Automatic-Test-ACF: A New Evaluation Tool Based on Automatic Correction Filters

Mercedes Vitturini[1,2], Laura Benedetti[1,3], Perla Señas[1,2]

[1] Laboratory of Research and Development in Informatics and Education (LRDInE),
[2]Department of Engineering and Computer Science, [3]Department of Humanities,
Universidad Nacional del Sur, Av. Alem 1253, Bahía Blanca
{mvitturi, psenas}@cs.uns.edu.ar, {laura.benedetti}@uns.edu.ar

Abstract
This paper presents the results obtained with regard to the evaluation process in Web-Based Learning Systems. It shows the design of *Automatic-Test-ACF* in detail, an evaluation management system with Automatic Correction Filters that can be included as a complementary module in a Learning Management System. The purpose of this approach is to develop a tool to help teachers in student follow-up and evaluation, aiming at reducing course drop out and adjusting the number of students / number of tutors ratio. The design achieved is the result of several research projects that include the development of a prototype and its assessment in field experiments.

Keywords: Automatic Correction Filters, Web-Based Education, Evaluation

1. Introduction

The current relation between ICTs and educational processes aim largely at satisfying the expectations that exist on Web-Based Education (WBE). Adaptive and intelligent systems are mainly looked for. WBE introduces new variations in formal education models or assumptions; learning to learn, learning communities, continuous education, self-learning, promotion of the students' genuine interest, and collaborative learning have become relevant and have acquired a new meaning within this new context [1]. In the field of ICTs, new technological progress in constantly searched in order to be able to design and basically maintain personalized learning environments, with contents and materials adequate to the needs and desires of each individual. For this to be possible, there are still some problems to be solved, among others those related to evaluation, learner follow-up, course drop out, and the number of students / number of tutors ratio. With regard to this last item, it is important to point out that student follow-up, whether for motivation, evaluation or support purposes - indispensable in order to avoid course drop out- and the economic balance between investment (course material + fees + technological resources) and the results obtained (number of students who passed / number of students enrolled) still present a challenge that is not completely solved. In this context, the focus is specially on

evaluation, not only as a final step to check if the student has achieved the goals set, but also, and essentially, as a process inherent to every educational phenomenon that allows teachers to support learners in a safer way and to rectify the direction every time it is deemed necessary [3]. It is at this point that teachers urgently need the help of technology. Teachers need that all those tasks that can be done automatically and without teacher intervention, be effectively done in that way. In order to carry out the evaluation process, it is necessary, among other things, to have adequate instruments and a reliable correction system. It is widely known that it is not easy to design evaluation instruments that allow for the automatic correction of the solutions presented by students. In addition, in some cases issues arise for which still there is no solution. For example: evaluation of essays, problem-solving processes, non-literal text translations, or design of programming algorithms. For the cases mentioned above, an array of tests that can be corrected automatically and that should be passed by each student before he or she approaches those exercises that cannot be corrected automatically are being considered. That is, the array of tests with automatic correction works as a filter that only allows those students who have the essential previous knowledge to solve the problems that shall be corrected by the teacher to get through. Such a simple approach allows for a considerable reduction in the teacher-hours/number of students ratio, a problem that is not yet solved in WBE. The key of such improvement lies in the following: it is considered that until students built certain knowledge, assessed through the array of automatic correction tests, they are not ready to solve those exercises that shall be assessed by the teaching team.

2. Automatic Correction Filters

Given an application exercise E, a program which has as input a resolution of E, and that after the corresponding analysis and according to the correction criterion entered, gives a Boolean output is defined as E's automatic correction filter (ACF_E). When the output takes the true value for all the tests associated to an exercise, the teacher should perform manual corrections to complete the evaluation of the exercise; when the output takes the false value the system that invoked the filter shall inform, both the student and the evaluation module, the unsatisfactory result and the cause for it. In this way, the ACF acts as a sifter retaining those cases that do not meet the minimum conditions to go to the stage of manual correction of exercises and only allows to pass those with chances of being approved [9]. Using this mechanism it is possible to reduce considerably the teacher hours/student ratio, a value that is still significantly high in good quality WBE approaches.

