
MACHINE LEARNING BASED METHODS USED FOR IMPROVING SCHOLAR PERFORMANCE

A PREPRINT

Radu BONCEA
ICI Bucuresti
radu@rotld.ro

Ionut PETRE
ICI Bucuresti
ionut.petre@rotld.ro

Victor VEVERA
ICI Bucuresti
victor.vevera@ici.ro

Alexandru GHEORGHITA
ICI Bucuresti
alex.gheorghita@rotld.ro

February 28, 2019

ABSTRACT

In their 1968 study *"The Teaching-Learning Paradox: A Comparative Analysis of College Teaching Methods"*, Robert Dubin and Thomas Taveggia found no evidence to indicate any basis for preferring one teaching method over another as measured by the performance of students on course examinations. The conclusion is based on systematic reanalysis of the data of almost 100 comparative studies of different college teaching methods. The teaching-learning process is virtually a black box in which the teaching methods do not influence scholar performance. So rather than focusing on teaching methods, we propose a method for improving scholar performance by a continuous and intelligent monitoring and assessing process of daily knowledge gains. Recent developments in machine learning and data analysis, allow us the use of techniques for unveiling the strengths and weaknesses in the learning process. Our proposed solution is "micro-assessing" each student and after each course, by sending a 5 minutes-long mini test on their smartphones, collects the result and send it for analysis to a dedicated platform. When the system has sufficient data, it can personalize the tests for each student with focus on areas the student is lacking. It can make recommendations, to teachers, students, school and competent authorities at local or national level. The solution is not a replacement for classical examinations, but it augments the learning experience through interactive and personalized quizzes. The teacher also has a better view over student's knowledge, thus he can do better assessments overall. Student abnormal *deviations* could be detected much faster, while competent authorities could assess the impact of their decisions in near-realtime.

Keywords scholar performance · assessment model · machine learning · knowledge gain · RLO · IAMA

1 Introduction

A study conducted in 1968 concluded that the teaching method has no impact on final exam performance [1], a principle known as *Teaching-Learning Paradox*. No matter the quality of teacher or the program, regardless of teaching methods and despite the circumstances before them, the students learned and they demonstrated their learning consistently across time. In his book "Why Don't Students Like School", Daniel Willingham provides a useful model for how the mind retains factual information (see Figure 1) and points out that factual information is not sufficient for learning, but is often the basis for real learning, since in order to apply concepts and make creative evaluations, we must have a database of knowledge from which to pull and make connections [2]. Thus, to achieve long-term memory, two things need to happen:

1) said data must be subject to an intense level of attention while it is readily available in working memory, and...

2) content previously stored in long-term memory must be pulled up and matched with the new data in order for it to permanently stick [3]....

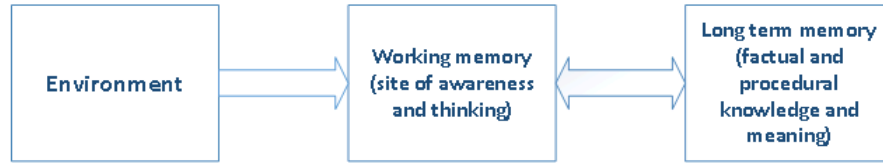


Figure 1: Daniel Willingham’s model of mind.

In this article we are proposing new methods that are addressing the above two challenges and improve scholar performance through means of demonstrating knowledge and understanding, by using an **Iterative** (daily base) and **Adaptive Monitoring and Assessing (IAMA)** process that relies on state-of-art technologies which uncover the power of *Data Intelligence* [4, 5, 6]. One such method implies a process of *micro-assessing* students right after each course using profiled Reusable Learning Objects (RLO) for assessment [7, 8, 9]. The assessment result is input data for generating and consolidating a student profile with focus on strengths and weaknesses. Data collected can also be used to monitor student performance and to alert and react to deviations. *IAMA* could also be seen as a game of words such as the *subreddit* question-and-answer *iAMA - interactive Ask Me Anything*.

It should be noted that the whole process is automated. The teacher allocates 5 minutes for each hour of course to assess students, by sending to their mobile electronic tablet, mini tests with 10 questions to be answered. Students receive the test and upon completion or expiring the time allocated, the applications installed on their mobile interfaces send the test results to a central platform (cloud based, data intensive), where they are stored. The system will process the results (compute scores, make inferences) and create and consolidate a knowledge-centered profile for each student. The profile is then used to recognize areas the student is lacking in knowledge and to provide with the input for future assessment adjustments and fine tuning.

