).

The navigation graph is based on the graph of dependences, which is constructed as a result of the analysis of learning objectives. In the process of its construction, we rely on the student model, which collects the relevant data for each individual learner: the marks (both, passed and failed) obtained in assessment tests for each competence. Therefore, the proposed student model consists of a set of pairs (*competence, mark*) and can be regarded as an extended overlay over the set of competences.

There is no assessment of competences prior to learning with the system. Hence, the student model initializes with no marks for all the competences (either atomic or aggregated).

5. Fuzzy student model

In navigation graph construction, we employ the student model to assign categories to competences regarding their levels of recommendation. The process of inferring from assessment marks involves some imprecision; thus the fuzzy set theory [9] is used as a mathematical theory for expressing this uncertainty.

For each competence we calculate a numerical level of recommendation and assign a corresponding category the competence belongs to. The category determines the colour of the node in the navigation graph.

When the student takes the final-assessment test for one competence, also the categories of related competences may change. Therefore, new categories are calculated for the tested competence and all competences that depend on it (all competences in the graph of dependencies which can be directly accessed from the given competence).

5.1. Level of recommendation

Each competence (with regard to a particular student) is assigned one of the following five categories: *learned*, *more recommended*, *recommended*, *less recommended*, or *forbidden*. These categories match the linguistic (fuzzy)

terms of the linguistic variable *level of recommendation*.

The calculated *level of recommendation* for the competence depends on the level of difficulty of the competence, the mark the student obtained in the final-assessment test for this competence (if any), and on how well the student knows/has learned the prerequisites of this competence (if there are any).

5.2. Level of difficulty

When the student starts with the course, the *level of difficulty* is calculated for each competence of the course. It depends on the initial level, specified by the course provider, and the marks that other students have obtained for this competence. The calculated levels do not change during the session.

The linguistic variable *level of difficulty* is described by the triple of membership function values $(\mu_{easy}, \mu_{norm}, \mu_{diff})$ and can take three linguistic values: *easy*, *normal*, and *difficult*.

The initial level of difficulty l for a competence c ranges from 1 to 10, where 1 means very easy and 10 very difficult competence. This value is fuzzified using the membership functions in Fig. 3 to determine the values of the triple.

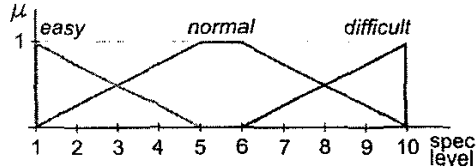


Figure 3. Membership functions for linguistic variable *level of difficulty*

They are calculated from the following equations:

$$\begin{aligned}\mu_{easy}(c) &= \max\left[\frac{5-l}{4}, 0\right] \\ \mu_{norm}(c) &= \min\left[\frac{l-1}{4}, \frac{10-l}{4}\right] \\ \mu_{diff}(c) &= \max\left[\frac{l-6}{4}, 0\right]\end{aligned}\quad (2)$$

If the initial level is not specified for any reason, the values (0,1,0) are used as default.

Then the values of membership functions are altered regarding the marks the (other) students obtained at the final-assessment test for this competence. If the marks are high, the

competence is considered to be easy. In the case of low marks, the competence is difficult. For each mark the following is applied:

- for mark 9 or 10 decrease the difficulty; apply equations (3),
- for mark 7 or 8 reinforce the medium difficulty; apply equations (4),
- for mark 6 or negative increase the difficulty; apply equations (5).

Decreasing the difficulty:

$$\mu_{easy}(c) = \mu_{easy}(c) + \mu_{norm}(c) \cdot r \quad (3)$$

$$\mu_{norm}(c) = \mu_{norm}(c) - \mu_{norm}(c) \cdot r + \mu_{diff}(c) \cdot r$$

$$\mu_{diff}(c) = \mu_{diff}(c) - \mu_{diff}(c) \cdot r$$

Reinforcing the medium difficulty:

$$\mu_{easy}(c) = \mu_{easy}(c) - \mu_{easy}(c) \cdot r \quad (4)$$

$$\mu_{norm}(c) = \mu_{norm}(c) + \mu_{easy}(c) \cdot r + \mu_{diff}(c) \cdot r$$

$$\mu_{diff}(c) = \mu_{diff}(c) - \mu_{diff}(c) \cdot r$$

Increasing the difficulty:

$$\mu_{easy}(c) = \mu_{easy}(c) - \mu_{easy}(c) \cdot r \quad (5)$$

$$\mu_{norm}(c) = \mu_{norm}(c) - \mu_{norm}(c) \cdot r + \mu_{easy}(c) \cdot r$$

$$\mu_{diff}(c) = \mu_{diff}(c) + \mu_{norm}(c) \cdot r$$

The constant r in the equations controls the rate of changes. The proposed value for r is

$$r = \min\left[\frac{1}{n}, \frac{1}{10}\right], \quad (6)$$

where n is the total number of all marks (positive and negative) received for the competence c .

