An Online System for Testing and Evaluation

CRISTEA BOBOILA[1], GEORGE V. IORDACHE[2], MARCELA S. BOBOILA[3]
[1]Faculty of Mathematics and Computer Science
University of Craiova
Al.I. Cuza Street, No. 13, Craiova RO-200585
ROMANIA
boboila@central.ucv.ro http://inf.ucv.ro/~boboila
[2]Computer Science Department
Stony Brook University
Stony Brook, NY 11794-4400
U.S.A.
georgei@cs.sunysb.edu
[3]Computer Science Department
Stony Brook University
Stony Brook, NY 11794-4400
U.S.A.
mboboila@cs.sunysb.edu

Abstract: - This paper describes an online application for test design and evaluation of trainees. We address two main topics: 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, targeted to specific topics of interest for users, and employs specific policies to adjust the difficulty level of tests. We also describe a taxonomy-based test design method, that enhances the testing environment with new usability and efficiency attributes.

Key-Words: - Online testing, E-learning, Computer adaptive tests, Activity flow modeling, Web-based training, Taxonomy

1 Introduction

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

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

Research in e-learning is comprised of two main directions: computer-based training (CBT) and web-based training (WBT) [25]. With CBT, digital technologies are used in particular to store and distribute multimedia training materials (e.g. CD-ROMS). WBT facilitates online training, and uses the internet to provide access to educational materials.

We have directed our research towards web-based training, which has great potential in education due to its flexibility and continuously increasing accessibility and usability. Training on the internet is becoming available to a higher number of learners nowadays, owing to continuous development of the internet. 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. Furthermore, 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 messaging and so on [25].

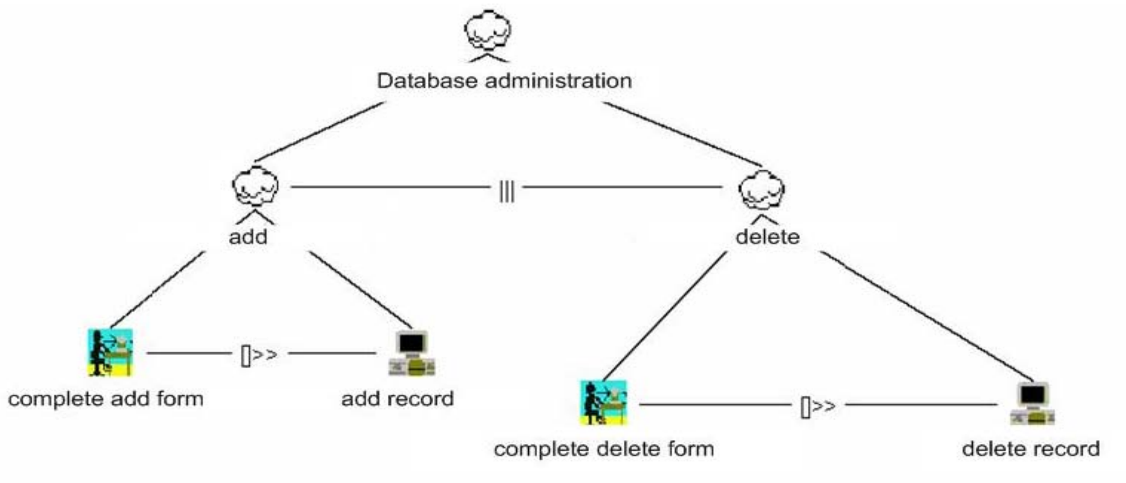

Fig.1. Modeling of the activity flow for database administration

In addition, the internet can provide close to unlimited storage capabilities, due to its distributed nature, while the CBT storage size is limited by 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 active flow modeling process that enforces usability and is the basis of well-structured applications. 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 the above 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. Section 5 discusses a taxonomy-based model for test design, where the online system uses a structural description of the testing domains. Finally, Section 6 concludes the paper.

