A METHODOLOGY FOR DESIGNING AUTOMATIC EVALUATORS USING *GLMP* PARADIGM*

GLORIA SANCHEZ-TORRUBIA and CARMEN TORRES-BLANC

Applied Mathematics Department, Technical University of Madrid,
Boadilla del Monte, Madrid, Spain

Formative assessment improves human learning processes as it provides learners with information about what they need to work on. To automate formative assessment, we built a general model (*GLMP*) that allows the design of systems aimed at reproducing instructor's reasoning for learning assessment and generate a natural language assessment report. This paper presents a methodology for designing these automatic evaluators, highlighting the main points to be taken into consideration by the designer.

1. Introduction and Preliminaries

Assessment is a key part of human learning process. There are two types of assessment: summative and formative assessment.[1] Summative assessment determines the learner's final level of knowledge while formative assessment provides learners with information about what they need to work on during the learning process. This type of assessment is conducted by teachers using their own criteria to assess the achievement level on the learning objectives. This task is resource intensive, and is not easy to automate if the criteria to be implemented are complex. To automate formative assessment, we set ourselves the challenge of building a model that reproduces instructor's reasoning for learning assessment. The expert systems based on this model will provide learners with natural language reports on their attainment. Besides, the numerical grades output by the systems will not be a direct result of counting correct and incorrect responses; the responses will be aggregated emulating the instructor's criteria. A challenge for instructorless automatic assessment systems, implementing complex criteria, is given by their computational complexity. This problem has been addressed by several authors and different solutions have been proposed.[2,3]

*This work is partially supported by UPM.

In 3, we designed a granular linguistic model for the human learning assessment phenomenon, which will be described below, as a specific case of the granular linguistic model of a phenomenon[3,4] (*GLMP*) paradigm, based on Zadehs computational theory of perceptions.[5] Based on this model, we have designed, implemented and used several automatic evaluators[3,6] that perform the learning assessment for different topics. This paper presents a methodology for designing these systems.

A ***GLMP*** is a network of Computational Perceptions (*CPs*) related by Perception Mappings (*PMs*). In the network, each *CP* covers specific aspects of the modeled phenomenon with a set granularity level while the *PMs* aggregate the information contained in the subordinate *CPs* (the *CPs* below it) to generate the information to be contained in the output *CP*. Thus, the information is aggregated, by fuzzy inference, from bottom to top.

Definition 1.1. A **Computational Perception** (*CP*) is the computational model of a unit of information acquired by the designer about the phenomenon to be modeled. In general, *CPs* correspond to specific parts of the phenomenon at a particular granularity level. A *CP* is a pair $(A,\ W)$:

- $A = (a_1, ..., a_n)$ is a vector of linguistic expressions or predicates (natural language words or sentences) that represents the whole linguistic domain of the *CP*. Each a_i describes the value of this *CP* in a particular situation with a specific granularity level. These sentences can be e.g., $a_i =$ *In this simulation the edge selection error is high.*
- $W = (w_1, ..., w_n)$ is a vector of validity degrees. Each $w_i \in [0, 1]$ is assigned to the corresponding linguistic expression a_i. The concept of validity is a function of the truthfulness and relevancy of each sentence.

Each *CP* represents a facet of the phenomenon under examination. There are two types of *CPs*: 1-*CPs* interpret answers in terms of the elementary concept considered by that *CP* and represents the proficiency attained on that concept. 2-*CPs* represent compound concepts and are explained by a set of subordinate *CPs*. Let us denote $(A, W) = \{(a_1, w_1), ..., (a_n, w_n)\}$ stressing that w_i is the validity degree of a_i.

Definition 1.2. The inference process to generate the information to be stored in the *CPs* is carried out by **Perception Mappings** (*PMs*). A *PM* is a tuple (U, y, g, T):

- U is a set of input *CPs*, $U = \{u_1, u_2, ..., u_n\}$, where $u_i = (A_{u_i}, W_{u_i}) = \left\{(a_1^{u_i}, w_1^{u_i}), (a_2^{u_i}, w_2^{u_i}), ..., \left(a_{n_{u_i}}^{u_i}, w_{n_{u_i}}^{u_i}\right)\right\}$. In the special case of a first-order perception mapping (*1-PM*), U is a variable defined in the input data domain ($U = z$).

