Tracking the Student Online Learning through a Tree of Objective Questions

Authors:

Luciana A M Zaina, University of São Paulo, Escola Politécnica, Departamento de Eng. da Computação e Sist. Digitais, Brazil, lzaina@larc.usp.br
Wilson V Ruggiero, University of São Paulo, Escola Politécnica, Departamento de Eng. da Computação e Sist. Digitais, Brazil, wilson@larc.usp.br
Graça Bressan, University of São Paulo, Escola Politécnica, Departamento de Eng. da Computação e Sist. Digitais, Brazil, gbressan@larc.usp.br

Abstract *— The tool presented in this work allows the teacher to create objective questions in order to get a formative evaluation having as its main goal to track the development of the student in the learning process. The aim of the tool is to create a dependence among test questions, building a tree of questions where the teacher can verify the "path" followed by the student during the test, getting a detailed tracking of the student test resolution to support the teacher evaluation. The teacher is allowed to construct a tree of questions where he can get concrete data about the student's performance. The learner can be led through different paths in the same question, which are determined by his answer providing the teacher with important information about the student learning. Besides this, the tool permits the teacher to build and to apply multiple-choice type questions to problems that have numerical values and formulas or algorithms related to their resolution, modifying the numerical values in the question description every time it is presented. The question correction is made automatically and both the teacher and the learner can analyze the path followed during the test in the feedback information.*

Index Terms *— formative evaluation, objective questions, parameter values, tracking the student evolution.*

INTRODUCTION

Tracking the student evolution during a course in the Web is not an easy task. Different ways have been studied different ways to identify lack in the students' learning, but teachers agree that there is not only one solution, and mainly that there is not an ideal solution to evaluate the student's knowledge. Forum, chat, discursive exercises are tools which could be used to track the student evolution in a Web course.

It is possible to find in the Web a variety of tools to support the construction and application of objective tests. The types of questions presented in these tools are multiple-choice, fill-in the blanks, crosswords, etc, being the multiple-choice question type the most frequently used. The multiple-choice type may not be a good option if the problem has numeric values associated with it, because if the student does the test more than once, this type of question may encourage the learner's memorization [1], [7],[8].

In some kinds of problem such as in numeric ones it would be better if the values related to the question could be changed. If each time that the question is presented to the student, the test tool sorts out the values that will be used to solve the question, the learner is forced to understand the problem in order to solve it [6]-[8]. The use of data parameterization allows the teacher to develop more dynamic questions [3] where the right answer of the question is always changed.

This paper presents a tool called AvaliaOnline, where the teacher can build a tree composed of objective questions, leading the learner through the tree of questions according to what he answered in the previous question. The teacher can link questions with dependent data to track the student's reasoning during the test resolution. For instance, the answer of a specific question could be used in the next question of the test. In this model, the teacher will have information to evaluate the learner's evolution step by step, identifying points and conceptions where he had learning problems. Although the teacher may use this benefit he should design the questions (which are based on the student answer) during the test planning, because the tool works with multiple-choice questions. For instance, if the student chooses the correct alternative it will lead him to a determinate question. On the other hand, if he chooses an incorrect alternative it will lead him to another question. It is important to highlight that it is possible to create questions that have no relation to one another.

Besides tracking the student through his "path" of questions, the tool allows the teacher to create and to apply tests, which will have parameterization questions. To use this tool the teacher must know a programming language, because he must be able to construct algorithm, which will be necessary to solve the problem presented to the learner. The teacher determines the parameters related to the problem and the algorithm used to get the problem solution. Each time the question

is submitted to the student, the system makes a random choice to obtain the parameters' values and applies them to the algorithm, leading to the right answer alternative. The same student may solve the test more than once and the teacher can analyze the different results obtained.

In summary, AvaliaOnline permits the teacher to create and apply tests that have numerical values involved in the resolution, improving the student is tracking through a tree of questions.

ARCHITECTURE OF THE TOOL

The tool described in this work is independent of any kind of course management system and it may be or may not be plugged to a system. Two modules compose the tool: Question Module and Test Module. These modules work independently, that is, one of them can be executed without the other, but there is dependence between the data used in these modules.

The control to identify the author's question or test is performed through the user identification issued during the login in the system. Likewise, the student identification is the same of the user login. To aggregate this tool to an existing course management system it is enough to adjust the user control of the tools to another system.

The Question Module is responsible for the cataloging of questions that will be used by the Test Module to compose the tests. However, it is necessary to consider that the Test Module deals with the information related to the construction of the test as well as with the record of the results of the tests in the student registry.

First the paper will describe the modules of the system without focusing on the construction of the tree of questions. It will present how AvaliaOnline handles questions and the test. After that, it will explain how to create the relation among the test questions, building a tree of questions where it will be possible to track the student's reasoning.