2 Related work

Recent research in online evaluation has ranged from testing techniques based on quizzes [12, 28] to specific problem solving with creative answers [2]. Grading in online settings has also been studied in the attempt to reduce time spent on grading by faculty [24]. Furthermore, several researchers have shared their online teaching practice experience [27], 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 [3], along with the need for distributed repositories for large testing databases [18]. Going deeper into technological aspects, educational software design and implementation has been the key concern in [11, 14, 16]. Software applications such as QUIZIT [21] and ASSIST [10] exist and 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 [5, 6, 19, 23, 26] and to meet e-learning requirements [20]. Computer-adaptive test design has been largely explored in recent years, proving to be a very efficient modality for testing the knowledge level of users [4, 7, 8, 9, 15, 17, 22].

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.

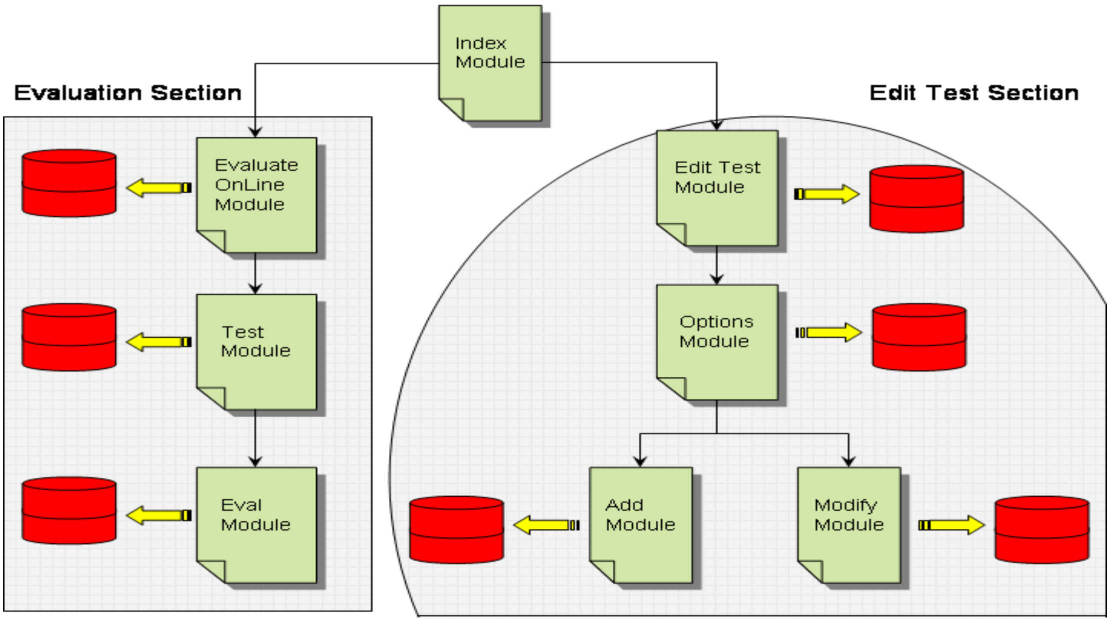

Fig.2. Structural and Functional Diagram

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.

Furthermore, the editing section expands the functionality of traditional testing systems, with the possibility of user participation 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 of various professions. 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 the 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 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 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 [13]. In comparison with previous approaches, such as Hierarchical Task Analysis, ConcurTaskTrees provides a more diverse set of notations, with more accurate meanings [13].

In the database administration modeling process, 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 neither 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 in 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 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 particular features that are described below.

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. In addition to increased security, authentication provides the means to store and retrieve test results and information associated with a user. Thus, the 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, (explained in Section 4) that provides the ability to build adaptive tests. The Test Module also interacts with the database to identify the questions.

With interactive tests [1], 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 [1] 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 test answers. Furthermore, it provides visual feedback with respect to the student evolution, and uses the stored 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 features several editing possibilities, included in the Options Module. The addition of new information to the online evaluation system is a feature provided by the *Add Module*, while the *Modify Module* is particularly concerned with editing or deleting existing data. Such information includes new domains, topics, subtopics, new tests or questions that depend on the level of granularity selected by the user. 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 our implementation. PHP and MySQL provide the possibility of making a dynamic web portal, and offer equally good implementation facilities to ASP and SQL-Server from Microsoft. In addition, they are open-source and cross-platform, and can be used on both Windows and Linux operating systems.