2 IAMA Building Blocks

2.1 Assessment Reusable Learning Object

RLOs are conceptualised as accessible, reusable, interoperable, and adaptable learning resources to facilitate developmental cost savings. Therefore, RLOs need to be designed as independent learning units that are free from context as well as links with external resources [10]. Characteristics of RLOs:

- *digital asset* - stored, retrieved, modified by electronic means, available online 24/7;
- *self-contained* - loosely coupled, each learning object can be taken independently;
- *reusable* – a single RLO may be used in multiple contexts for multiple purposes;
- *searchable* - easy to find learning material; each RLO is tagged with metadata (see LOM subsection 2.2);
- *atomicity* - RLOs are 5-15 minutes long, distinct, small *units* of knowledge information;
- *flexible* – easy to update and change;
- *standardized* – adopt the same organizational structure from an enterprise architecture perspective;
- *aggregability* – learning objects can be grouped into a larger collection of content, including traditional course structures;
- *interoperability* – blend into Learning Management Systems (e.g. WebCT Vista, Moodle);
- *digital adaptability* - RLOs are suited to address a new type of learner – “*Net-generation learner*” adapted to multi-tasking and digital technologies;
- *student-focus* - enhance student-centered learning.

An assessment RLO (aRLO) is a RLO that is in essence a question associated with a set of possible answers and a set of correct answers and has additional properties:

- *mutability* - given a aRLO and a specific algorithm, we can obtain a set of distinctive aRLOs. A very simple algorithm would be to randomly sort the answers. A more complex algorithm is to make an inference on the question as seen in Figure 2.
- *affiliation* - aRLOs are associated with one ore more knowledge domains (parts of the curriculum) and memory domains (e.g. recall, identify, recognize, recount, relate,etc).

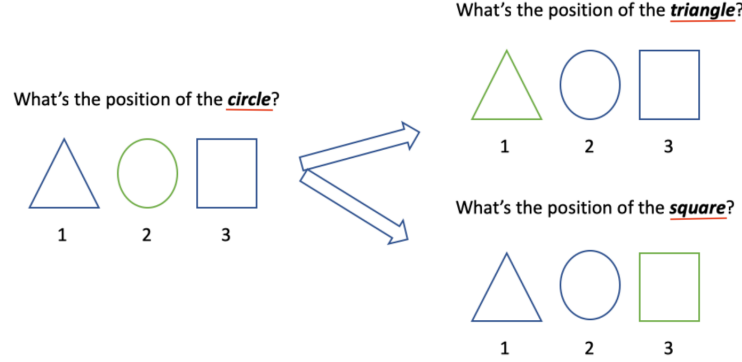


Figure 2: Inference based mutability of RLOs

2.2 Learning Object Metadata

Learning Object Metadata (LOM) [11] is a data model, used to describe a *learning object* and similar digital resources used to support learning. It is based on 1484.12.1 IEEE multi-part standard and has been developed with the purpose to facilitate search, evaluation, acquisition, and use of learning objects, for instance by learners or instructors or automated software processes. LOM also facilitates the sharing and exchange of learning objects, by enabling the development of catalogs and inventories while taking into account the diversity of cultural and lingual contexts in which the learning objects and their metadata are reused. It is a natural choice to describe RLOs.

The LOM conceptual data schema has a hierarchical tree structure composed of the following nine categories [12]:

1. *General*: information that describes the learning object as a whole.
2. *Lifecycle*: history and current status of the learning object and those who have contributed to its creation.
3. *Meta-metadata*: information about the metadata describing the learning object, as opposed to the learning object itself.
4. *Technical*: technical requirements and characteristics of the learning object.
5. *Educational*: educational and pedagogic characteristics of the learning object..
6. *Rights*: intellectual property rights and conditions of use of the learning object.
7. *Relation*: relationship between the learning object and other related objects.
8. *Annotation*: comments on the educational use of the learning objects, including when and by whom the comments were created.
9. *Classification*: classification schemes used to describe different characteristics of the learning object.

As seen in Figure 3, the LOM schema is relatively extensive and, in most cases, application profiles based on the standard generally restrict the elements used, designate certain elements as mandatory or optional, specify vocabulary usage and interpretation, and add organization or community specific classification schemes. And in some cases, the LOM schema must be extended with new properties. In our case, the aRLO should extend the *general* category, so it includes the two possible/correct answer sets.

