Online Evaluation with Trainee-Adaptive Tests

CRISTEA BOBOILA
University of Craiova
Faculty of Mathematics and Computer Science
Al.I. Cuza Street, No. 13, Craiova RO-200585
ROMANIA

MARCELA S. BOBOILA
Stony Brook University
Computer Science Department
Stony Brook, NY 11794-4400
USA

Abstract: This paper presents an online application for test design and evaluation of trainees. We advance our research in two directions: activity flow modeling and adaptive test design. With activity flow modeling, we achieve high usability and structural coherence, while the adaptive test design method that we propose facilitates dynamic generation of tests based on topic relevance. Our method ensures the creation of adaptive tests that target to specific topics of interest for users, and employs specific policies to adjust the difficulty level of tests.

Key–Words: Online testing, E-learning, Adaptive tests, Activity flow modeling, Web-based training

1 Introduction

The development of new technologies is having a tremendous impact over our society in current years. A pivot factor in the society evolution has been the internet, which has become more and more present at our workplace and in our learning methods. The great merit of internet is the easy access to information that has implicitly led to a new, fast and handy range of tools and capabilities for various fields of activity.

Teaching and learning is one of the areas that greatly benefits from the technological explosion. The use of computers and internet influences several components of the educational activity, and introduces a high degree of flexibility with respect to time, place, delivery process and learning process.

The research in e-learning has been developed in two main directions: Computer-based training (CBT) and web-based training (WBT) [2]. With CBT, digital technologies are particularly used, such as CD-ROMS to store and distribute multimedia training materials. WBT facilitates the online training, and uses the internet to provide access to educational materials.

We have directed our research towards web-based training, whose great potential in education comes from its flexibility and continuously increasing accessibility and usability. Training on the internet is becoming more available to every learner at any hour, along with the continuous development of the internet. Moreover, the information can be updated easily, leading to the great popularity of e-learning. In contrast, with CBT, if a read-only device is used (e.g. CD) the difficulty of updating the information imposes a constraint on the educational process. Also, the internet supports the delivery and communication in e-learning. For example, the learning content (e-books, e-courses or e-tests) can be delivered to students through internet and communication can be performed through e-mail, discussion forums, instant messages and so on [2]. In addition, the internet can provide unlimited storage capabilities, due to its distributed nature, while the CBT storage size is limited by the device (e.g. CD, hard disk).

Our work describes an online application for test editing and evaluation. Apart from a detailed structural and functional description, we present our research in two directions that have been addressed with respect to our online testing environment. We discuss the activy flow modeling process that enforces usability and is the basis of a well-structured application. We also introduce the concept of *question-relevant keyword set (QKS)*, which extends testing to flexible, adaptive test design, and facilitates training in specific topics of interest for the user.

The rest of the paper is structured as follows: Section 2 presents related work in this area. Section 3 describes our online testing application from a structural and functional point of view. Section 4 presents the adaptive model used for test design. Finally, Section 5 concludes the paper.

2 Related work

The recent research in online evaluation has ranged from testing techniques based on quizes [3][4] to specific problem solving with creative answers [1]. Grading in online settings has also been studied in the attempt to reduce the faculty time spent on grading [5].

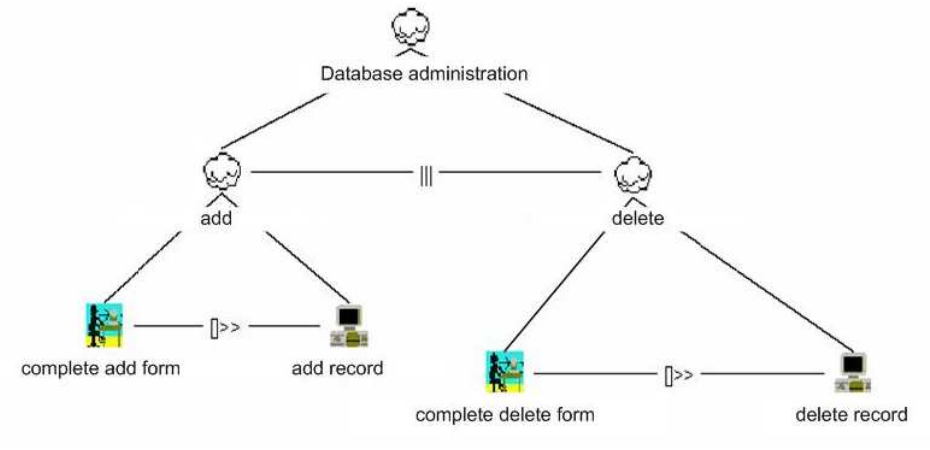

Figure 1: Modeling the activity flow for database administration

Furthermore, several researchers have shared their online teaching practice experience [7], making a strong point with respect to the benefit of this type of training.