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 $d_i \in [1 \ldots R]$, 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:

$$QKS\ (q_i) = \{k_1, k_2 \ldots k_n\}, \quad (1)$$

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

Let us consider the following sample question:
q_3: *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 the Java language or hashtables.

Therefore, the QKS assigned for this question can be:

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

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 also from 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 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 performed. This is a *static assignment*, since it is performed by the user. We also propose a *dynamic assignment* of keywords, carried out by the application, which is based on user interests. Some topics might be of greater interest for users than other topics. The frequency of the test topic requested gives a good measure of users' interest in that particular area. Our solution is to use a daemon process (application) that works in the background, or in periods of low activity, 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.

In order to maintain consistency and prevent the user from requesting inexistent topics, our testing system stores an internal database with possible keywords. The keywords database has been populated through an English dictionary lookup and consists of nouns only, ignoring prepositions, conjunctions, etc.

To illustrate the background daemon's functioning, we will 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 dynamically establish the difficulty level of the next test to take into account the score history of the learner, and at the same time to give the learner the chance to improve in topics where he has more weaknesses. Specifically, if a trainee 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 database 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. The learner tests his knowledge on four topics related to databases.

If the trainee has not been tested in these domains 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).

Number	Topic	Questions
1	indexing	q_1, q_2, q_4
2	deadlocks	q_3, q_4, q_5
3	queries	q_3, q_6, q_7
4	object-oriented	q_6, q_7, q_8, q_9

Table 1. The (topic, question set) association.

For the next tests, we enforce an ordering relation such that:

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

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 scored less.

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:

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

In practice, we prefer to take an average on the number of 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 comparing the learner's performance to a threshold T, such that:

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

$d^{k+1}(t_i)$ 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 keyword sets pertaining to these questions intersect. The algorithm employed in our portal implementation gives more credit to more specific 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 $QKS\ (q) = \{t_i,\ t_j\}$, then the probability of assigning q to the topic t_i in the next test $k+1$ is:

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

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 Taxonomy-based test design

We have enhanced the adaptive test design described in Section 4 with structure, in order to increase usability. The system learns a taxonomy that describes the relations among topics and subtopics, as class / subclass interdependencies. A node in the taxonomy denotes a topic (class). The child nodes are subtopics (subclasses). Figure 3 presents the taxonomy for the "database" concept.

5.1 Score statistics

There are several advantages to our approach. First, the e-learning system can compute a series of aggregates illustrating score statistics for the user. For example, when reporting scores, the system tells precisely the sub-domains where the user has scored the lowest and the highest. The statistical information does not only depict the subtopics chosen by the user at the test design, but can also refer to a smaller granularity, based on the taxonomy.

If a trainee tests his knowledge in relational databases (Figure 3), the Eval Module shows the percent of correct answers in any subtopic in the taxonomy, by accessing the corresponding node. Also, the Eval Module presents the highest score and corresponding domain (e.g. deadlocks) and the lowest score and associated domain (e.g. unclustered indexing).

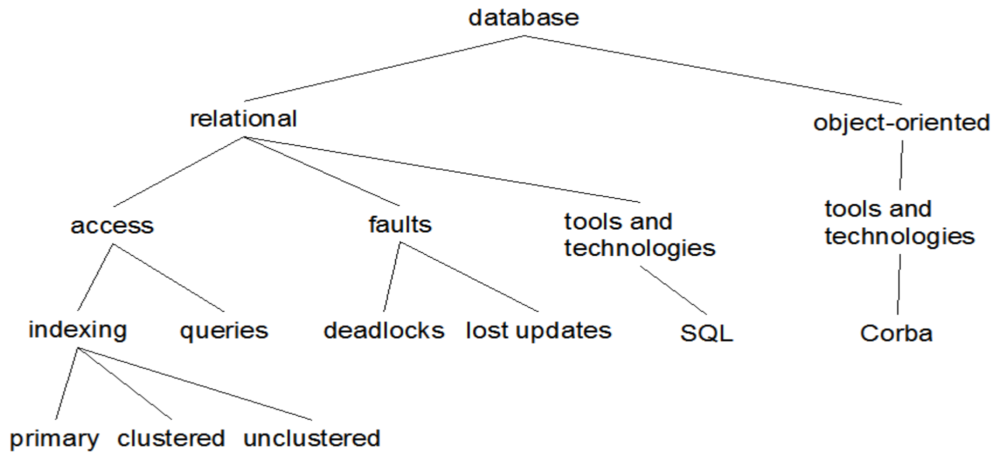

Fig.3. Taxonomy-based test design

For the highest and the lowest score, we only look at the leaves in the taxonomy tree, not at the inner nodes, since an inner node shows a total score over all its children.