2.3 Semantic Repository

A semantic repository is a database management systems, capable of handling structured data, taking into consideration their semantics and is perfectly suited to storing RLOs [13]. We can use Resource Description Framework (RDF) [14] to express RLO metadata following the IEEE LOM standard. A perfect candidate for storing and handling

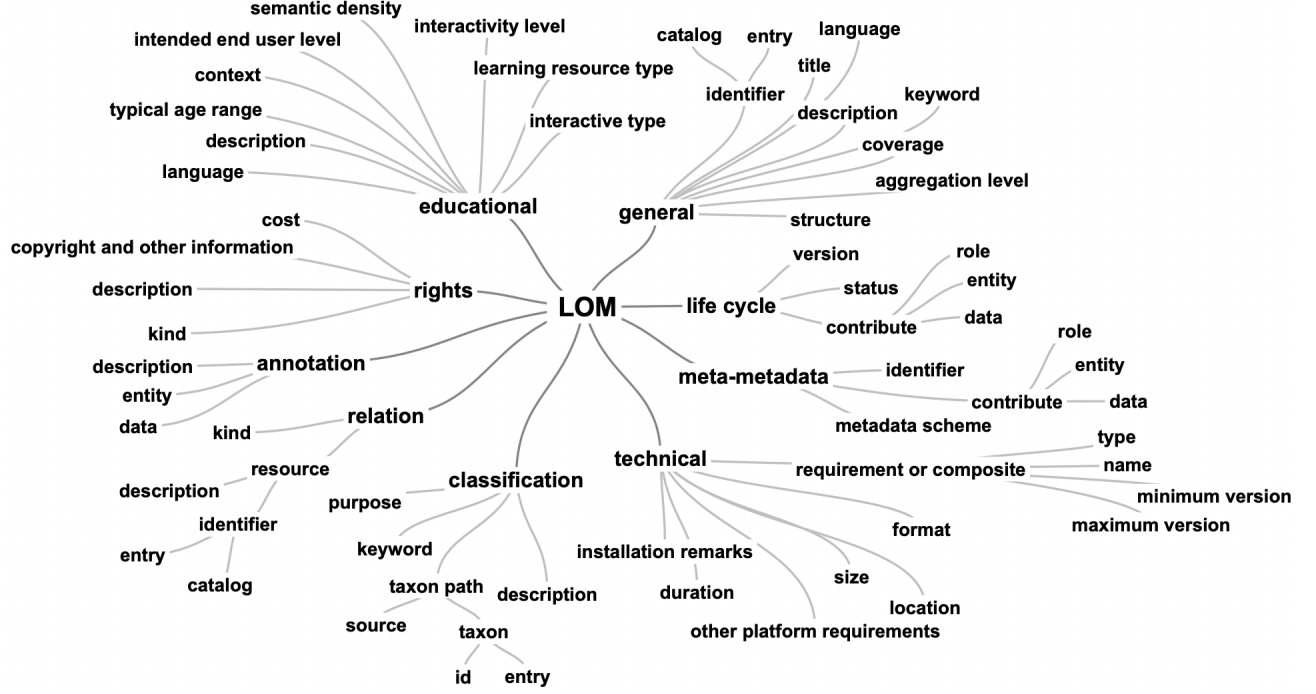


Figure 3: LOM conceptual data schema

RDF structures is Apache Jena [15], a free and open source Java framework for building semantic web and linked data applications that offers:

- RDF API to create and read RDF graphs and serialize the triples using popular formats such as RDF/XML or Turtle;
- graph-oriented query language ARQ, which is a SPARQL 1.1 compliant engine. ARQ supports remote federated queries and free text search via Lucene;
- a fast persistent triple store that stores directly to disk (TDB);
- ontology API working with models, RDFS and the Web Ontology Language (OWL) to add extra semantics to RDF data;
- inference API wrapping functionality around built-in OWL and RDFS reasoners.