Methods of setting up an online learning environment have also been explored [6], along with the need for distributed repositories for large testing databases [8]. Going deeper into technological aspects, educational software design and implementation has been the key concern in [9][10]. Software applications such as QUIZIT [15] and ASSIST [14] exist, which provide online testing facilities for questions whose correct answers respect a specific regular expression.

Moreover, the research area of semantic web has strong ties with e-learning, providing means to personalize the learning process [11][13] and to meet e-leaning requirements [12].

Our work addresses a problem that encompasses both semantic web and online testing techniques. We approach the issue of on-the-fly, adaptive test design, based on representative keywords and knowledge level of trainee.

3 Structural and functional concepts

We are developing the online application as a dynamic internet website for user auto-evaluation. Its first distinguishing feature is usability, enforced by an activity flow modeling process during the design phase that ensures a well structured application and a logical, easy to use interface.

Moreover, an important achievement of our work is the adaptive test design, based on logical decomposition in topics during the testing process. Therefore, the purpose of the evaluation system is not only to assist users in verifying their knowledge online, but also to create a stimulating environment, where users improve their knowledge gradually.

Also, the editing section expands the functionality of traditional testing systems, with the possibility of user interaction in test editing. In this way, the learner can himself provide training material, which becomes available to others who access the online system.

3.1 Modeling the flow of activities

An online testing system is available to people from various categories of work. Most of them do not have a background in computer science, and have a very limited knowledge of information technology. Therefore, an important part of our research has been directed towards usability, which mainly represents the ease of application utilization. In our view, usability can be achieved by having a clear picture of the flow of actions and operations employed by a user who accesses the application. The activity flow modeling improves the description of functions carried out by the application from the point of view of both the user and the designer. It provides the user with a clear and logical way in which the application can be utilized. Furthermore, it helps the designer to better and coherently describe the main tasks carried out by the application he develops.

Building on these considerations, we carried out a thorough study of the activity flow during the design phase. The first step pertains to a logical decomposition in actions that must occur during a user interaction with the online application. The result is a hierarchy of actions, where not all the operations are performed by the user. For example, the insertion of a question in the database is carried out by the application itself, and the user can only ”trigger” the event. Some of the actions in the hierarchy flow are abstract, and may stand for a group of more concrete activities. Basically, actions are assigned to different levels

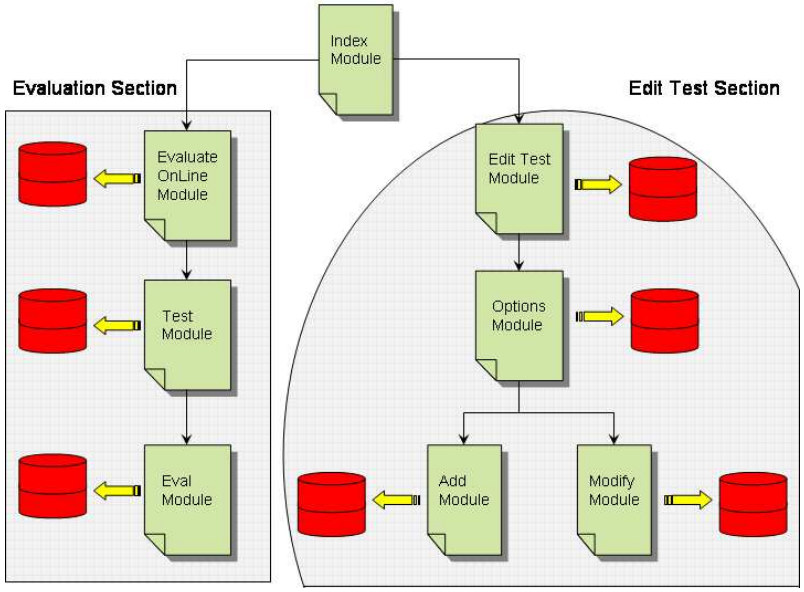

Figure 2: Structural and Functional Schema

and nodes, depending on the abstraction level, starting with the most abstract task at the root and going to more practical and concrete operations at the leaves. Following the logical decomposition in a hierarchy of actions, we identify the temporal relations between operations situated on the same level in the hierarchy. Also, we identify the way in which the actions on the same level are correlated.

