

Student Modelling Based on Fuzzy Inference Mechanisms

Alenka Kavčič*, Angel Navia-Vázquez**, and Rafael Pedraza-Jiménez**

*Faculty of Computer and Information Science, University of Ljubljana
Tržaška 25, Ljubljana, Slovenia
email: alenka.kavcic@fri.uni-lj.si

** Department of Signal Processing and Communications, University Carlos III of Madrid
Avda. de la Universidad 30, Leganés, Spain
email: navia@tsc.uc3m.es, rpedraza@tsc.uc3m.es

Abstract -- The paper presents a competence-based instructional design system and a way to provide a personalization of navigation in the course content. The navigation aid tool builds on the competence graph and the student model, which includes the elements of uncertainty in the assessment of students. An individualized navigation graph is constructed for each student, suggesting the competences the student is more prepared to study. We use fuzzy set theory for dealing with uncertainty. The marks of the assessment tests are transformed into linguistic terms and used for assigning values to linguistic variables. For each competence, the level of difficulty and the level of knowing its prerequisites are calculated based on the assessment marks. Using these linguistic variables and approximate reasoning (fuzzy IF-THEN rules), a crisp category is assigned to each competence regarding its level of recommendation.

Index Terms--Educational systems, fuzzy logic, personalized navigation, user modelling.

I. INTRODUCTION

CURRICULUM sequencing is one of the oldest technologies in intelligent tutoring systems, which was adopted in educational hypermedia as an adaptive navigation support technique called direct guidance. This technique helps each student to find an individualized optimal path through the learning material [2]. The personalization of the navigation is done through the student model, which collects vital information on each individual student.

Among different possible approaches to introducing navigation assistance in the system we searched for the solution, which could be integrated into the existing teleeducation platform InterMediActor. It relies on prediction model built upon the information from the student's previous interaction with the system. The navigation graph is constructed for each particular student and changes appropriately as the student advances through the course.

In the paper, the competence-based methodology for content production and reuse is presented first. We briefly

review the methodology proposed for a distance and computer-based education platform InterMediActor. The possibilities of introducing a student model into the system are discussed. We decided to use fuzzy user modelling, which is based on fuzzy inference. The proposed solution for the presented educational platform operates with linguistic variables and uses linguistic rules.

II. INTERMEDIATOR PLATFORM

InterMediActor [7], [10] introduces architecture for development, production, deployment, and use of educative content. Basically, it is an instructional design system, which provides an environment for instructional content design, production, and reuse, as well as student evaluation. The instructional model used in InterMediActor platform is based on competences, which are educational equivalents of grounded learning objectives, and employs decomposition of the learning objectives into a hierarchical structure of corresponding competences. The proposed method establishes a two-phase process to develop educational content:

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

In the first phase, the top-down analysis of learning objectives proceeds by partitioning and refining overarching, all-encompassing learning objectives into more detailed and concrete objectives, until the 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 (a tree-like hierarchy), and *depends-on* relationship between objectives that belong to the same more general objective (a finite graph).

After performing the top-down analysis of the learning objectives, each atomic objective is expanded into an atomic competence that consists of:

- the *advance organizer* that introduces the subject matter to be learned,
- the *content* to be learned (which can be a complex multimedia object),

- the *summary* that states what the learning outcome should be,
- some *self-assessment tests* and *final-assessment tests* based on the introduced content,
- the *prerequisites* of the competence, which have to be grasped before trying to understand this particular competence (part of which encode the *depends-on* relationship with competences of the same granularity).

When the set of atomic competences is specified, the bottom-up synthesis of competences is performed following the *part-of* relationship in the hierarchy graph. This way, more complex, higher-level aggregated competences can be synthesized. 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 an aggregated, second-order competence can then be used to synthesize new ones as needed by the analysis, until each learning objective in the analysis coalesces into a competence and the resulting content has the shape of a competence graph.