2.4 Machine Learning Models and Frameworks

Monitoring student performance over time based on small iterative tests implies that we should *feed* the test results into an algorithm that outputs a score or a list of scores for a domain or a set of knowledge domains. There are several machine learning models that are well suited for doing classification and regression [16]:

1. *Naive Bayes Classifier* is a probabilistic classifier based on Bayes' Theorem with an assumption of independence among predictors and particularly useful for very large data sets. It is simple to implement and is known to outperform even highly sophisticated classification methods.
2. *Logistic Regression* is the appropriate regression predictive analysis to conduct when the dependent variable (the outcome) is dichotomous (binary).
3. *Decision Trees* builds classification or regression models in the form of a tree structure, by breaking down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.
4. *Random Forests* are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees. This model is very accurate, runs efficiently on large data sets and has a very useful feature: it gives estimates of what variables are important in the classification.

5. *Neural Network* consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer. Generally the networks are defined to be feed-forward: a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Weightings are applied to the signals passing from one unit to another, and it is these weightings which are tuned in the training phase to adapt a neural network to the particular problem at hand.
6. *Nearest Neighbor* is a supervised, non-parametric, classification and regression algorithm, where the input consists of the k closest training examples in the feature space.

These algorithms can be implemented using low level frameworks, such as SciPy [17] and Scikit Learn [18], highly used, simple, yet efficient Python libraries built on *NumPy*, *pandas*, and *matplotlib*. For implementing neural networks, specifically deep learning models, we should use higher level frameworks such as TensorFlow [19], CNTK [20], PyTorch [21] or Keras [22].

3 The Solution

Let $\mathcal{S} \in \mathcal{R}^{m \times n}$ annotate the associative matrix of computed scores for all m students and all n aRLOs, where \mathcal{R} is the closed interval $[0, 1] = \{x \in \mathbb{R}; 0 \leq x \leq 1\}$. As mentioned before an aRLO is an assessment reusable object that has attached to it a set of possible answers and a set of correct answers and a difficulty level $\delta \in (0, 1]$ where $\delta = 1$ means the most difficult level. Each element σ_i in the matrix \mathcal{S} is calculated as $\sigma_i = \gamma_i \times \delta_i$ where γ_i is the grade for the corresponding aRLO (students get maximum grade 1 if they get all correct answers right) and δ_i is the difficulty level of the aRLO [23].

Let $\mathcal{D} \in \{0, 1\}^{n \times p}$ annotate the matrix of mappings between knowledge domains and aRLOs and $\mathcal{M} \in \{0, 1\}^{n \times q}$ be the matrix of mappings between memory domains and aRLOs, where n is the number of aRLOs, p is the number of knowledge domains and q is the number of memory domains.

Then, we calculate the scores for associated knowledge domains $\mathcal{S}_{\mathcal{D}}$ with the formula $(\mathcal{S}_{\mathcal{D}})_{ij} = \frac{\sum_{k=1}^n \mathcal{S}_{i,k} \times \mathcal{D}_{k,j}}{N} = \mathcal{S} \otimes \mathcal{D}$, where N is the number of domain associations of each aRLO (the count of 1s). Similarly, we calculate the scores for associated memory domains as \mathcal{M} : $\mathcal{S}_{\mathcal{M}} = \mathcal{S} \otimes \mathcal{M}$.

For exemplification, let's consider the following matrixes: the grade scores G_s with m questions and n students, the knowledge domains associated matrix \mathcal{D} and the memory domains matrix \mathcal{M} ,

$$G_s = \begin{matrix} & \begin{matrix} S_1 & S_2 & S_3 & \dots & S_n \end{matrix} \\ \begin{matrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \dots \\ \gamma_m \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 1 & \dots & 1 \end{pmatrix} \end{matrix}$$

$$\mathcal{D} = \begin{matrix} & \begin{matrix} \gamma_1 & \gamma_2 & \gamma_3 & \dots & \gamma_m \end{matrix} \\ \begin{matrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \dots \\ \delta_p \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 1 & \dots & 0 \end{pmatrix} \end{matrix}$$

$$\mathcal{M} = \begin{matrix} & \begin{matrix} \gamma_1 & \gamma_2 & \gamma_3 & \dots & \gamma_m \end{matrix} \\ \begin{matrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \dots \\ \mu_q \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 1 & \dots & 0 \end{pmatrix} \end{matrix}$$

and the difficulty levels $Diff$

$$Diff = \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 & \dots & \gamma_m \\ 0.8 & 0.6 & 0.6 & \dots & 1 \end{pmatrix}$$