Question module

The Question Module handles information related to the development and management of questions such as their inclusion, updating and deletion.

Data recorded by this module will support the construction of questions during the running of the test. When the author inserts a question, he must define the following fields:

- **Educational objective:** defines which educational objective the question must reach.
- **Description:** this field defines how the problem will be presented to the student. The description of the question may include images and animations beside the text. The text may depend on parameters and the author must use a special notation (Figure 1), which has a special meaning for the system, to indicate a parameter.
- **Algorithm:** is a formula or an algorithm written in a language to be interpreted by the system. The formula depends on the parameters defined in the text and it may use arithmetic operators and function calls available in the language. It is used to calculate the value to be associated to the right alternative.
- **Deviation:** is a value, which will be used to choose the values for the wrong alternatives around the right answer.
- **Parameters:** definition of the parameters to be used in the text of the question description and used in the algorithm to define the right alternative. There is no limit in the number of parameters for the problem.

It is necessary to inform the range of values for the parameter or to define an algorithm to calculate the parameter value as shown in Figure 2. When a parameter is composed of a formula or algorithm, the sub-module must generate the function that calculates the value for the parameter. The dynamic generation of the question consists of randomly choosing a value between the lower and upper limits. The author does not need to supply any of the alternatives because the system will generate dynamically all of them using the parameter values.

Test module

The Test Module is responsible both for the management of tests, with operations like inclusion, exclusion and updating of tests by the author, and for the presentation of the tests to the students.

Questions chosen during the test creation by the author are not restricted to only one objective but it is advisable to keep the questions restricted to few objectives. From a pedagogical point of view, too many educational objectives in a test may cause the analysis of the results to be very complex [1]. As the main purpose of the test is to follow the student's learning, it is better to have a subset of objectives to analyze each time. Introducing many questions with different objectives in the same test may disturb the analysis.

During the test assembling, associating weights to questions is allowed. In this way, it is possible to differentiate the importance of some questions in relation to others. One question may have different weights in different tests because it may be essential to one approach and secondary to another, depending on the target of the test. Although the kind of evaluation intended by the tool is formative, and not summative, the assignment of weights to questions helps the teacher to quantify the level of understanding of the student [8].

Besides the value of the questions, the teacher must inform the minimum grade the student must acquire. In case the student does not reach this minimum grade, he is allowed to do the test again. Besides specifying the minimum grade to be fulfilled, the teacher may also specify the maximum number of times the student can do a test.

When the students are performing a test, the system retrieves the stored questions, calculates the values of parameters, substitutes the parameters in the question description by the calculated values, and through the main formula or algorithm, gets the right answer to the question. Finally, the question is presented to the student as it is illustrated in Figure 3. Figure 3 presents two different views of the same questions solved by the same student, showing that question values are randomly chosen each time the question is exhibited.

The system performs the automatic check of the students' tests answers but the teacher is allowed to verify the right and wrong answer of the student as in the case where the student did not get the minimum grade of the test. This analysis is a support to the teacher, so he can have feedback of the student's evolution, which will not occur if the teacher has access only to the final grade of the student. Another aspect to be considered is the possibility to compare the results of the student every time he performs an evaluation. The learner is allowed to execute the test many times and the teacher can compare the student's answer in different times the test is executed [1][6][8].

The teacher may register all questions another create the test. It is important to highlight that questions don't need to be registered in the same objective test, because the teacher chooses the questions of different objectives during the test creation.

Building a tree of questions

The questions included on the test are called "root questions" and there is a number associated to each question that represents the order that the question will be showed during the test. This order number is called "main order". When a question is inserted in the test it does not have another question linked to it. The teacher should create the relation among the questions if he finds it necessary. In other words, there may be a test even if the questions are not dependent one another. there are not dependent questions.

The teacher may have a test, which is composed by linked questions and non-linked questions. When a question is related with another question, the teacher must connect one question with the right alternative and another question with wrong alternatives. That is, if the student is solving question "A" and he chooses the right alternative, it will lead him to question "B". On the other hand, if the student chooses a wrong alternative it will lead him to the question "C" as shown in Figure 4. If the teacher wishes he can link right and wrong alternative with the same question. Figure 4 presents question "F" that does not have a relation with other questions and it will be presented independently of the alternative that was showed before that.

A tree of questions is composed of a group of registered questions where the first question of the sequence is called "root question", because it is in the main level of the test. The tool supports the creation of more than one tree of questions and it is not necessary for the trees to be related, providing the flexibility to construct questions with or without dependency. Each "root question" has a sequential numerical value associated to it called "main sequence" and this numeration in "dependence questions" is called "secondary sequence". Both numerical sequences are used during the test exhibition. First the tool verifies the "main sequence" and if there is a dependent question linked to it, the tool takes the "secondary sequence" as the used sequence.

