Title:
Adaptive Learning Web Application to Improve CAD Learning in Engineering

Authors:
Pedro Victor Gabriel Cerna, Universitat Politècnica de Catalunya, Spain
Antoni Perez-Poch, Universitat Politècnica de Catalunya, Spain
Francisco Alpiste Penalba, Universitat Politècnica de Catalunya, Spain
Jordi Torner Ribé, jordi.torner@upc.edu, Universitat Politècnica de Catalunya, Spain

Keywords:
Adaptive learning, computer-aided design, engineering education, and web-based applications.

DOI: 10.14733/cadconfP.2024.110-114

Introduction:

One of the biggest problems for students who use CAD systems is the acquisition and reinforcement of special skills and industrial standards that allow them to develop 3D parts more effectively. Additionally, given that each student develops these skills in a highly personalized manner, they require tools that help them in this process and are adapted to their needs (adaptive learning)[1], [2], [3], [4].

This article proposes the utilization of a web-based review system that adapts to the needs of students using AI engines, as described by several authors [2]. This tool has been tested to improve the grades of graphic expression students with good results.

As a final result, the tool uses EventSource technologies in combination with heuristic systems to produce a predictive algorithm that is able to adapt in a personalized way to the student presenting the content and adjust it to their cognitive needs.

Thus, we propose a system capable of providing a personalized agenda to each student based on the concepts required for the use of CAD systems.

Main Idea:

The idea of the system is to enable students to learn and review the concepts necessary to use CAD systems, from the theoretical part of the concepts to a particular part of the CAD systems in which the student wants to specialize.

The adaptive learning system requires a set of components that include those shown in Fig. 1, and that contemplate from contents ordered in a hierarchical way in categories, subcategories, content creation panels, student progress tracking panels, and gamified systems that present the exercises and are able to adapt in real-time to the user, authors the ones cited in these articles [1], [3], [5], [6] indicate the need for education systems to adapt to the reality of today's rapidly changing technology.

The method consists of the student performing five activities daily and not using more than 17 minutes for this purpose. Each time the student finishes each activity, it is corrected in the evaluation system, and the next content to be presented to the student is determined. The complexity of the content is adjusted automatically, depending on the performance of the previous activity; if this activity has poor or very poor performance, the student will be made to repeat it, but with a lower level of complexity to ensure that the student understands the concept of learning or review.

As the contents to be studied are ordered in the manner described above, they are ordered in independent evolutionary lines of content that work independently and are referred to as routes. With this, the system guarantees that the contents with greater difficulty will work independently from those with which the student has no problems.

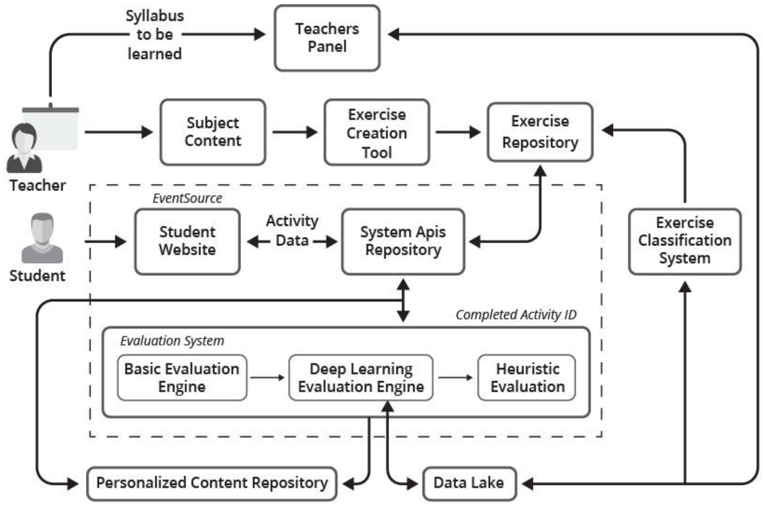

Fig. 1: Structure of the internal components of the adaptive learning web system.

Before a student can utilize the system, it is essential to classify and present the content to be learned in visual exercises that offer multiple representations, such as Button Panel, Button Panel with Audio, Button Panel with Time, Button Panel with Text, Memory, Relate, Drag and Drop Text, Drag and Drop Order, Reading, What's Missing, Word Search, Grid, Puzzle, Word Count, Reading and Questions, Reading Speed, Drag Drop Sentences, Drag Drop Groups, Speaking and Cloze Text. These various forms of representation are referred to as engines. The engines have been grouped to focus on fundamental aspects of understanding Computer-Aided Design (CAD) systems. The Visual Pattern Recognition Engines, for instance, aid in the visual comprehension of drawings and 3D parts, as depicted in Figure 2, and the engines of attention and understanding of the regulations that involve the creation of 3D prototypes such as ISO and UNE standards see Figure 2.