The instructions for the development of ACFs were to design a software tool to act as a correction aid. In this way, those solutions that do not meet the minimal conditions for the students to go to the stage of manual correction exercises are automatically identified and ACFs only allow those answers with chances to pass to continue. When a student does not pass a test, the ACF informs the student. It can be programmed so that in addition to pointing out the mistakes made, the ACF requests the student to send a new solution or to activate the sending of a new test on the same

subject. This tool allows for a significant reduction in the teacher-hours/student ratio in average or large groups.

The concept map in Figure 1 is a synthesis of the original model. It shows the concepts of the domain that constitute the basis for the construction of the prototype and the way in which they relate to each other. The reformulation of this original model is presented later in this paper, sections 3 and 4.

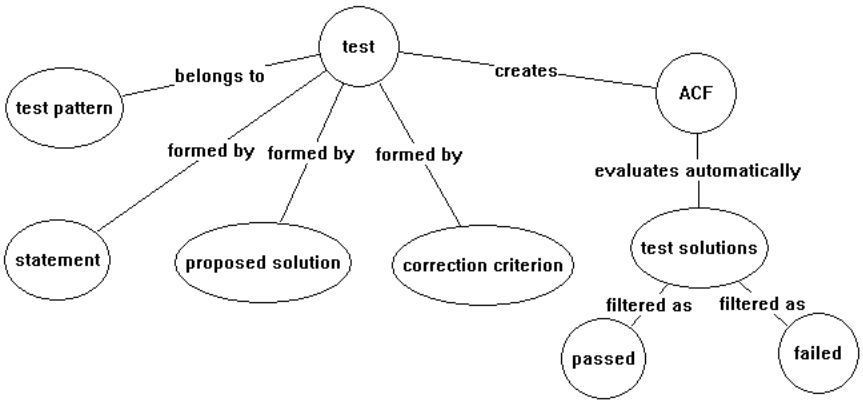

Figure 1: Original Test Model with ACFs

This approach is further complemented with a typification of exercises and different evaluation criteria that can be associated to each of them**.**

3. The Automatic-Test-ACF tool

The *Automatic-Test-ACF* tool is presented to be included as a complementary module in a Learning Management System (LMS). It is an extension of the ACF prototype detailed under [10]. The types of tests to be considered and their correction criteria are analyzed next. A hierarchy is built from the factoring of common properties.

3.1. Typification of Tests

The construction of the ACF prototype was based on a Web-Based Education project applied to University English Reading Comprehension Courses. A set of nine types of different exercises suggested by the professor responsible for the course was considered, see Table 1.

Once the nine types of tests that the tool had to manage were defined, they were carefully studied and analyzed [2]. Then, when these data characteristic of each type of exercise were known, similarities and differences were identified. Thus, common patterns and subsets of characteristics shared by several types of tests were found.

- Exercises MS, SS, and SW basically require the execution of the same pair of operations: first, selection of a set of elements -sentences or words- included in the text of the statement and characterized by a special feature and, then, considering the elements pointed out before, selection of a new set of elements that should also be considered as essential, that is, that should necessarily be present in any correct solution of the exercise. Both operations common to these three types of exercises can be generalized in only two operations: *Select-Special-Elements* and *Select-Essential-Elements*.
- In exercises MC, PAT, REF, and NPh there is one operation repeated in all of them, that is the operation related to the input of scores for certain elements included in the solution of the exercises. For this reason, the implementation of only one Input-Score operation that could be reused for each of the four types of exercises according to the elements present in each particular case was the decision adopted.
- The T/F and Q&A exercises are very specific and no common characteristics between the two could be factored; thus, the operations were individually implemented in separate modules.