- y is the output *CP*, $y = (A_y, W_y) = \left\{ (a_1^y, w_1^y), (a_2^y, w_2^y), ..., \left(a_{n_y}^y, w_{n_y}^y\right)\right\}$.
- g is a fuzzy information aggregator with output $W_y = \left(w_1^y, w_2^y, ..., w_{n_y}^y\right) = g(W_{u_1}, W_{u_2}, ..., W_{u_n})$, where W_y is the vector of validity degrees assigned to each linguistic expression in y and W_{u_i} are the validity degrees of the input perceptions. In many cases, but not always, g is implemented using a set of fuzzy rules, in this case g is a vector of fuzzy aggregation functions in $[0, 1]$ (see Proposition 2.1). In the special case of 1-*PMs*, g fuzzifies the data (thus not being an aggregation function in $[0, 1]$), and g is built using a set of membership functions: $W_y = g(z) = \left(\mu_{a_1^y}(z), \mu_{a_2^y}(z), ..., \mu_{a_{n_y}^y}(z)\right) = \left(w_1^y, w_2^y, ..., w_{n_y}^y\right)$, where w_j^y is the validity degree assigned to a_j^y and z is the input data.
- T is a text generation algorithm resulting in the generation of sentences in A_y. In our case, T is a linguistic template, e.g. *The edge selection error is {very low | low | medium | high | very high}*.

2. Designing methodology

The model described in the previous section is able to represent instructor assessment reasoning. Using specific fuzzy aggregators and different linguistic expressions, the *GLMP* paradigm provides enough resources to design automatic assessment tools that emulate the assessment process enacted by an instructor. To get this, the general model will have to be tailored to each case by defining its constituent elements. The following sections describe the characteristics to be met by these elements in order to facilitate coherence, meet the design goals and improve local system behavior. Additionally, other details that must be taken into account when defining the structure of the system are also exposed.

2.1. *Linguistic labels*

The set of linguistic labels describing a perception is an important manifestation of the designer's criteria. Also, in most cases, these labels are part of the antecedent in the inference engine rules. Therefore, in order to properly represent the linguistic domain of this *CP* and prevent interpretability and coherence problems, we recommend selecting, as linguistic labels, an orthogonal set of triangular or trapezoidal functions. These functions values can be easily calculated, thus their computational performance is quite efficient. Moreover, both the number of labels used and the way in which the parameters of these labels are chosen affect the design, e.g. selecting a trapezoidal-shaped label similar to a crisp set membership function causes the system to locally generate echelon-shaped outputs. On the other hand, selecting low-slope triangular labels locally increases the system fuzziness. Also, when different amplitude triangular labels are used, the area where the triangles vertices are closer is processed by the system more thoroughly.

2.2. *Inference engine: features and performance of the sets of fuzzy rules*

In most 2-*PMs*, the fuzzy information aggregator g, given in Definition 1.2, is implemented using a set of fuzzy *if-then* rules. In this section we describe the implementation of those rules, by using t-norms and t-conorms, and obtain in Proposition 2.1 that, with this implementation, g is a vector of aggregation functions in $[0,1]$. Let U be the domain of discourse and $x \in U$,

- If the rule takes the form $R \equiv$ *If* A *then* B, and the antecedent A is true to some degree of membership $\mu_A(x)$, then the validity degree of the consequent B is $\mu_A(x)$.
- If the rule takes the form $R \equiv$ *If* x_1 *is* A_1 *and* $\ldots$ *and* x_n *is* A_n *then* B i.e. $x = (x_1, \ldots, x_n)$ and the antecedent is a set of requirements on $A = (A_1, \ldots, A_n)$, then the validity degree of the antecedent is $\mu_A(x) = T\left(\mu_{A_1}(x_1), \ldots, \mu_{A_n}(x_n)\right)$, where $\mu_{A_i}(x_i)$ is the validity degree of x_i *is* A_i and T is a t-norm.
- If $\{R_1, \ldots, R_k\}$ is the subset of the rule set with the same consequent B, $R_1 \equiv$ *If* A_1 *then* B,$\ldots$,$R_k \equiv$ *If* A_k *then* B, such that $\mu_{A_j}(x) = w_j$, then the validity degree of B is $w = S(w_1, \ldots, w_k)$, where S is a t-conorm.

Proposition 2.1. *Let* (U, y, g, T) *be a 2-PM whose information aggregator* g *has been defined as an inference engine implemented by a set of fuzzy if-then rules, evaluated as described above. Then* $g = \left(g_1, \ldots, g_{n_y}\right)$*, where* $g_k : [0,1]^\beta \to [0,1]$*, is a fuzzy aggregation function in* $[0,1]$ *and* β *is the total number of labels in the set of subordinate perceptions* U*.*

Let us note that, in several 2-*PMs*, e.g. those using quantifiers,[6] the functions g_k in $g = \left(g_1, ..., g_{n_y}\right)$ are not fuzzy aggregation functions in $[0,1]$.

2.3. *Comments on the systems design*

Systems designed using *GLMP* paradigm process errors without significant information loss, as each 2-*CP* is explained based on the validity degree vectors and each vector covers the whole linguistic domain of the corresponding subordinate *CP*. Additionally, computational complexity is very low as the information processing is performed by sums and products on the validity degree vectors instead of using fuzzy sets membership functions.

