NewOX - extending the Online eXam system to provide automated formative evaluation of student text-based assignments

Ralf Keidel[1], Heinz Dreher[2], Eric Sean Conner[1], Marc Willhelm Küster[1]

[1]University of Applied Sciences Worms, ZTT, [2]Curtin University of Technology, Perth, Western Australia

Key words: *e-Assessment, Multiple-Choice, Summative Assessment, Formative Assessment, Short Essay Assessment, Case Study*

Abstract:

A cheat-resistant and foolproof distributed electronic application for summative assessment of student learning outcomes has been developed and field tested "Online eXam"(OX). Good usability experience and feedback features for the students as well as the lecturers were design goals of the implementation. The field tests showed the robustness of the system in a heterogenic PC pool environment, the ease of use and a good student learning outcome assessment ability (summative assessment) but a lack of a formative impact on the student learning efforts. The incorporation and adoption (for European needs) of a short essay assessment algorithm with a rich user interface is proposed.

1 Introduction and rationale

The Center of Technology Transfer and Telecommunications at the University of Applied Sciences Worms, Germany, has developed a set of distributed rich client JAVA applications to provide a computer aided summative assessment environment for set time IT lab examinations. The main concerns tackled were

- closely coupled client and server applications for maximum control during examinations,
- full control of the client desktop to suppress any electronic cheating,
- a print out of the individual student exams for signing to ensure legal validity,
- short turn around times for correction and publication of the results,
- ease of use,
- ease of deployment and of course
- stability in terms of network and power failures.

This system has been used and constantly improved in a variety of occasions in the fields of computer sciences and business administration. This system is called "Online eXam" (OX). Development of OX was funded by the Innovations Fund of the „Ministerium für Bildung, Wissenschaft, Jugend und Kultur" of Rheinland-Pfalz, Germany.

The motivation for this development was to provide early feedback on the learning outcomes of students especially during the beginning of their studies and give support and guidance for their learning efforts with very limited manpower but good hardware infrastructure.

2 Methods

2.1 Workflow

With the OX system summative student learning outcome assessment begins with the writing of a stem, which describes a problem situation and several alternatives for a possible solution [2]. This can be a question with at least one correct answer and several incorrect answers (distracters) that seem reasonable for the unknowing exam participant. This problem (question with correct and incorrect answers) is part of an exam. The authoring of problems is supported by a rich client application, which supports cut, copy and paste with other programs running on the lecturer's hardware and organizes problems in a customizable hierarchical repository. Creating an exam is a matter of dragging and dropping problems from the repository to an exam representation. An exam representation is stored in a single XML [1] file, which must be deployed to the OX server (see Figure 1, section 1) prior to conducting the exam.

The OX server software runs continuously on dedicated hardware in a secure campus data centre. It hosts all exams that are ready to be conducted, manages all necessary communication between lecturer and participants during the exam and handles retrieval and export of results. The lecturer communicates with the OX server through the OX pilot application that is typically installed on the lecturer's laptop. The OX pilot application initialises the exam by deploying the exam file, generating a password for the exam participants and monitoring the logon procedures of the participants. During the logon procedure the participant is required to provide all examination relevant details, such as name, unique student ID and email address. The successful logon procedure ends on a waiting screen for the participant and as a list entry in the OX pilot application for the lecturer.

The OX exam client application immediately persists every action of the exam participant. The OX pilot, by means of the OX server, controls the client application. The OX exam client can be started, paused and stopped on behalf of the OX exam pilot. The exam client displays one multiple-choice problem with the possibility to switch freely from problem to problem and check or uncheck appropriate answers. After stopping the exam procedure the results are collected and merged in one XML file. This file is exported to the OX pilot system for the subsequent grading procedure. As an optional final step it is possible to generate a printout of each participant's exam. This printout can be compared to a read-only representation of the exam in the OX exam client application and, if necessary for legal reasons, signed by the participant.

The OX grading application is a rich client application supporting

- Statistics in grading distribution
- Normalized number of clicks per answer
- Normalized difficulty level of the problems
- Interactive list of exam participants with exam view
- Text and HTML format for notice board and online result posting
- Printout with mark, grading statistics and revised exam per exam participant
- Possibilities to change the evaluation schemes
- Possibility to change parameters for the grading procedure

All parameter changes are stored immediately in the XML exam file. This can then be marked unchangeable for the grading application to prevent further manipulations. This file should be kept in a safe place for later references.