Figure 1 models the action flow that occurs in the interaction with the database. We used the notation and representation system provided by the ConcurTaskTrees (CTT) tool [16]. In comparison with previous approaches, such as Hierarchical Task Analysis, ConcurTaskTrees provides a more divers set of notations, with precise meanings [16]. In the database administration modeling, the root node and its children (i.e. the adding and deleting of data) are of abstract type, since they do not include a single concrete action in terms of user-computer interaction. As a general principle, the interaction with a database must allow adding and deleting of records. Between these two types of operations, a temporal order can not be established, since no one of the two operations precedes the other in all scenarios. We use the $\|$ CTT notation to suggest that the two tasks are performed independently in time. The ”complete add form” node models data adding, and reflects a system-user interaction. The ”add record” action is performed by the application itself, during a database access operation. As we emphasized before, the flow model should also incorporate information about actions on the same level in the hierarchy. In the particular case of data adding, we denote a relation of precedence accompanied by delivery of data with the $[] >>$ CTT notation. The data removal is conceptually modeled is the same way.

Similarly, during the design phase we are performing the flow modeling of the activities associated with all structural components. The hierarchical model is particularly useful in achieving a unitary and logical design of the online application as a whole, and defining inter-module interactions and communication, which are described in Section 3.2.

3.2 The modular structure

The online evaluation system has been designed with flexibility and extensibility in mind, in order to provide also interactive editing facilities, in addition to testing and grading. Figure 2 synthesizes the modular structure of the *Evaluation Section* and *Test Editing Section*. Each of the modules interacts with the other components and provides some particular features that are further described.

The main scope of the *Evaluate OnLine Module* is to facilitate the user authentication with a (name, password) pair and also to select the domain of interest. Apart from security reasons, authentication provides a mean to store and retrieve test results and information associated with a user. Thus, adaptive learning process is built on previously stored results (Section 4), and the application adjusts the difficulty of new tests based on past scores. Moreover, the application can provide the user with an evolution diagram constructed from the *user score history*.

The *Test Module* delivers the current test and keeps track of the elapsed testing time. This module

has an important algorithmic and decision component, which will be explained in Section 4, that provides the facility to build adaptive tests. It also interacts with the database to identify the questions. With interactive tests [17], the questions are displayed one at a time, allowing the interference of the user, who can decide to receive feedback on the current question or leave the test at any moment. The category of passive tests [17] has also been addressed in our implementation. With passive tests, all test questions are displayed on a single web page, and the response order is thus more flexible: the user can answer the questions in any order he chooses. The *Eval Module* applies the grading system to evaluate the test answers. Furthermore, it provides a visual feedback with respect to the student evolution, using the user history to deliver an evolution diagram over taken tests.

We have also developed the editing facility, where the access is granted in the *Edit Test Module* by a secure login to the editing section. Our implementation has been advanced in several directions, included in the *Options Module*. The adding of new information to the online evaluation system, such as new domains, topics, sub-topics, new tests or new questions depending on the level of granularity selected by the user is a feature provided by the *Add Module*, while the *Modify Module* is particularly concerned with editing or deleting existing data. Data modification can be done at different levels of granularity, ranging from domains to particular questions, time and score settings. Furthermore, the modules interact with the database in order to retrieve, store or modify information.

3.3 Technological aspects

With interoperability and free access in mind, we have used the PHP and MySQL technologies in out implementation. PHP and MySQL provide the possibility of making a dynamic web portal, and offer equally good implementation facilities as ASP and SQL-Server from Microsoft. In addition, they are open-source and cross-platform, and can be used on a Linux operating system.

4 Adaptive test design

The construction of adaptive tests is an incremental process. Past performances of the same trainee directly impact which questions are extracted from the database to form the current test. In order to adapt the difficulty of the test to the knowledge level of the trainee, we must start with a smaller granularity: the difficulty level of a question.

In general, the difficulty level of questions is not constant for any test. Some of the questions may be easier to answer; others may pose a higher intellectual challenge. For a test question q_i, we associate a difficulty level $d_i \in [1 \cdots R]$, where R is the range, or highest level defined. We have considered that 10 levels of difficulty can usually offer sufficient flexibility in test design, and therefore we are using the value 10 for R.

4.1 Question-relevant keyword set

In this work, we are introducing the concept of *question-relevant keyword set (QKS)*, and describe how this concept can be used to design adaptive tests. The QKS denotes the set of keywords that are associated with a question. In other words, a question tests learner's knowledge from the topics given by the keyword set. We can define the QKS for a question q_i as the set:

$$QKS(q_i) = \{k_1, k_2 \cdots k_n\} \ , \qquad (1)$$

where $k_1, k_2 \cdots k_n$ represent the relevant keywords for question q_i.