When designing an automatic evaluator, the first step is choosing the target to be assessed and the system from which the input data will be extracted. Then, the elementary concepts to be represented by 1-*CPs* should be selected. Figure 1 shows two *GLMP* designs for assessing Prim's algorithm[7] simulations. Both of them have been built using four input data: E_1 *data update error*, E_3 *flow control error*, E_4 *edge selection error* and T spent *time*. These data are processed by 1-*PMs* to generate the 1-*CPs* representing the elemental facets of the algorithm. For example, 1-CP_4 describes the level of knowledge achieved by a student in the edge selection in Prim's algorithm. It can be viewed as the *edge selection error* linguistic variable and is described by five qualifiers or linguistic labels (*very low, low, medium, high, very high*). Once the 1-*CPs* have been defined, they should be grouped together to establish 2-*CPs*. Each group should be defined joining related concepts, so that the output 2-*CP* represents a learning objective and the report is coherent from a pedagogical point of view. Additionally, when defining the *GLMP* structure, the designer should take into account the following recommendations.

- Each perception should not exceed three subordinate *CPs*.
- The number of labels in a *CP* should be greater or equal than the number of labels of each subordinate *CP*.
- The lowest levels perceptions should not have many linguistic labels.
- The top order perception (the highest level *CP*) should have a large number of labels.

A design problem is how to generate the maximum numerical grade (10), as this is a crisp concept difficult to obtain using fuzzy inference techniques. To solve this problem we have attempted two strategies and compared their results. The first strategy consists in duplicating the *CPs* that represent the most important concepts to be assessed (see Figure 1(a)). The second strategy (see Figure 1(b)) consists in adding a *perfect* label in most *CPs* to discriminate the *perfect* knowledge. This linguistic label will be nearly crisp and its centroid will be located at 1 or 0 position. Both strategies increase the number of rules in the inference engine.

The first strategy generates more perceptions, and therefore more rules, than the second strategy, but many of these rules are never fired and could

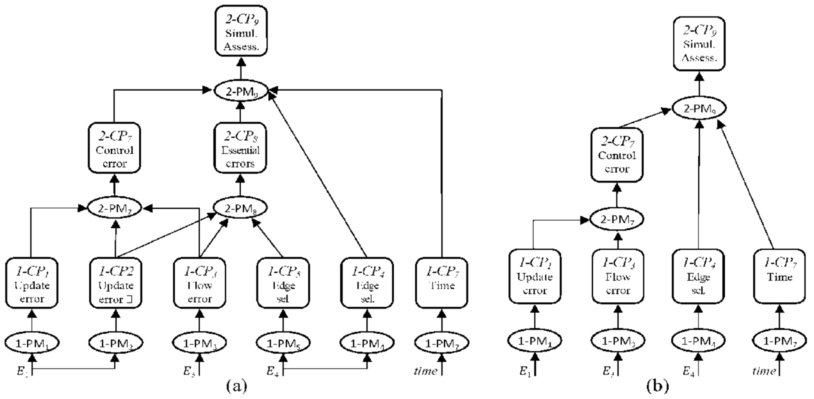

Fig. 1. (a) *GLMP* for assessing Prim's algorithm simulation using duplicating strategy, (b) *GLMP* for assessing Prim's algorithm simulation using *perfect* label strategy.

be deleted. Moreover, the second strategy has fewer rules but it is necessary to implement all of them, therefore the system is more difficult to adjust. However, the second strategy will better simulate the criterion of a teacher who wants to reward students for executing *perfectly* part of the exercise.

3. Conclusions

In this paper we presented a methodology for designing automatic evaluators based on the granular linguistic model for the human learning assessment phenomenon. This methodology describes the requirements to be met by the constituent elements of the systems in order to facilitate coherence, meet the design goals and enhance local system behavior.

References

1. D. R. Sadler, *Instructional Science* **18** (1989).
2. E. A. Soares, L. S. Machado and R. M. Moraes, *Performance Evaluation of Online Training Assessment on Embedded Systems*, in *Decision Making and Soft Computing*, (World Scientific, 2014), ch. 30, pp. 167–173.
3. M. G. Sanchez-Torrubia, C. Torres-Blanc and G. Trivino, *Expert Syst. Appl.* **39**, 12177 (2012).
4. G. Trivino and M. Sugeno, *Int. J. Approx. Reason.* **54**, 22 (2013).
5. L. Zadeh, *IEEE Trans. Circuits Syst.* **45**, 105 (1999).
6. M. G. Sanchez-Torrubia, *Especificaciones* eMathTeacher*, creación del Modelo Granular Lingüístico de la evaluación del aprendizaje humano...*, PhD thesis, UPM, (Spain, 2016), pp. xxvii + 177.
7. R. C. Prim, *Bell System Technical Journal* **36**, 1389 (1957).