For instance, Figure F shows a group of questions, where questions "A", "F" and "H" are "root questions", having a "main sequence" numerical value associated with each of them. There are two trees of questions where questions "A" and "H" are "root questions" of different trees, "Tree 1" and "Tree 2" and the question "F" does not have dependency on another question. During the test exhibition the "root question" of "Tree 1", question "A", is shown in the begin. After that is the next question in the sequence will be available, analyzing if there is or not a "secondary sequence". Then the alternative chosen by the student will be checked and the tool shows question "C", in case the right alternative is chosen, or "D", in case the wrong is one chosen. Achieving the end of "Tree 1" the tool will look for the next "main sequence" which will be question "F", a question without dependence.

Besides the tree creation the teacher might indicate that the answer value chosen by the student must be used as a parameter of the next question in the tree sequence as shown in Figure 5. In other words, a question answer may be used in the next question description allowing the teacher to verify the building of the student reasoning during the test performance. Tracking the learner's reasoning step by step from answer values allows the teacher to get concrete data for a better evaluation.

CONCLUSION

The tool presented in this work allows not only the application of objective tests, but also and mainly the possibility to get concrete data to track the student evolution. The conception of tree of questions supports ways to verify step by step the learner reasoning. Nevertheless, it is worth pointing out that the teacher must design paths of test, in other words, the teacher determines the sequence of the test, based on the answer chosen by the student.

It is important to highlight that the parameterization of numeric values should have fundamental analysis to get a coherent result: the values of the parameters must converge in the problem resolution. When a problem is composed by many parameters that are sorted out in a determined range, these parameters values may not converge to a true result. This kind of problem is solved using an algorithm to analyze the impact of using the parameters together. This work did not have the goal of applying and analyzing the parameters values convergence when they are applied together. The tool allows the teacher to determine the parameters range through formulas or algorithms. The teacher is responsible for controlling the range of the parameters.

Some tests have been applied to graduate students providing a practical experience with this tool. Nowadays, we have researched about intelligent methods and collected data about the graduate student experience. We intend with these results to improve new resources to AvaliaOnline providing a more intelligent tool to track de student's evolution.

REFERENCES

[1] Bridgeman, S et al, "PILOT: An Interactive Tool for Learning and Grading", The 31th SIGCSE Technical Symposium on Computer Science Education - SIGCSE Bulletin, Vol 32, No 1, March 2000, pp. 139-143.

[2] Burger, C et al, "A Framework to Support Teaching in Distributed Systems", ACM Journal of Educational Resources in Computing, Vol 1, No 1, Spring 2001, pp. 1-13 (article 7).

[3] Driscoll, M, Web-Based Training, Jossey-Bass/Pfeiffer, 1998.

[4] Jiménez-Peris, R et al, "New Technologies in Computer Science Education", Computer Science Education in the 21th Century, Springer-Verlag, 2000, pp. 113-136.

[5] Korhonen, A et al, "Algorithm Simulation with Automatic Assessment", In: Proceedings of The 5th SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, Vol 32, No 3, September 2000, pp. 160-163.

[6] Simonson, M et al, Teaching and Learning at a Distance. Foundations of Distance Education, Prentice Hall, 2000.

[7] Thelwall, M, "Computer-based assessment: a versatile educational tool", Computer and Education,, Vol. 34, No 1, January 2000, pp. 37-49.

[8] Zaina, M, Luciana et al, "Tool to develop and apply objective tests", In: Proceedings of Frontiers in Education 2002, 2002, Available: http://fie.engrng.pitt.edu/fie2002/papers/1162.pdf.

FIGURES AND TABLES

FIGURE. 1
IDENTIFYING PARAMETERS USED IN THE FORMULA DURING THE QUESTION REGISTER

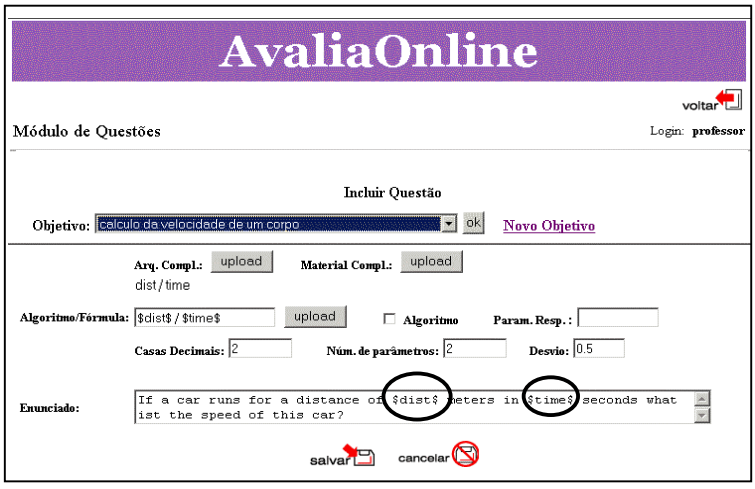

FIGURE. 2
REGISTERING DATA PARAMETERS.