III. USER MODELLING IN INTERMEDIATOR

The main goal of user modelling in the InterMediaActor platform is to provide a particular student model and an ad hoc personalization of navigation in the course content. As the model of InterMediaActor is already fully developed, we have to construct such a user model that fits well in its context.

A. Possibilities of Introducing a Student Model

First, we have to study the possibilities of introducing a student model in the developed InterMediaActor platform. We proceed from the competences and the prerequisite relations among them.

A heterogeneous graph of dependences, which is constructed as a result of the analysis of learning objectives, can be regarded as two independent graphs: hierarchy graph and graph of dependencies. The arcs in the former indicate the *part-of* relationship, while the arcs in the latter the *depends-on* relationship. The nodes in both graphs are competences. The graph of dependencies contains also the information on prerequisite competences. The students have to learn all the prerequisite competences before they start learning the competence.

The most relevant data about a particular student are the marks obtained in self-assessment and final-assessment tests, and the sequence of visiting the competences. Therefore, the only logical way to introduce a student model is a model based on student's marks for competences.

B. Proposed Student Model for InterMediaActor Platform

The proposed student model is founded on the marks the student gets in assessment tests. It consists of a set of competences, each of which has attached all marks of final-assessment tests (both, passed and failed). We decided not to

use the self-assessment marks, because they are intended for student's self-evaluation only and are not collected. The path the student has used to navigate the course content (the sequence of visiting the competences) is also saved in the model. The student model is hence an extended overlay model over the set of competences.

Because the process of inferring from assessment marks involves some imprecision, vagueness and a great deal of uncertainty, we decided to use fuzzy set theory [11] as a mathematical theory for expressing this uncertainty.

C. Personalization of Navigation in the Course Content

From the data in the student model we construct an individualized navigation path through the graph of competences. The path can be extracted from the navigation graph, which is created for a particular student. The navigation graph is based on the graph of dependences; its nodes are competences and arcs show all possible ways of visiting other competences of the same level.

The navigation graph is displayed with the nodes in different colours: grey indicates already learned competences, red nodes signify forbidden competences (at the moment), and green nodes are recommended competences. There are three shades of green, indicating more, ordinary, or less recommended competence to follow.

IV. FUZZY LOGIC AND FUZZY INFERENCE

The theory of fuzzy logic is a part of a broader fuzzy set theory. It extends classical Boolean logic by continuous range of truth values from the closed interval $[0,1]$, which allows intermediate values to be defined between conventional binary.

A. Fuzzy Sets

Fuzzy set theory was formalized by Professor Lotfi Zadeh [11] in 1965. It extends the limiting bivalent sets in a way that allows smooth transition between sets, especially convenient for describing natural phenomena. In fuzzy sets, an element is not strictly a member or not a member of a set, but can also be only partially in the set, which means it is present in the set to some extent only. Hence, a membership function for a fuzzy set takes values in the unit interval $[0,1]$ rather than in the $\{0,1\}$ as in the classical sets.

Let X be the universe of discourse and its elements marked as x . A fuzzy set A with a membership function μ_A is defined as:

$$\begin{aligned} A &\subseteq X, \\ A &= \{\mu_A(x_i)/x_i\}, \quad x_i \in X, \\ \mu_A &: X \rightarrow [0,1]. \end{aligned} \tag{1}$$

Fuzzy set A is a subset of a given fundamental set X , whose elements are contained in A only to a certain degree. Membership function μ_A maps the fundamental set X onto

the closed interval $[0,1]$ and expresses, how much the element x from X is contained in A .

B. Fuzzy Inference

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 several rules (implications), a set of facts, and a conclusion [4]. The fuzzy production rules connect premises with conclusions, conditions with actions [6]. They have the form of IF-THEN rules, where the IF part of the rule is the premise (condition), and the THEN part of the rule is the conclusion (action).

Performing fuzzy reasoning involves the following processes [9]:

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

Fuzzification is the process of determining degrees of membership of crisp values in each fuzzy set. Crisp inputs are fuzzified into linguistic values (with defined membership functions) to be related to the linguistic variables. Linguistic variables take on linguistic values, which are words (linguistic terms) with associated degrees of membership in the set.