Table 1: Typification of Exercises

Type of Exercise	**Identifier**
Mark Main Sentences	MS
Select an Answer (Multiple Choice)	MC
Indicate the True Value of Statements	T/F
Identify Text Patterns	PAT
Identify Reference Words	REF
Translate Noun Phrases	NPh
Identify the Sentences that have Certain Characteristics	SS
Identify Words that have a Certain Characteristic	SW
Answer Questions including Key Words in their Answers	Q&A

3.2. Exercise Correction Criterion

With regard to the correction criterion of each type of exercise, it was determined that there are essentially two possible types of criteria. Both have common attributes such as the maximum score possible, the minimum score necessary to pass, and the messages for both passing and failing students. However, both types of correction criteria determined make a distinction in the way they perform the correction of exercises. A detailed observation of each of them will clarify which is the correction criterion used in each case:

- The first one is applied in those exercises in which an absolute correction shall be performed, that is, where each element that is part of the solution shall be simply considered as correct or incorrect, and, in addition, where there will be only one score associated to those correct elements and another one applicable to incorrect elements. SS, MS, SW, T/F, and Q&A exercise types have this correction criterion.
- The second type of correction criterion observed is related to a more flexible correction, in which more than one possible combination may exist for the elements

included in the solution and, therefore, this can be considered correct, incorrect or partially correct. But, how is this distinction made? Basically, weighing each element: as correct, when the solution assigned matches the highest score; as incorrect, when the solution associated has the lowest score; and as partially correct, when the solution assigned has an intermediate score in between the highest and the lowest score possible. REF, PAT, NPh, and MC exercise types all have this correction criterion.

3.3. New Exercise Hierarchy

Based on the study carried out on the nine exercise types proposed and, taking into account the conclusions presented as the result of previous studies on the subject [2], there arises this classification.

Exercise comprises all the characteristics that are common to all the exercises that the tool manages. Basically, it defines the data structure that the exercise shall have, the operations to administer such structure, and a few operations generic to all the exercises, operations that are independent from the type of exercise.

There are three descendent classes of the *Exercise* class. Each one represents a group of exercises with a common characteristic. The division of the whole set of exercises into three groups was performed by distinguishing the activity for input of the exercise statement. That is, the exercises have been grouped into three sets according to the activity executed in that second phase. As previously mentioned, three possible activities were distinguished; therefore, three new types of exercises are defined, all descending from the *Exercise* type.

The first group of exercises was called *Simple Statement Exercise* and includes those exercises in which the activity to be carried out during the second input stage of the statement is to request the input of a special feature or condition that the elements included in the solution of the exercise should have. This can be considered a clear and simple activity since the teacher should only input one characteristic that the elements could have, this is the reason why the name of this type of exercise includes de adjective "*simple*". There are three exercises included in this category: Mark Main Sentences (MS), Mark Sentences with Certain Characteristics (SS), and Mark Words with Certain Characteristics (SW). A type including all the instances for each type of exercise was created for each of them. They were called *MS Exercise*, *SS Exercise*, and *SW Exercise* respectively. As a result of the study performed on them it was also deduced that the exercise Mark Main Sentences is a special case of the exercise Mark Sentences with Certain Characteristics, in which the characteristic requested is to be a main sentence. For this reason, the *MS Exercise* type is directly inherited from the *SS Exercise* type, that, in turn and together with the *SW Exercise* type, are inherited from the *Simple Statement Exercise* type. It is also worth mentioning that for the exercises that are instances of the *MS Exercise* type, the activity in which the input of the special condition of the elements in the solution is requested shall not be executed because such condition is implied in the nature of the exercise.

The second group of exercises was called *Exercise with Statement with Input of Elements* because it includes three types of exercises in which, as part of the input of a statement, the teacher is asked to input which are the elements that he or she wishes to

add to the statement of the exercise he or she is working with. This category includes the following exercises: Multiple Choice (MC), the exercises of this type shall be instances of the *MC Exercise* type and the elements to input are the items and options that shall be added to the statement of the exercise; True/False Clauses (T/F), the exercises shall be instances of the T/F Exercise type and the elements to input are the clauses to add to the statement of the exercise; Questions (Q&A) including Key Words in their Answers, the exercises shall be instances of the *Q&A Exercise type* and the elements to input shall be the questions that shall appear in the statement of the exercise.