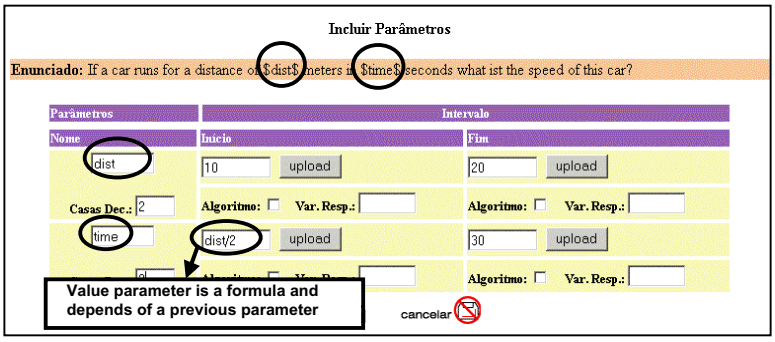

FIGURE. 3
DIFFERENT PARAMETER VALUES USED TO THE SAME QUESTION IN DIFFERENT MOMENTS .

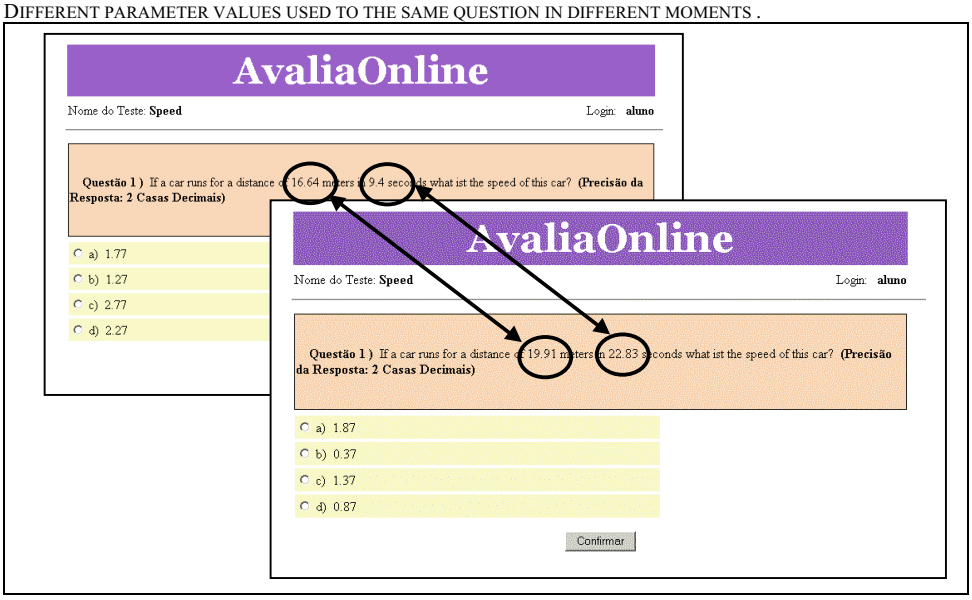

FIGURE. 5
SEQUENCE OF QUESTIONS COMPOSED BY TREES OF QUESTIONS AND INDEPENDENT QUESTION.

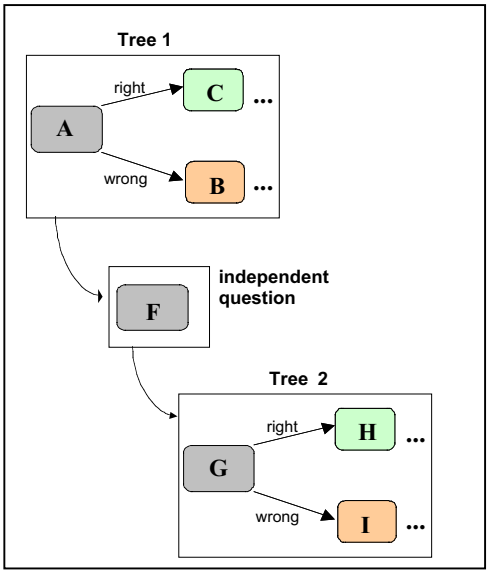

FIGURE. 4
ALTERNATIVE CHOSEN BY THE LEARNER IS USED AS A PARAMETER OF ANOTHER QUESTION.