Fuzzy inference matches fuzzy facts against fuzzy conditions and assigns fuzzy output set. A fuzzy IF-THEN rule has the form

$$\text{IF } X_1 = A_1 \text{ and } X_2 = A_2 \text{ and } \dots \text{ and } X_n = A_n \text{ THEN } Y = B, \quad (2)$$

where X_i and Y are linguistic variables, and A_i and B are linguistic terms. In contrast to crisp rules, each rule is allowed to fire in a fuzzy system. Thus, the order in which the rules execute is not important.

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 term. Then, either the minimum (MIN) or the product (PROD) of the degrees of truth of the conditions is computed as the degree of truth of the IF part. This is the degree of support for the rule and is assigned for the degree of truth of the THEN part of the rule.

Composition is the process of computing the values of the rule's conclusion. To determinate the degrees of truth of each linguistic term of the output linguistic variable, 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 are usually used.

The last step is the defuzzification of the linguistic values of the output linguistic variables into crisp values. One of the most common techniques used is Centre of Maximum (CoM) method. The crisp value is computed as the best compromise for the most typical values for each linguistic term for an output linguistic variable, and respective degrees of

membership.

V. PROPOSED FUZZY USER MODEL

In the process of navigation graph construction we employ the fuzzy student model to assign categories to competences regarding their levels of recommendation.

For each competence we calculate a numerical level of recommendation, and a crisp category the competence belongs to (which determines the colour of the node in the navigation graph).

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

A. Level of Recommendation

To each competence (regarding a particular student) is assigned exactly one of the five crisp categories: *learned*, *more recommended*, *recommended*, *less recommended*, or *forbidden*. These categories correspond to 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 obtained in the final test for the competence (if any), and on how well the prerequisites of this competence are known/learned (if there are any).

B. Level of Difficulty of a Competence

When the student enters the course, the value of the linguistic variable *level of difficulty* is calculated for each competence of the course. It is a function of the marks that other students have achieved on this subject, described by three linguistic terms: *easy*, *normal*, and *difficult*.

The initial level is specified by the content provider (a professor). If not, the middle level of difficulty (*normal*) is used by default. The specified values of initial levels are first fuzzified to match the linguistic terms and then altered depending on the marks the (other) students obtained at the final-assessment test for this competence. If the marks of the students are high, the competence is considered easy. In the case of low marks, the competence is difficult. For each mark we do the following:

- if the mark is 9 or 10, decrease the difficulty;
- if the mark is 7 or 8, reinforce the medium difficulty;
- if the mark is 6 or less, increase the difficulty.

To enforce the changes to the values of membership functions we use similar approach as applied in ML-Modeler [3], shifting the values towards difficult/easy when applying upward/downward changes using fixed rules.

The levels of difficulty calculated this way do not change during the session.

C. Mark of the Final-Assessment Test

When the student takes the final-assessment test for a

competence, the mark for this competence is obtained. The linguistic variable *marks* corresponds to the obtained mark of the final-assessment test for the competence and can take values *negative*, *positive*, or *no mark*. The value of the variable *marks* changes, when the student takes the final test. Although the variable *marks* is a crisp variable, we use membership functions (with only two possible values: 0 or 1) for representation of its values.

D. 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 obtained by the student in final-assessment tests. The values of membership functions for linguistic terms are calculated from the average mark and the minimum mark of the prerequisites. If a competence does not have any prerequisites, it is treated as all its prerequisites are *very well* known.

E. Linguistic IF-THEN Rules

We use rules in natural language for describing relations between the *level of difficulty* of the competence (*easy*, *normal*, *difficult*), the *marks* obtained in the final-assessment test for the competence (*positive*, *negative*, *no mark*), and the estimated *knowing the prerequisites* of the competence (*not*, *little*, *enough*, *well*, *very well*). 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*, or *forbidden*. The set of rules like the following is created:

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*.