To enforce changes to the values of membership functions we use similar approach as employed in ML-Modeler [3], where probability masses for seven values (range from *definitely* to *definitely not*) are assigned to each concept. These concepts are then upgraded or downgraded using fixed updating rules that shift the mass of the probability toward the more probable (upgrading) or less probable (downgrading) ends of the distribution.

5.3. Marks

When students take the final-assessment test for a competence, they get a mark for that test, which is a crisp value between 1 and 10 (value 0 signifies no mark). The linguistic variable *marks* of the final-assessment test for the competence can take the following values: *negative*, *positive*, and *no mark*. The variable is fully described by a triple $(\mu_{no}, \mu_{neg}, \mu_{pos})$. The obtained mark is

transformed into the linguistic values using the membership functions in Fig. 4.

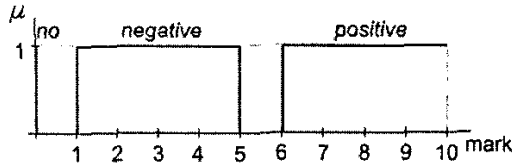


Figure 4. Membership functions for linguistic variable marks

For a given mark m , the values of membership functions can be calculated using equations (7).

$$\begin{aligned}\mu_{no}(m) &= \max[1 - m, 0] \\ \mu_{neg}(m) &= \max[\min[m, 1, 6 - m], 0] \\ \mu_{pos}(m) &= \min[\max[m - 5, 0], 1]\end{aligned}\quad (7)$$

5.4. Knowing the prerequisites

For each competence that has some prerequisites (in the graph of dependencies), the *level of knowing* these *prerequisites* is calculated and the level they are known (learned) is described using five linguistic terms: *not*, *little*, *enough*, *well*, and *very well*. The estimation of knowing the prerequisite competences is based on the marks the student obtained when taking final-assessment tests. If a competence does not have any prerequisites, it is treated as if all its prerequisites are *very well* known.

Let the prerequisite competences of the competence c be c_1, c_2, \dots, c_p , and their corresponding marks m_1, m_2, \dots, m_p , respectively. The competence c thus has p prerequisites. The minimum mark m_{min} and average mark m_{ave} of the prerequisites are calculated as follows:

$$m_{min} = \min_{i=1..p}[m_i] \quad (8)$$

$$m_{ave} = \frac{1}{p} \sum_{i=1}^p m_i \quad (9)$$

The level of knowing the prerequisites of a competence c is described by a quintuple $(\mu_{not}, \mu_{lit}, \mu_{en}, \mu_{well}, \mu_{very})$. If the competence has no prerequisites ($p = 0$), the quintuple has the value $(0, 0, 0, 0, 1)$. If one of the prerequisite competences has a negative mark or no mark at all ($m_{min} < 6$), the value is $(1, 0, 0, 0, 0)$. Otherwise, the values of membership functions are calculated from the

average mark (m_{ave}) using the following equations (for $m_{min} \geq 6$):

$$\begin{aligned}\mu_{not}(m_{ave}) &= 0 \\ \mu_{lit}(m_{ave}) &= \max[\min[1, 4 - 2m_{ave}], 0] \\ \mu_{en}(m_{ave}) &= \max[\min[2m_{ave} - 13, 1, 16 - 2m_{ave}], 0] \\ \mu_{well}(m_{ave}) &= \max[\min[2m_{ave} - 15, 1, 19 - 2m_{ave}], 0] \\ \mu_{very}(m_{ave}) &= \max[\min[2m_{ave} - 18, 1], 0]\end{aligned}\quad (10)$$

The corresponding membership functions are depicted in Fig. 5.

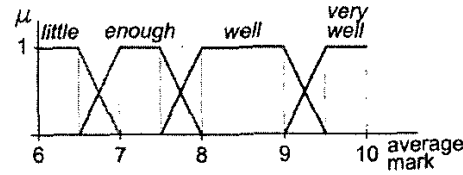


Figure 5. Membership functions for linguistic variable knowing the prerequisites

5.5. IF-THEN rules

We use rules in natural language for describing the relations between the *level of difficulty* of the competence, the *marks* obtained in the final-assessment test for the competence, and the estimated *knowing the prerequisites* of this competence. This kind of rules is easier comprehended and therefore easier to construct. We operate with the described linguistic variables to assign each competence its *level of recommendation* (with values *learned*, *more recommended*, *recommended*, *less recommended*, and *forbidden*). We created a set of 17 fuzzy rules. Each fuzzy rule is an IF-THEN rule as defined in (1); the following rules are some examples:

IF the student has a *positive* mark for the final-assessment test for the competence, THEN the competence is *learned*.