An important observation to be highlighted is that at first it was thought that the exercise with clauses could be considered a particular multiple choice case in which each element had two fixed options associated, true and false, instead of several options. This is true but only if the statements of the exercises are compared, when analyzing the two remaining components, the solution and the correction criterion, it is found that both types of exercises differ greatly. For example, for the multiple choice it was decided that each option of an item could have a score associated to distinguish between correct and incorrect options, however, this policy in the case of clauses is unnecessary because there are only two possible values, where one is correct and, therefore, the other one is considered incorrect. Then, since more differences than similarities were found between both types of exercises, it was decided against establishing any inheritance relationship between them. The types that correspond to these three types of exercises are directly inherited from the *Exercise with Statement with Element Input* type.

The third group of exercises that is directly inherited from the *Exercise* type is the one called *Exercise with Statement with Choice of Elements*. The name comes from the fact that as part of the input of the statement of any exercise included in this group, the teacher is asked to choose certain elements present in the text of the statement, so that only those elements selected shall be taken into account for the rest of the input stages of the exercise. These elements shall be added to the statement in a specific way, in addition to being shown within the text that includes them. This category includes the following exercises: Reference Words, the exercises of this type shall be instances of the *REF Exercise* type and the elements to be selected shall be certain words from the text that are considered as referents for certain expressions; Pattern Identification, the exercises shall be instances of the *PAT Exercise* type and the elements to be selected shall be some paragraphs of the text for which later their patterns shall be asked; Translation of Noun Phrases, the exercises shall be instances of the *NPh Exercise* type and the elements to select shall be certain noun phrases included in the text that later should be translated. The types that correspond to these three types of exercises are directly inherited from the *Exercise with Statement with Choice of Element* type.

4. Architecture of the Automatic Correction Filters Tool

In the design of the *Automatic-Test-ACF* tool, an architecture formed by three sub-systems or modules that share a traditional ACF database was defined. The modules

considered are: the test management module with ACFs, the automatic evaluation module and the follow-up module. The organization of these modules is shown in Figure 2.

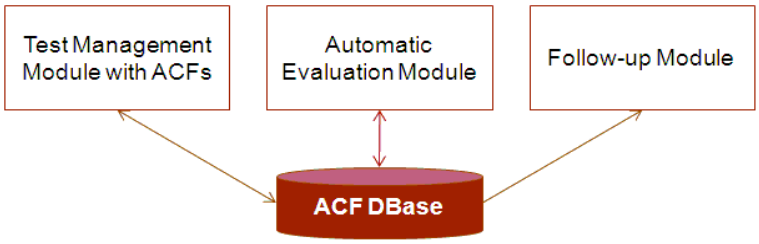

Figure 2: Architecture of the *Automatic-Test-ACF* tool

The first sub-system is responsible for supplying the services to create and administer exercises or tests with ACFs, that are stored in the ACF database. The evaluation module leads the sending and correction process for the test solutions sent by students using ACFs. Basically, it comprises the selection and provision of the corresponding form to the group of students selected by the teacher, the correction of the answers received and the later sending of the appropriate message to the students; the results of this automatic evaluation also update the ACF database with the record of the students' activity. Finally, the follow-up module supplies features for a teacher or student to search and view the information on the results of general and/or particular automatic evaluations. The ACF database serves different purposes: it stores the ACF information that allows for the automatic correction of the exercises solved by the students, it permanently stores tests and their ACF associations so that they could eventually be reused, and it keeps a detailed track of the results obtained in the corrections. In the following sections the services proposed for each sub-system according to the user profile are explained in detail.

4.1 Test Management Module with ACFs

The Test Management Module with ACFs provides services only to the teacher profile. The main work of a teacher is to *create a test with ACFs* following one of the exercise patterns supplied by the *Automatic-Test-ACF* tool. The input of a test is a guided work that is carried out following the three following basic activities: input test statement and instructions, add solution and, finally, add correction criterion or rules.

