Poster Paper in Proceedings of the 16th Annual NACCQ, Palmerston North New Zealand July, 2003 (eds) Mann, S. and Williamson, A. www.naccq.ac.nz

Automatic Assessment Generator

dr Xiaosong Li
Zhong Tang
Jun Li
SCIT, UNITEC
Auckland, NZ
xli@unitec.ac.nz

ABSTRACT

It is a usual practice when lecturers prepare an assessment they often look up the last couple of year's exam questions and modify them to produce a new one for current use. The previous experience showed that without the help of an intelligent automatic assessment system, this process was inefficient and the quality of the produced assessment could not be guaranteed. This poster will introduce the project: creating a software application, Automatic Assessment Generator, which can generate assignments or final exam papers automatically. We focus our discussion on the system architecture and its initial prototype.

Keywords: Assessment, education resources, question bank, generator, UML, ASP.NET

1. OVERVIEW

This project will be carried out in multiple phases. The first phase is to build up a question bank (a content resource). It is a fundamental work for the whole project. In this phase, all questions of the past year assignments and final exams for each course will be collected and classified according to the format, topic and weight of the questions. The examples of the question categories in format may be *coding questions*, *discussion questions*, *description questions* or *multiple choices questions*. The examples of the question categories in topic may be *function calling questions*, *variable scope questions*, or *data type questions*. The question weight could be *5 marks, 10 marks, 15 marks* and so on. By setting up some criteria, lecturers can submit a query for the system to generate the assessment paper. The query could be formatted based on question topic, format or weight. For example, the query "an exam with four multiple choices questions on variable scope, two description questions on function calling and three discussion questions on data type" will direct the system to search the question bank for all suitable questions and present them on the screen for tutor to choose. The system should allow the tutor to add new questions to the exam and the new questions will also be added into the question bank for future use.

The second phase of this project is to create an assessment quality improvement system (an assessment analyzer) that is an intelligent system. The main function of this subsystem is to record the results of all previous assessment and calculate the correct ratio for each question. This statistical information will be used as an input for the system's weight generating algorithm to update the weight for each question, so that the quality of the assessment could be improved.

The third phase is using content development tools to create learning objects from the content resource. This allows us to use the content resource in more contexts and facilitates the share and reuse of the content resource.

2. ARCHITECTURE

The application will be designed using the Unified Modeling Language (UML) and implemented on the ASP.NET platform with web client server architecture. The primary use case captures the relationships among the different components of the system. The lecturer is responsible for marking, adding new question, setting up criteria and submitting query for the system to generate assessment paper. All old questions and newly added questions will be organized into content resource.

3. INITIAL PROTOTYPES

In the first phase of the project, an initial prototype could help us understand the basic requirements of the system.

The question entry form allows a lecturer to add a new question or transplant an old question into the question bank. The generation of assessments could be further divided into the following three steps: a) setting up criteria; b) submitting a query and choosing questions from the list; and c) preview and produce the results.

4. THE FUTURE WORK

Currently we only consider the assessment generation for the two programming courses. Other courses could be included late upon the initial success. Potential extensive work is an online marking system. This is an intelligent system, which does critical reflection and provides critical feedbacks. Another potential extensive work is to extend the functionality of the system such that it is not limited to assessment.