we calculate the score matrix \mathcal{S} , the knowledge domain scores $\mathcal{S}_{\mathcal{D}}$ and the memory domain scores $\mathcal{S}_{\mathcal{M}}$

$$\mathcal{S} = \begin{bmatrix} 1 \times 0.8 = 0.8 & 1 \times 0.8 = 0.8 & 0 \times 0.8 = 0 & \dots & 1 \times 0.8 = 0.8 \\ 1 \times 0.6 = 0.6 & 0 \times 0.6 = 0 & 1 \times 0.6 = 0.6 & \dots & 1 \times 0.6 = 0.6 \\ 0 \times 0.6 = 0 & 1 \times 0.6 = 0.6 & 1 \times 0.6 = 0.6 & \dots & 1 \times 0.6 = 0.6 \\ \dots & \dots & \dots & \dots & \dots \\ 1 \times 1 = 1 & 1 \times 1 = 1 & 1 \times 1 = 1 & \dots & 1 \times 1 = 1 \end{bmatrix} = \begin{bmatrix} 0.8 & 0.8 & 0 & \dots & 0.8 \\ 0.6 & 0 & 0.6 & \dots & 0.6 \\ 0 & 0.6 & 0.6 & \dots & 0.6 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

$$\mathcal{S}_{\mathcal{D}} = \mathcal{S} \otimes \mathcal{D} = \begin{bmatrix} 0.8 & 0.8 & 0 & \dots & 0.8 \\ 0.6 & 0 & 0.6 & \dots & 0.6 \\ 0 & 0.6 & 0.6 & \dots & 0.6 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 1 & \dots & 0 \end{bmatrix} = \begin{bmatrix} \frac{0.8+0.8}{2} & \frac{0.8}{1} & \frac{0.8}{2} & \dots & 0 \\ \frac{0.6}{2} & 0 & \frac{0.6}{2} & \dots & \frac{0.6}{1} \\ 0.6 & 0.6 & 0.6 & \dots & 0.6 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

$$= \begin{matrix} & \delta_1 & \delta_2 & \delta_3 & \dots & \delta_p \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ \dots \\ S_n \end{matrix} & \begin{pmatrix} 0.8 & 0.8 & 0.8 & \dots & 0 \\ 0.6 & 0 & 0.6 & \dots & 0.6 \\ 0.6 & 0.6 & 0.6 & \dots & 0.6 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix} \end{matrix}$$

$$\mathcal{S}_{\mathcal{M}} = \mathcal{S} \otimes \mathcal{M} = \begin{bmatrix} 0.8 & 0.8 & 0 & \dots & 0.8 \\ 0.6 & 0 & 0.6 & \dots & 0.6 \\ 0 & 0.6 & 0.6 & \dots & 0.6 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} =$$

$$= \begin{matrix} & \mu_1 & \mu_2 & \mu_3 & \dots & \mu_q \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ \dots \\ S_n \end{matrix} & \begin{pmatrix} 0 & 0.8 & 0.8 & \dots & 0.8 \\ 0.6 & 0 & 0.6 & \dots & 0 \\ 0.6 & 0.6 & 0.6 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix} \end{matrix}$$

In our previous example we can conclude that the n^{th} student has perfect score in each domain of knowledge and each domain of memory. The knowledge domain that is best demonstrated is δ_3 , while δ_p is the worst.

We then store the tensor $\mathcal{T} = (\mathcal{S}, \mathcal{S}_{\mathcal{D}}, \mathcal{S}_{\mathcal{M}})$ in a non-relational database like MongoDB or Cassandra and, after a certain number of iterations or mini tests, we can infer new relations using the machine learning models described before. For instance we could classify the students on knowledge levels (classification problem) or we can compute a score in the interval $[0,1]$ (multivariate regression problem). The result could then be used to provide with profiled tests to student in domains they are less knowledgeable.

Figure 4 shows a high level generic architecture of a IAMA solution, with three important application components:

1. *aRLO Management Component* is responsible with the development of aRLOs using LOM and RDF as semantic assets. aRLOs are stored and retrieved from a semantic repository, such as Apache Jena. This component also handles the aRLO mutability, that is, given aRLO and inference rules, the component will generate a set of related aRLOs.
2. *Assessment Component* main functionality is to generate profiled tests for students and handle the test results by storing them into a datastore.
3. *Data Analysis Component* uses machine learning models to generate the student profiles and have new insights.