Another major advantage of using a taxonomy to describe the structure of the test domains is the common comparison basis that the taxonomy represents when evaluating several students. If the students take the same test, the Eval Module correlates and compares their scores in subtopics, and presents per-subtopic statistics regarding the students' overall results (average scores, student with highest/lowest score).

Based on statistics, the educator who designs the test (the user or the administrator who added the questions to the system's online database) may decide to increase training in one domain or another, to diversify the questions or adjust (re-scale) the difficulty level or the score of the questions.

Besides the informative value for the user (trainee and trainer), the statistics also have a functional value for the learning system. The system uses the score aggregates in the adaptive test design process, to compute the proportion in which the subtopics are represented in future tests.

5.2 Learning patterns

The system learns patterns that are used to minimize user effort and maximize efficiency of the learning process. For example, the system learns similarity patterns among testing topics chosen by users. Based on these patterns, the online testing system suggests other testing areas that might be of interest for him. If the student chooses to test his knowledge in the topic 'X', the system also suggests topic 'Y' for testing. The pattern applied here consists in finding topics chosen by other users with similar interests (users who selected `X' also chose `Y').

5.3 Insertion and pruning

In a static taxonomy model, the tree is given in the beginning, and is immutable (it does not change). We have directed our research towards a dynamic taxonomy model, which can be updated by insertion and pruning. The administrator can insert new topics in the tree, and delete some topics. At the same time, we have integrated the taxonomy model with the *QKS* update operation presented in Section 4.1, where the topics requested by the user are added by a background daemon during periods of low activity. In order to do the insertion automatically (by the system), we keep the *QKS* as an ordered set, based on the parent-child relation. Let us take for example the question *q* and its corresponding *QKS*:

$$QKS\ (q) = \{databases,\ relational\ databases,\ database\ access\}. \tag{8}$$

We can see that: 'database access' < 'relational databases' < 'databases', where the operator '<' denotes the relation 'is child of'.

Let us suppose that a user has requested the topic 'indexing', which is not present in any *QKS* and neither in the taxonomy. The background daemon first finds all *QKS* to which the topic should be added (Section 4.1). Next, the daemon looks at the QKS where it has just added the topic (i.e. indexing), retrieves the topic registered in the *QKS* on the previous position (i.e. database access) and inserts the node called 'indexing' under the node 'database access' in the taxonomy. The method described works well when the topics that are inserted conceptually correspond to new leaves in the taxonomy tree. We are currently developing an algorithm for inserting inner nodes as well, a process which poses a higher challenge since we need to keep the taxonomy logically consistent.

The pruning operation is automatically executed if a particular topic has not been requested for a time period. We are using a 3-month time threshold for our system.

For leaf nodes, we simply delete the node from the taxonomy, and remove the topic from all the keyword sets in which it appears. For inner nodes, we only remove the topic in case neither of the subtopics corresponding to the child nodes on all the paths down to the leaves has been used during the last 3 months. The advantage of pruning (removing the unused nodes) is an improvement of the search process and computation speed increase.

6 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 describe how activity flow modeling is performed during the design phase, in order to maximize application usability. Moreover, we introduce our results in regards to adaptive tests, where score history is a determining factor for the design of the next training tests.

We introduce the concept of *question-relevant keyword set (QKS)* to define the topics that a particular question may test. We believe 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. We also describe a taxonomy-based test design method, where the online evaluation system has full knowledge about the topics that may appear during the testing process and the interdependencies among them, as class/subclass relations. We present the advantages offered by this approach, in regards to computations of score statistics, system updates and pattern learning.

As future directions of research, we intend to advance our work in the area of adaptive and personalized tests, and study more complex scenarios of test-mapping on user levels of knowledge.