Let us consider the following sample question:

q3: Which of the interfaces below does the Hashtable class implement?

- Table;
- List;
- Map.

Although this question might have been added by a user to the programming languages domain, the question can also be relevant for more specific topics, such as Java language or hashtables.

Therefore, the QKS assigned for this question can be:

$$QKS(q_3) = \{programming, Java, Hashtable\} \qquad (2)$$

In this way, we have provided the user with a more powerful evaluation tool. He can now train not only from a general domain (i.e. programming languages), but can target to more specific topics, such as Java language. In addition to already formed tests, the Evaluate OnLine Module is extended with on-the-fly generated tests that incorporate questions from different topics of interest for the learner. The user must provide the domain and the topics, and the online application will select the questions based on the QKS and difficulty level.

When a user edits or adds a test question to the database, the keywords assignment can also be carried out. This is a *static assignment*, since it is performed by the user. We also propose a *dynamic assignment*

No.	Topic	Questions
1	indexing	q1, q2, q4
2	deadlocks	q3, q4, q5
3	queries	q3, q6, q7
4	object-oriented	q6, q7, q8, q9

Table 1: The (topic, question set) association. The learner tests his knowledge on four topics related to databases.

of keywords, carried out by the application, which is based on user interests. Some of the topics might be of greater interest for users than other topics. The frequency of test topic request gives a good measure of users' interest in that particular area. Our solution is to employ a daemon application, that works in the background, or in periods of low activity (i.e. at night), and searches for frequently requested topics in the text of the questions. Next, it updates the QKS for the questions where the frequent topics have been found.

For example, let us assume that several learners are concerned with Java interfaces, and specifically ask for this topic in their test setting requirements. The "interface" keyword does not appear in the QKS for q_3, although this question discusses an aspect related to Java interfaces (i.e. the interface that is implemented by the Hashtable class). Therefore, the application will dynamically assign the "interface" keyword to $QKS(q_3)$:

$$QKS(q_3) = \{programming, Java, Hashtable, interface\} \quad (3)$$

4.2 Integrating difficulty levels with topics

We want to adapt the difficulty test level to the score history of the learner, and to give him the chance to improve in topics where he has more weaknesses. Specifically, if a learner scores low in a particular topic, the next test should have more questions of lower difficulty level in comparison with the other topics that he wants to be included in his test.

Let us assume that a learner wants to master the databases domain. In particular, he wants to test his knowledge in the following databases topics: indexing, deadlocks, queries and object-oriented databases. A simplified description of the questions stored in our evaluation system relating to these topics is presented in Table 1.

If the trainee has not been tested on these topics before, the topics would be represented in the first test in equal proportion (25% each in our case, since we have 4 fields of interest). For the next tests, we enforce an ordering relation such that:

$$\begin{aligned} &If\ n_c^k(t_i) > n_c^k(t_j)\ , \\ &Then\ n^{k+1}(t_i) < n^{k+1}(t_j)\ , \end{aligned} \quad (4)$$

where $n_c^k(t_i)$ represents the number of correct answers on topic t_i in test k and $n^{k+1}(t_i)$ is the number of questions that will appear in test $k+1$ on topic t_i. Therefore, the next tests contain more questions related to topics in which the learner has previously done worse.

Furthermore, we can employ a proportionality relation with respect to the number of questions from each topic that appear in the next test. Mathematically, we can represent the relation for topics i and j as:

$$\frac{n_c^k(t_i)}{n_c^k(t_j)} = \frac{n^{k+1}(t_j)}{n^{k+1}(t_i)}\ . \quad (5)$$

In practice, we prefer to take an average of the number on correct answers on each topic, over a chosen number of previous tests. This is motivated by several subjective factors that may appear during a test and influence the score of the learner. The last performance is usually not the best indicative of the knowledge level on a particular subject.

The difficulty level is adjusted incrementally by referring the learner performance to a threshold T, such that:

$$\begin{aligned} &If\ n_c^k(t_i) \geq T\ , \\ &Then\ Increment(d^{k+1}(t_i))\ , \\ &Else\ Decrement(d^{k+1}(t_i))\ . \end{aligned} \quad (6)$$

where $d^{k+1}(t_i)$ is the difficulty level for questions in topic t_i, for the test $k+1$.

Moreover, we note that, in general, the topics are not disjoint in terms of questions contained. Some questions can be relevant for more than one area; therefore the QKS pertaining to these questions intersect. The algorithm employed in our portal implementation gives more credit to non-common keywords. Let us assume that question q is related to topics t_i and t_j from the current test. If the learner has scored better on topic t_i in the previous test, then q has more chances to be assigned to t_i than to t_j. The reasoning behind this is to allow the user to be trained on more specific questions from the domain in which his knowledge level is lower.