The three components are well integrated in order to provide with business services that can be used by a large variety of actors (e.g. teachers, students, parent, local authorities, government), including the integration into larger application services ecosystem.

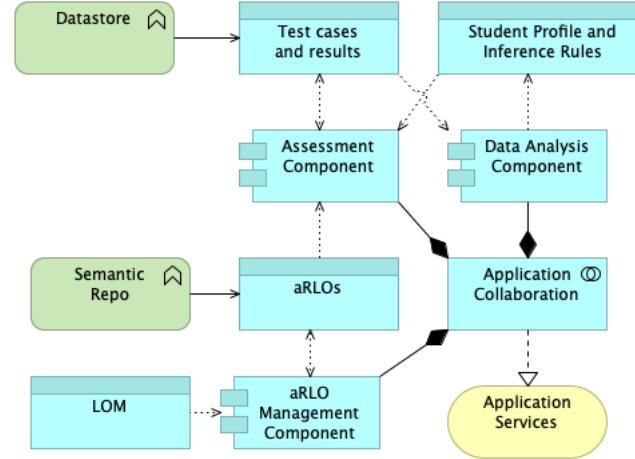


Figure 4: IAMA Reference Architecture

4 Conclusions

The solution addresses key challenges in teaching-learning process, by using data intelligence and machine learning in order to monitor the student performance over time. The IAMA model is based on an iterative and adaptive evaluation process that's using semantic and highly extensible reusable learning objects implemented on LOM and serialized and stored as RDF structures. The RLOs are associated with one or more knowledge domains (identifiable components of curriculum), one or more memory domains or long-term memory capabilities (e.g. recall, recount, recognize, relate, identify) and a difficulty level. The test results are stored for future retrieval and processed using machine learning algorithms suited for solving classification and regression problems. The main objective of this process is to obtain a student's profile which is continuously updated and improved. The student profile is then used to tune the assessment with focus on domains the student is less knowledgeable.

Except for RLO creation process, in IAMA, all other processes should be fully automated. The system should provide with distinct tests to all students in the class, making sure there are no students completing the same test. Assessments should contain a certain number of RLOs that are associated with previous courses so that all memory domains are sufficiently addressed. The same requirement is for knowledge domains coverage. But, the RLO creation could also be automated in some areas, though human supervision is a requirement that can't be overlooked. New RLOs can be procedurally generated using inference rules or more simple algorithms such as permutations of possible answers. Procedurally generated RLOs inherit most of the properties from the parent RLO, including affiliation (domain associative matrixes).

Besides the score, domain and memory matrixes, we can always add new features. For instance, we could additionally add the response time for each RLO or the sex, age and student's gender, or provide with data regarding the classroom, such as the number of students in class. And we can always verify which are the most important features, which features have greater impact on the classification or regression final result, by using *Random Forests*. This process can be further extended by including features regarding the school, the city, the country. Thus, the results can be aggregated into critical information for better decisions by teachers, students, parents and competent authorities. It is the feature extensibility that requires our solution to store the assessment results and associated data into a schema-free and non-relational database.

Bibliografie

- [1] Robert Dubin and Thomas C. Taveggia. *The Teaching-Learning Paradox: A Comparative Analysis of College Teaching Methods*. Tech. rep. Oregon Univ., Eugene. Center for Advanced Study of Educational Administration, 1968.
- [2] Daniel T. Willingham. *Why Don't Students Like School?: A Cognitive Scientist Answers Questions About How the Mind Works and What It Means for the Classroom*. Jossey-Bass, 2009. ISBN: 978-0-470-59196-3.