For example, there is a mathematical problem to be solved using equation systems, a T_1 test could be “mark the sentences in the statement that contain relevant data for solving the problem”. The input process for this exercise includes: uploading the T_1 statement and indicating that the test responds to the exercise pattern mark sentences in a text (SS). Then, the teacher inputs the expected solution; in this case, he or she marks the sentences of the statement that satisfy the instructions. Finally, the teacher inputs the correction criterion: points to add for each sentence marked correctly,

points to deduce for each incorrect sentence, points to deduce for each sentence that is not marked and minimum passing score. As an option, additional rules can be set such as sentence/s that should necessarily be marked for the solution to be considered correct. Finally, the ACF is completely defined by indicating the action/s to follow in case the solution fails to pass (either send the correct solution to the student or request to re-send the exercise) and the messages to send automatically to the students that pass/fail. Once these activities are confirmed, the T_1 test and its ACF_1 are stored in the ACF database permanently becoming part of the test library. As a result of the process of creating a new test with ACFs the tool automatically generates the *sample form to input the* SF_1 solution. A copy of said form is then sent to each student for them to input their answers. This is explained in further detail in the automatic evaluation module section.

Finally, the other group of activities available in the Test Management with ACFs sub-system is *browse the test library* activity group. Under this task, traditional features such as *browse* or test search with ACFs stored are included. This was done in this way since in the definition of tool requirements it could be seen that often the same statements can be reused in different tests. For the T_1 example here, another test, T_2, could be answered by using a true or false criterion (T/F) to the assumption on the number of unknown variables reusing the same statements.

4.2 Automatic Evaluation Module

The automatic evaluation module is responsible for the main feature for which the *Automatic-Test-ACF* tool was designed: automatic correction of test array solutions. For this, the tool supplies different services to the teacher profile and the student profile, as can be seen in diagram of UML use cases of Figure 3.

The process is triggered when teachers schedule a new sending of exercises or tests to their students. The basic activities to carry out this task include: choose the group of students addressed, select the test/s that form the array to send and define the delivery calendar. The selection of students allows the teacher to personalize the student/s who will receive the forms with the exercises to solve. It includes features to choose all the students, the students in a group or individually selected students. It is also possible to schedule the dates when the exercises will be visible and/or until when the solutions will be received. This is an option for the teacher to organize the course.

Then the test/s with ACFs to be sent are selected. In this case the tool *creates a form instance* FI_i for each student on the addressee list and for each test according to the associated form pattern *FP*. Metadata are added to each form instance that will then allow the ACF to associate the solution with the student who did it. In the array example that is being followed with tests T_1 and T_2, each student shall then receive two personalized forms $IF_{1\_student}$ and $IF_{2\_student}$ In the first case, the students should mark the sentences that satisfy the instruction of having data that are relevant to the solution (T_1) and in the second they should assign the true value on the number of unknown variables (T_2).

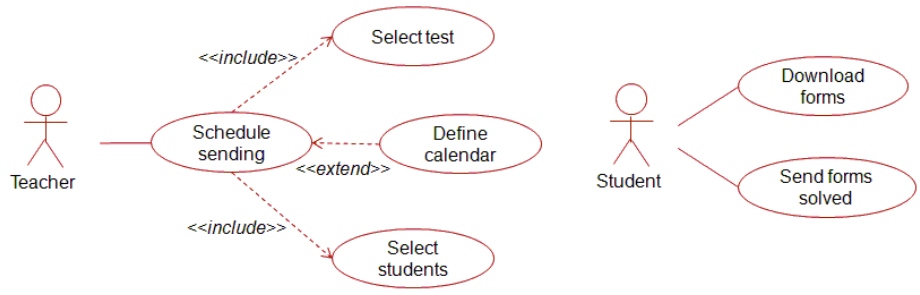

Figure 3: Automatic Evaluation Module- Diagram of Use Cases

The students from their profile shall be able to *download their forms* where they should have to input their solution. When they decide to do so, they shall send the form/s with the solutions. The sending of a form starts the following series of automatic activities:

1. $IF_{i\_student}$ is stored in the ACF's database.
2. The ACF corrects the solution received.
3. The results of the correction are stored in the ACF database.
4. The notification of the result is sent to the student (pass or fail) together with the message previously defined by the teacher.
5. If suitable according to the test's correction criterion, additional information is sent (for example, the student is asked to send the correct solution or any other suitable remedial action).