The method used to improve the understanding of CAD systems and the knowledge associated with them consists of the student performing five activities daily and not using more than 17 minutes for this purpose. Each time the student finishes each activity, it is corrected in the evaluation system, and the next content to be presented to the student is determined.

The time of 17 minutes is the result of an experiment described in [4] in which the students performed the daily activities corresponding to each of the above-mentioned sub-subjects. The average time spent by the students per day was 17 minutes, and the time per activity was 4 minutes except for activity one, which is a mental warm-up and whose exercises are not directly related to the subject to be studied, only aim to introduce the student to the various types of engines that will be found in each question and whose average duration was 50 sec

The complexity of the content is adjusted automatically, depending on the performance of the previous activity; if this activity has low or very low performance, it will be repeated to the student but with a lower level of complexity to ensure that the student understands the concept of learning or review. on the other hand, the web system adapts the complexity in real-time to the exercises of the activity that the student is performing; this complexity aims to increase or decrease the complexity. Once each activity has been performed, the results are sent to a task manager to be sent to the evaluation process; this allows the system not to collapse due to overuse of users.

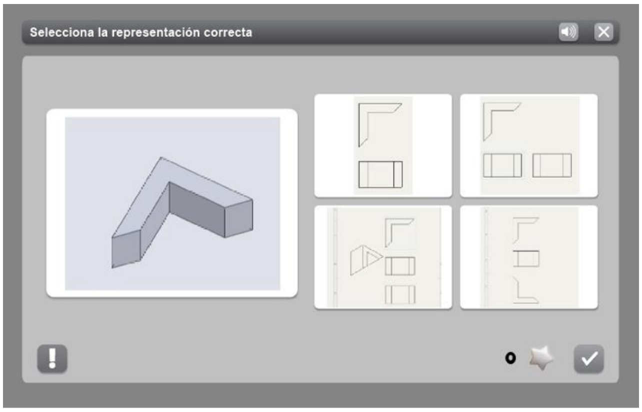

Fig. 2: Example of content presentation in which you are working and representing a 3D object.

The evaluation system works in three phases: the first is a general evaluation that evaluates what the student has done with respect to what the average of students who have done the activity, then goes through a process of deep learning that works with a neighborhood analysis algorithm (KNN) as described in [7], [8], [9], [10]; upon completion, it looks to see if the student has problems in other areas of the syllabus or other sub-subjects of the syllabus, and recommends a general level change if that is the case, e.g. figure 3.

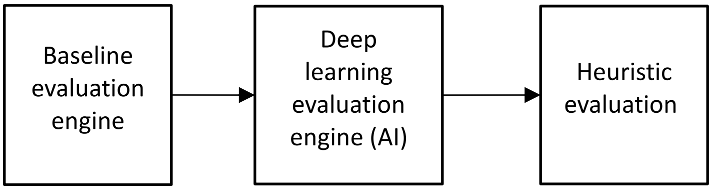

Fig. 3: Steps in the evaluation system.

Baseline evaluation engine
The purpose of this phase is to know if the student has successfully completed an activity, taking into account the number of successes and errors made in the time of the activity, using the average of the exercises done by the students throughout the last year, in this case the school year 21-22.

The evaluation equation (2.1) of this phase uses MId, it is the average number of exercises done at that level by the students in the last year.

$$effectivity = \frac{\sum successes - Kr * \sum failures}{MId} \tag{2.1}$$

Where Kr is a number that is initialized in 1, the 0.2, and 0.4 that are multiplied, imitate the current evaluation of the exams, in which each time you perform an exercise repeatedly, each failure has greater relevance, and on the other hand for each success made more than once has a lower weight.

If the evaluation is less than 6, the system assumes that there is some kind of problem with the student, which can range from simple oversight to the student not having the basics to take that content; in all cases, the system allows the user to perform the same level twice but with the difference that each time the exercises that were categorized as less difficult are presented, this categorization is done once a month, using the statistical data of use of each of them.

Deep Learning evaluation engine

The purpose of this evaluation phase is to know if the students who have similarities to the current student have the same behavior proposed by phase one; otherwise, learn and recommend the next activity to learn.

The system was developed using the K-Nearest Neighbors (KNN) algorithm, as described by Zhang in detail [7]. This algorithm was chosen because it requires minimal data to start learning and identifying similarities between users. In order to recommend the next level for a learner (Ui), the model must consider the behavior of other users who have shown similar patterns in the last four levels, from here we say number of levels to be considered in the vectors is k.