Therefore, if $QKS(t_i) \bigcap QKS(t_j) = \{q\}$, then the probability of assigning q to the topic t_i in the next test $k+1$ is:

$$P^{k+1}(q, t_i) = \frac{n_c^k(t_i)}{n_c^k(t_i) + n_c^k(t_j)}\ , \quad (7)$$

where the ordering relation $n_c^k(t_i) > n_c^k(t_j)$ ensures that q is more likely to be considered in the question-set associated with t_i for the next dynamically created test.

5 Conclusion

Online training has highly benefited from the fast development of the internet in the recent period. The continuously increasing accessibility and flexibility of the internet provides means to develop educational techniques using the web environment.

This work describes an online application for testing, where both the evaluation and test editing facilities are provided to users. We present how activity flow modeling is performed during the design phase, in order to maximize application usability. Moreover, we describe our results in the direction of adaptive tests, where the user score history is a determinant factor for next training tests. We introduce the concept of *question-relevant keyword set (QKS)* to define the topics that a particular question may test. We consider that training is successful if not only the user, but also the system itself learns. Therefore, our method proposes a continuous update of relevant topic-defining keywords, carried out by the application. As future directions of research, we intend to advance our work in the rich area of adaptive and personalized tests, and study complex scenarios of test mapping on user levels of knowledge.

References:

[1] S. Bridgeman, M.T. Goodrich, S.G. Kobourov, and R. Tamassia, PILOT: An Interactive Tool for Learning and Grading, *SIGCSE*, 2000.

[2] P. Wallace, The Internet in the workplace: How new technology is transforming work, *Cambridge: Cambridge University Press*, 2004.

[3] D.M. Woit, D.V. Mason, Enhancing student learning through online quizzes, *SIGCSE*, pp. 367-371, 2000.

[4] D.V. Mason, D.M. Woit, Integrating technology into computer science examinations, *SIGCSE*, pp. 140-144, 1998.

[5] R. Vigilante, Online Computer Scoring of Constructed- Response Questions, *Journal of Information Technology Impact*, 1:2, 57-62, 1999.

[6] M.S. Cohen, T.J Ellis, Validating a criteria set for an online learning environment, *FIE 2004*, 34th Annual Volume, Issue 20-23, pp. 23-7, vol. 2, 2004.

[7] D.M. Woit, D.V. Mason, Effectiveness of online assessment, *SIGCSE*, pp. 137-141, 2003.

[8] C. Schmitz, S. Staab, R. Studer, G. Stumme, J. Tane, Accessing Distributed Learning Repositories through a Courseware Watchdog, *Procs. of the E-Learn 2002* - World Conference on E-Learning in Corporate, Government, Healthcare 4 Higher Education, 2002.

[9] M. Muhlhauser, Multimedia Software for eLearning: An Old Topic Seen in a New Light, *ISMSE'03*, 2003.

[10] J. Roschelle, et al., Developing educational software components, *Computer*, pp. 50-58, 1999.

[11] P. Dolog, N. Henze, W. Nejdl, M. Sintek, Towards the adaptive semantic web, *1st Workshop on Principles and Practice of Semantic Web Reasoning*, 2003.

[12] L. Stojanovic, S. Staab, and R. Studer, eLearning based on the semantic web, *WebNet2001*, World Conference on the WWW and Internet, 2001.

[13] A. Schmidt, C. Winterhalter, User Context Aware Delivery of E-Learning Material: Approach and Architecture, *J. of Universal Computer Science*, vol.10, no.1, January 2004.

[14] D. Jackson, M. Usher, Grading student programs using ASSYST, *Proc. of the 28th SIGCSE*, Technical Symposium on Computer Science Education, pp. 335-339, 1997.

[15] L. Tinoco, E. Fox, D. Barnette, Online Evaluation in WWW-based Courseware, *Proc. of the 28th SIGCSE*, Technical Symposium on Computer Science Education, pp. 194-198, 1997.

[16] G. Mori, F. Patern, C. Santoro, CTTE: Support for Developing and Analysing Task Models for Interactive System Design, *IEEE Transactions on Software Engineering*, pp. 797-813, Vol. 28, No. 8, IEEE Press, 2002.

[17] A. Arora, E. Barker, U.P. Karadkar, P. Dave, L. Francisco-Revilla, R. Furuta, F. Shipman, S. Dash, Z. Dalal, The Walden's Paths Quiz Engine, *E-Learn 2003*, AACE pp. 2052-2059, 2003.