4.3 Follow-up Module

The follow-up module of the tool *Automatic-Test-ACF* is responsible for offering search and view services for the correction results stored in the ACF database. These services are available both for the teacher and student profile, except that students can only see their own personal information while teachers can see the information of all the students.

This module, unlike the previous ones, does not modify the information stored in the ACF database; it recovers the data generated by the automatic evaluation module. The approach is to offer inquires on the corrections stored with the possibility to filter[1] or define conditions on the set of results searched and give options to show the information either as text or graphics. For example, the teacher may wish to inquire for general results about the test T_1.

[1] In this case, the term filter is not related to ACFs but to the conditions on the results expected.

5. Conclusions

In this paper the detailed design of a new ACF tool has been introduced. It is based on a simpler previous approach already implemented and tested in a b-learning educational experience. The new design overcomes the aspects pointed out by the different actors that took part in the experience mentioned above.

From a pedagogical point of view, this approach allows to strengthen the concept of previous learning needed for the construction of significant learning. From a computational point of view, this is included within the framework of prototype-based developments. With regard to Web-Based Education Management, it is considered an important contribution for reducing the teacher-hours/student ratio.

Bibliography

1. Andrej, K. and Bieliková, M. "Improving adaptation in web-based educational hypermedia by means of knowledge discovery. Proceedings of the sixteenth ACM conference on Hypertext and hypermedia". 2005.
2. Benedetti, L. Vitturini, M. Fernández Coria, C. and Señas, P. "Evaluating Learning Objects for E-Learning: Automatic Correction Filtres". Proceedings of the V Conferencia Internacional sobre Multimedia y TICs en Educación. España. 2006.
3. Camilloni A. R. W. de "La evaluación de los aprendizajes en el debate didáctico contemporáneo" Ed. PAIDOS. 1999.
4. Changjian Fu, Jian Zeng, Zhihua. "Leng Empirical Research on the Evaluation Model of Distance Education Based on the Gray and Rough Set Theory in China". International Conference on Computer Science and Software Engineering. 2008.
5. Derntl, M., Mangler, J. "Web Services for Blended Learning Patterns". IEEE International Conference on Advanced Learning Technologies (ICALT'04), 2004.
6. Millard, D., F. Tao, K. Doody, A. Woukeu, and H. Davis, "The Knowledge Life Cycle for e-learning". International Journal of Continuing Engineering Education and Lifelong Learning: Special Issue on Application of Semantic Web Technologies in E-learning, 2006. 16(1/2): pp. 110-121.
7. Sancho, P., I. Martínez, and B. Fernández-Manjón. "Semantic Web Technologies Applied to e-learning Personalization in e-aula". Journal of Universal Computer Science, 11(9), pp. 1470-1481. 2005.
8. Sheard, J. "E-learning communities: strategies for establishment and management". Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education ITiCSE '04, Volume 36 Issue 3. 2006.
9. Vitturini, M., Benedetti, L. y Señas, P. "Filtros de Corrección Automática como Objetos de Aprendizaje Evaluativos para Sistemas Educativos Basados en la Web". Proceedings of CACIC-2005, Vol 1, pp.814-825, 2005.
10. Vitturini, M., Panozzo, B. y Señas, P. "Sobre la Confluencia de dos modelos: Educación Basadas en la Web y Formación Continua". VI International Conference on Engineering and Computer Education. Buenos Aires. 2009.
11. Yong Yang, Guoyin Wang, "An Evaluation Model for Web-Based Learning Support Systems" IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), pp.680-683. 2005.