To implement this evaluation engine, vectors of variables were created: time spent performing the level, number of correct exercises, number of incorrect exercises, efficiency achieved in the level, branch-course-level structure, difficulty of the exercises performed in the level, difficulty of the exercises performed in the next level by other users with similar patterns, and GPS coordinates where the activity is performed.

Heuristic Evaluation

The function of this phase is to validate that the user who has a performance less than 60% obtained in the two previous phases does not have problems in more than three evolutionary paths that describe the content of the subject to learn if this is the case, the system can jointly lower the user to a marked stage such as the general level of competence, or to the beginning of a previous course if the system contains more than one concatenated course such as could be graphical expression 1 and graphical expression 2.

To validate this method, an experiment was designed in which content was created for the subject of Graphic Expression of the Degree in Industrial Engineering at the Diagonal-Besòs campus of the Universitat Politècnica de Catalunya.

The subject of graphic expression allows the use of this system because it has a large theoretical component and allows the measurement of the results owing to specific tests, which can be expressed as review exercises in the method described.

The results showed that the use of the system not only improved the grades of the evaluations, but also improved heuristic decision-making on the content to be taught by incorporating them into their evaluation processes, resulting in an improvement in the design and development of 3D models.

Conclusions:

This work demonstrates that it is possible to successfully create adaptive learning systems within the contents of CAD systems. The results obtained from the application of the experiment at the Diagonal-Besòs campus of the Universitat Politècnica de Catalunya show an improvement in the grades of the theoretical exams, ranging between 6% and 8%. The level of satisfaction among the students who used the adaptive system was high, as shown by the results of the surveys carried out.

Acknowledgements:

We thank the Universitat Politècnica de Catalunya, teachers, and students who collaborated in the realization of this experiment and who have been very helpful.

References:

[1] Al-Othman, M. A.; Cole, J. H.; Zoltowski, C. B.; Peroulis, D.: An Adaptive Educational Web Application for Engineering Students, IEEE Access, 5, 2017, https://doi.org/10.1109/ACCESS.2016.2643164.

[2] Kahraman, H. T.; Sagiroglu, S.; Colak, I.: Development of adaptive and intelligent web-based educational systems, in 4th International Conference on Application of Information and Communication Technologies, AICT2010, 2010. https://doi.org/10.1109/ICAICT.2010.5612054.

[3] Martín E.; Carro, R. M.: Supporting the Development of Mobile Adaptive Learning Environments: A Case Study, IEEE Transactions on Learning Technologies, 2(1), 2009, https://doi.org/10.1109/TLT.2008.24.

[4] Victor, G. C. P.; Perez-Poch, A.; Penalba, F. A.; Ribe, J. T.: An Adaptive Methodology for the Improvement of Knowledge Acquisition by a Multimedia Web Tool, International Journal of Engineering Education, 39(3), 2023, 643–652.

[5] van Riesen, S. A. N.; Gijlers, H.; Anjewierden, A. A.; de Jong, T.: The influence of prior knowledge on the effectiveness of guided experiment design, Interactive Learning Environments, 30(1), 2022, https://doi.org/10.1080/10494820.2019.1631193.

[6] de Jong, T.; et al.: Understanding teacher design practices for digital inquiry–based science learning: the case of Go-Lab, Educational Technology Research and Development, 69(2), 2021, https://doi.org/10.1007/s11423-020-09904-z.

[7] Zhang, S.: Challenges in KNN Classification, IEEE Trans Knowl Data Eng, 2021, https://doi.org/10.1109/TKDE.2021.3049250.

[8] Lopez-Bernal, D.; Balderas, D.; Ponce, P.; Molina, A.: Education 4.0: Teaching the basics of knn, lda and simple perceptron algorithms for binary classification problems, Future Internet, 13(8), 2021, https://doi.org/10.3390/fi13080193.

[9] Maheswari, K.; Priya, A.; Balamurugan, A.; Ramkumar, S.; WITHDRAWN: Analyzing student performance factors using KNN algorithm, Mater Today Proc, 2021, https://doi.org/10.1016/j.matpr.2020.12.1024.

[10] Nugroho, A.; Riady, O. R.; Calvin, A.; Suhartono, D.: Identification of Student Academic Performance using the KNN Algorithm, Engineering, MAthematics and Computer Science (EMACS) Journal, vol. 2, no. 3, 2020, https://doi.org/10.21512/emacsjournal.v2i3.6537.