References:

[1] 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, *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (E-Learn)*, 2003, pp. 2052-2059.

[2] S. Bridgeman, M.T. Goodrich, S.G. Kobourov, and R. Tamassia, PILOT: An Interactive Tool for Learning and Grading, *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE)*, 2000.

[3] M.S. Cohen, T.J Ellis, Validating a criteria set for an online learning environment, *Proceedings of the Frontiers in Education Conference (FIE)*, 34th Annual Volume, Issue 20-23, Vol. 2, 2004, pp. 23-7.

[4] M. De Beer, Use of CAT in dynamic testing, In D. J. Weiss (Ed.), *Proceedings of the GMAC Conference on Computerized Adaptive Testing, 2007.*

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

[6] A.A. Economides. Personalized feedback in CAT, *WSEAS Transactions on Advances in Engineering Education*, Issue 3, Vol. 2, 2005, pp. 174-181.

[7] S. Fu, M. Desmarais, Multidimensional computerized adaptive testing based on Bayesian theory, *Proceedings of the Education and Technology Conference*, Calgary, 2006.

[8] E. Georgiadou, E. Triantafillou, and A. Economides. A review of item exposure control strategies for Computerized Adaptive Testing developed from 1983 to 2005, *Journal of Technology, Learning, and Assessment*, Vol.5 No. 8, 2007.

[9] E. Georgiadou, E. Triantafillou, and A. Economides. Evaluation Parameters for Computer Adaptive Testing, *British Journal of Educational Technology*, Vol. 37, Issue 2, 2006, pp.261-278.

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

[11] M. Lilley, T. Barker, C. Britton. The development and evaluation of a software

prototype for computer-adaptive testing, *Computers & Education*, Vol. 43, 2004, pp. 109–123.

[12] D.V. Mason, D.M. Woit, Integrating technology into computer science examinations *Proceedings of the 29th Technical Symposium on Computer Science Education* (*SIGCSE*), 1998, pp. 140-144.

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

[14] M. Mühlhäuser, Multimedia Software for eLearning: An Old Topic Seen in a New Light, *Proceedings of the IEEE Fifth International Symposium on Multimedia Software Engineering* (*ISMSE)*, 2003, pp. 2-4.

[15] G. Raîche, J.G. Blais, D. Magis, Adaptive estimators of trait level in adaptive testing: Some proposals, In D. J. Weiss (Ed.), *Proceedings of the 2007 GMAC Conference on Computerized Adaptive Testing*, 2007.

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

[17] L.M. Rudner, An on-line interactive computer adaptive testing tutorial, Online resource: http://edres.org/scripts/cat/catdemo.htm, Accessed: March 1st, 2008.

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

[19] A. Schmidt, C. Winterhalter, User Context Aware Delivery of E-Learning Material: Approach and Architecture, *Journal of Universal Computer Science*, Vol. 10, No. 1, January 2004.

[20] L. Stojanovic, S. Staab, and R. Studer, eLearning based on the semantic web, *Proceedings of the World Conference on the WWW and Internet (WebNet)*, 2001.

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

[22] E. Triantafillou, E. Georgiadou, A. Economides, Applying adaptive variables in computerised adaptive testing, *Australasian Journal of Educational Technology*, 23(3), 2007, pp. 350-370.

[23] E. Triantafillou, E. Georgiadou, A. Economides, The Role or User Model in CAT: Exploring Adaptive Variables, *Journal of Technology, Instruction, Cognition and Learning*, Vol. 5, 2007, pp. 69-89.

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

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

[26] D.M. Woit, K.J. Bell, Student communication challenges in distributed software engineering environments, *Proceedings of the Integrating Technology into Computer Science Education (ITiCSE)*, 2005, pp. 286-290

[27] D.M. Woit, D.V. Mason, Effectiveness of online assessment, *Proceedings of the 34th Technical Symposium on Computer Science Education (SIGCSE)*, 2003, pp. 137-141.

[28] D.M. Woit, D.V. Mason, Enhancing student learning through online quizzes, *Proceedings of the 31st Technical Symposium on Computer Science Education (SIGCSE)*, 2000, pp. 367-371.