- [3] Andrew Neuendorf. *The Teaching-Learning Paradox Revisited*. 2014. URL: <https://andrewneuendorf.com/2014/01/02/the-teaching-learning-paradox-revisited-part-3-solving-the-paradox/> (visited on 02/19/2019).
- [4] Pavla Dráždilová et al. “Computational Intelligence Methods for Data Analysis and Mining of eLearning Activities”. In: *Computational Intelligence for Technology Enhanced Learning*. Ed. by Fatos Xhafa et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 195–224. ISBN: 978-3-642-11224-9. DOI: 10.1007/978-3-642-11224-9_9. URL: https://doi.org/10.1007/978-3-642-11224-9_9.
- [5] Cristóbal Romero and Sebastián Ventura. “Educational Data Mining: A Review of the State of the Art”. In: *Trans. Sys. Man Cyber Part C* 40.6 (Nov. 2010), pp. 601–618. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2010.2053532. URL: <http://dx.doi.org/10.1109/TSMCC.2010.2053532>.
- [6] Lipo Wang. *Data Mining with Computational Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2009. ISBN: 364206387X, 9783642063879.
- [7] Filip Neven and Erik Duval. “Reusable Learning Objects: A Survey of LOM-based Repositories”. In: *Proceedings of the Tenth ACM International Conference on Multimedia*. MULTIMEDIA ’02. Juan-les-Pins, France: ACM, 2002, pp. 291–294. ISBN: 1-58113-620-X. DOI: 10.1145/641007.641067. URL: <http://doi.acm.org/10.1145/641007.641067>.
- [8] Pithamber Polsani. “Use and Abuse of Reusable Learning Objects”. In: *Journal of Digital Information* 3.4 (2006). ISSN: 1368-7506. URL: <https://journals.tdl.org/jodi/index.php/jodi/article/view/89>.
- [9] Sean W. M. Siqueira, Ma Helena L. B. Braz, and Rubens N. Melo. “Developing Learning Objects through Concepts and Contexts”. In: *Advances in Web-Based Learning - ICWL 2003*. Ed. by Wanlei Zhou et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 485–496. ISBN: 978-3-540-45200-3.
- [10] Joyce Koh. “Designing and integrating reusable learning objects for meaningful learning: Cases from a graduate programme”. In: *Australasian Journal of Educational Technology* 33.5 (2017). ISSN: 1449-5554. DOI: 10.14742/ajet.3072. URL: <https://ajet.org.au/index.php/AJET/article/view/3072>.
- [11] “IEEE Standard for Learning Object Metadata”. In: *IEEE Std 1484.12.1-2002* (2002), pp. 1–40. DOI: 10.1109/IEEESTD.2002.94128.
- [12] Lorna Campbell. *Learning Object Metadata (LOM)*. DCC Digital Curation Manual, 2007. URL: <https://www.era.lib.ed.ac.uk/handle/1842/3334?show=full>.
- [13] Xavier Ochoa, Gladys Carrillo, and Cristian Cechinel. “Use of a Semantic Learning Repository to Facilitate the Creation of Modern e-Learning Systems”. In: *Proceedings of the XV International Conference on Human Computer Interaction*. Interaccion ’14. Puerto de la Cruz, Tenerife, Spain: ACM, 2014, 92:1–92:7. ISBN: 978-1-4503-2880-7. DOI: 10.1145/2662253.2662345. URL: <http://doi.acm.org/10.1145/2662253.2662345>.
- [14] Graham Klyne and Jeremy J. Carroll. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. Tech. rep. URL: <http://www.w3.org/TR/rdf-concepts/>.
- [15] *Apache Jena - A free and open source Java framework for building Semantic Web and Linked Data applications*. URL: <https://jena.apache.org/> (visited on 02/19/2019).
- [16] Mandeep Sidana. *Types of classification algorithms in Machine Learning*. 2017. URL: <https://medium.com/@sifium/machine-learning-types-of-classification-9497bd4f2e14> (visited on 02/19/2019).
- [17] *SciPy*. URL: <https://www.scipy.org/> (visited on 02/19/2019).
- [18] *Scikit-Learn*. URL: <https://scikit-learn.org> (visited on 02/19/2019).
- [19] *Tensorflow*. URL: <https://www.tensorflow.org/> (visited on 02/19/2019).
- [20] *The Microsoft Cognitive Toolkit*. URL: <https://www.microsoft.com/en-us/cognitive-toolkit/> (visited on 02/19/2019).
- [21] *PyTorch*. URL: <https://pytorch.org/> (visited on 02/19/2019).
- [22] *Keras*. URL: <https://keras.io/> (visited on 02/19/2019).
- [23] R. Boncea, A. Zamfiroiu, and E. Mitan. “PROPOSING ALGORITHM TO IMPROVE STUDENT EVALUATION PROCESS”. In: *EDULEARN18 Proceedings*. 10th International Conference on Education and New Learning Technologies. Palma, Spain: IATED, 2018, pp. 5799–5805. ISBN: 978-84-09-02709-5. DOI: 10.21125/edulearn.2018.1398. URL: <http://dx.doi.org/10.21125/edulearn.2018.1398>.