Each fuzzy rule is an IF-THEN rule as defined in (2). For combining rules, we use product (PROD) for aggregation of the degrees of truth and sum (SUM) for their composition.

F. Defuzzification of the Level of Recommendation

In the end, the linguistic values of the linguistic variable

level of recommendation are defuzzified into crisp values.

We use the Centre of Maximum (CoM) method, which first determinates the most typical values for each linguistic term, and then computes the crisp value as the best compromise for the given typical values and respective degrees of membership using weighted mean. The degrees of membership are used as weights, and the most typical value of each linguistic term is the maximum of the respective membership function.

G. Applying the Categories

The assigned crisp category can be used for annotating the nodes (competences) in the navigation graph. All learned competences are displayed in grey colour and there are no arcs leading to them. All forbidden competences are red and have no arcs as well. The recommended (more/less) competences are green (dark/light) and are connected with the current competence by an arc. The arcs in the navigation graph are always leading only from the current competence (the one the student is studying or has just finished studying) to all recommended (more/less) competences (green nodes). Red nodes are forbidden (temporarily) and thus no path leads to them. Grey nodes are already learned; the student can return to see them, but no path leads to them because they were already visited (studied).

We can also assign weights to the arcs in the navigation graph. The weights equal to the value of defuzzified level of recommendation, which is calculated for the competence the arc is leading to. This way, the student receives more detailed information on recommendation of competences. Although two competences are in the same category (same colour of nodes), they might have different weights, which enables easier selection of the more recommended one.

VI. CONCLUSION

We have described a way of suggesting the navigation path through the course material in the teleeducating platform InterMediActor. The predictions of the suitability of the next subject to study are made out of the student's marks obtained in the final-assessment tests for other already learned subjects.

The proposed fuzzy student model was developed and tested on simulated user data. It still has to be integrated into the InterMediActor platform and validated on real student data. Then the parameters in the model may be fine-tuned to assure best performance.

REFERENCES

- [1] P. Brusilovsky, "Methods and Techniques of Adaptive Hypermedia," *Adaptive Hypertext and Hypermedia*, Kluwer Academic Publishers, 1998.
- [2] P. Brusilovsky, "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, Montreal, Canada, June 19-23, 2000), Berlin: Springer Verlag, 2000.
- [3] D. W. Güreş, M. desJardins, and M. Schlager, "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] J. W. Harris and H. Stocker, *Handbook of Mathematics and Computational Science*, Springer-Verlag New York, 1998.
- [5] A. Kavčič, "Enhancing Educational Hypermedia: Personalization through Fuzzy Logic," *Proceedings of the 1st COST #276 Workshop on Information and Knowledge Management for Integrated Media Communication*, Leganés, Spain, November 2001.
- [6] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall PTR, 1995.
- [7] R. Pedraza-Jimenez, F. J. Valverde-Albacete, J. Cid-Sueiro, H. Molina-Bulla, and A. Navia-Vazquez, "Hierarchy-Based Methodology for Producing Educational Contents with Maximal Reutilization," *Proceedings of the Learning'02 Conference*, Madrid, Spain, October 2002.
- [8] R. Pedraza-Jimenez, A. R. Figueiras-Vidal, and A. Navia-Vazquez, "Guiding students through competence-based educational materials," *Proceedings of the Learning'02 Conference*, Madrid, Spain, October 2002.
- [9] PRIS Project, "Fuzzy Expert System Model", *Soft Computing Guidebook*,
<http://www.comp.nus.edu.sg/~pris/FuzzyLogic/ModelDetailed1.html>
- [10] F. J. Valverde-Albacete, J. Cid-Sueiro, P. Diaz-Perez, R. Pedraza-Jimenez, H. Molina-Bulla, and A. Navia-Vazquez, "InterMediActor: an Environment for Instructional Content Design Based on Competences," *ACM Symposium on Applied Computing SAC'03*, Melbourne, USA, March 2003.
- [11] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, Vol. 8, 1965.