IF the student has a *negative* mark for the final-assessment test for the competence AND the competence is *easy*, THEN the competence is *more recommended*.

IF the student has *no mark* for the final-assessment test for the competence AND the student knows *well* all the prerequisites of the competence AND the competence is *easy*, THEN the competence is *more recommended*.

IF the student has *no mark* for the final-assessment test for the competence AND the student knows *little* all the prerequisites of the competence AND the competence is *difficult*, THEN the competence is *less recommended*.

When combining rules, we use product (PROD) for aggregation of the degrees of truth and sum (SUM) for their composition.

5.6. Defuzzification

In the end, the linguistic values of the linguistic variable *level of recommendation* are defuzzified into crisp values.

We use method Centre of Maximum (CoM), which computes the crisp value from the most typical values and respective degrees of membership. The most typical value of each linguistic term is the maximum of the respective membership function. For the variable *level of recommendation* the most typical values for the linguistic terms *learned*, *more recommended*, *recommended*, *less recommended*, and *forbidden* are 4, 3, 2, 1, and 0, respectively (see Fig. 6).

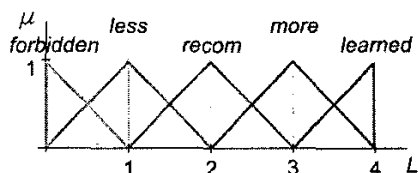


Figure 6. Membership functions for linguistic variable *level of recommendation*

The crisp value L for the competence c is computed as the best compromise for the given typical values and respective degrees of membership using weighted mean. The respective degrees are used as the respective weights:

$$L = \frac{\mu_{\text{forb}}(c) \cdot 0 + \mu_{\text{less}}(c) \cdot 1 + \mu_{\text{rec}}(c) \cdot 2 + \mu_{\text{more}}(c) \cdot 3 + \mu_{\text{learn}}(c) \cdot 4}{\mu_{\text{forb}}(c) + \mu_{\text{less}}(c) + \mu_{\text{rec}}(c) + \mu_{\text{more}}(c) + \mu_{\text{learn}}(c)} \quad (11)$$

The calculated value L can be used for annotating the nodes (competences) or for assigning weights to the arcs in the navigation graph. This way, the student receives more detailed information on suggested competences.

6. Conclusion

We have described a way of introducing the navigation support in the e-learning platform InterMediActor. When constructing the

navigation graph, we make a prediction on suitability of the competences, based mainly on the marks of the final-assessment tests. We proposed a fuzzy student model, which was developed and tested on simulated user data. The next step is to integrate it into the e-learning platform and validate it on real student data.

7. References

- [1] Brusilovsky P. Adaptive hypermedia: From intelligent tutoring systems to Web-based education. Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1839; Proceedings of 5th International Conference on Intelligent Tutoring Systems, ITS 2000; Jun 2000; Montreal, Canada. Berlin: Springer Verlag; 2000.
- [2] Gagne RM. Principles of Instructional Design. Third Edition. Holt, Rinehart and Winston; New York; 1988.
- [3] Güreş DW, desJardins M, Schlager M. Representing a Student's Learning States and Transitions. AAAI Spring Symposium on Representing Mental States and Mechanisms. Stanford, CA, USA. AAAI technical report, SRI International; 1995.
- [4] Harris JW, Stocker H. Handbook of Mathematics and Computational Science. Springer-Verlag; New York; 1998.
- [5] Klir GJ, Yuan B. Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall PTR; 1995.
- [6] Pedraza-Jimenez R, Valverde-Albacete FJ, Cid-Sueiro J, Molina-Bulla H, Navia-Vazquez A. Hierarchy-Based Methodology for Producing Educational Contents with Maximal Reutilization. In: Proceedings of the Learning'02 Conference; Oct 2002; Madrid, Spain; 2002.
- [7] PRIS Project: Fuzzy Expert System Model. Soft Computing Guidebook. <http://www.comp.nus.edu.sg/~pris/FuzzyLogic/ModelDetailed1.html> [29/2/04]
- [8] Valverde-Albacete FJ, Cid-Sueiro J, Diaz-Perez P, Pedraza-Jimenez R, Molina-Bulla H, Navia-Vazquez A. InterMediActor: an Environment for Instructional Content Design Based on Competences. In: Proceedings of the 2003 ACM Symposium on Applied Computing SAC'03; Mar 2003; Melbourne, USA; 2003.
- [9] Zadeh LA. Fuzzy Sets. Information and Control 1965; 8.