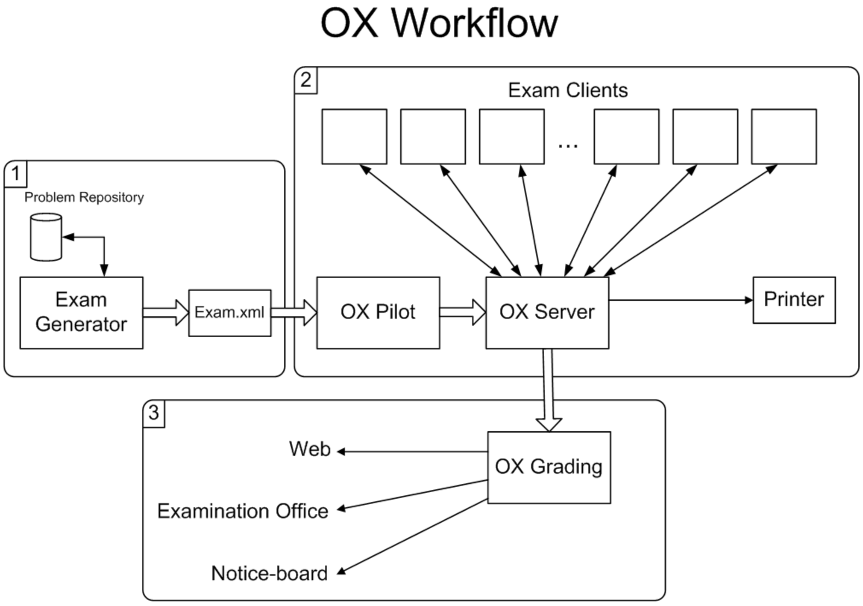

Figure 1: Workflow of the OX system.
Subsection 1: preparation of the exam.
Subsection 2: conducting the exam.
Subsection 3: grading the exam.

2.2 Security

The exam participants interact with the OX system through a kiosk style application, that takes control of the entire user interface of the hosting system on launch. This effectively suppresses the use of resources and software not explicitly permitted for the exam. Every exam consists of an arbitrary number of questions with correct answers und incorrect answers - the so-called problem. The order of the answers as well as the order of the problems is individually shuffled by means of random number generators for each exam participant. This effectively prevents fraud even if the displays are in close proximity.

The OX server runs in a stable operating system environment on dedicated hardware supported by an uninterruptible power supply. Every change in the state of the system is immediately persisted. In the unlikely case of a power failure data loss is kept to minimum and the system automatically restarts to the last valid state before the power failure.

In case of a client failure such as accidental shutdown or network interrupt a re-connect to the running system while preserving the client’s state is easily managed.

3 Results and Discussion

3.1 Use cases

The OX system has been in productive use for examinations in computer sciences and business administration since 2005. It has been employed in ten courses in computer sciences and three in business administration. The overall number of examinations in different topics is 47 with 1544 participants. The number of participants per examination varies from 4 to 66. The number of problems per exam varied from 10 to 50. The typical duration of an examination is one and a half hour.

The system worked reliably without any failures. Problems encountered were largely due to network misconfiguration:

- blocking of network traffic to the OX server by miss administrated firewall settings or routers,
- printer out of order or blocked by firewall settings.

Both problem types were detectable before starting the examination procedure and appropriate counter measures could be taken.

3.2 Time and effort considerations

The time used to build an exam from scratch naturally depends on the subject and the lecturer him- or herself. The authors observed the following approximate times:

Table 1: Average time involved for the creation of a typical exam item in three different categories.

Type of MC item [2]	**Target**	**Average time involved to build an item**
Knowledge	Measure learned by heart	1 minute
Comprehension	Measure lowest level of understanding	10 minutes
Application	Measure understanding	30 - 60 minutes

The examination procedure itself has an overhead of several minutes for the logon procedure of the participants. Depending on the number of the participants, the amount and size of problems and the performance of the printer up to 15 minutes for 50 exams have to be added, but only if a paper based version for legal issues is mandatory.

The grading is done with the default parameters within a few seconds. Posting of the results is subsequently done by means of web pages and printouts.

The moderate effort to build, conduct, grade and post the results of an exam lead to the idea of weekly short summative assessment of the learning outcome during the semester instead of performing only one summative assessment at the end of the semester. The assumption was to have a formative impact on the student learning outcome due to the short turn around time between the accomplishment of the exam and the posting of the grading.

3.3 Frequent summative assessment of student learning outcomes

To review the hypothesis of the formative impact of consecutive exams a row of six short exams in six weeks were performed. The exams began in November 2007 and ended in December 2007. The topics were fundamentals of information technology and computer science such as UNIX commands, number systems, regular expressions and object-oriented systems. The first two exams began with knowledge items, followed by three exams with comprehension items and ended with one exam with application items.

The average of the marks as well as the standard-deviation of the sixteen successful participants is shown in Figure 2. The average summative student assessment outcome shows no tendency to better results of the cohort.

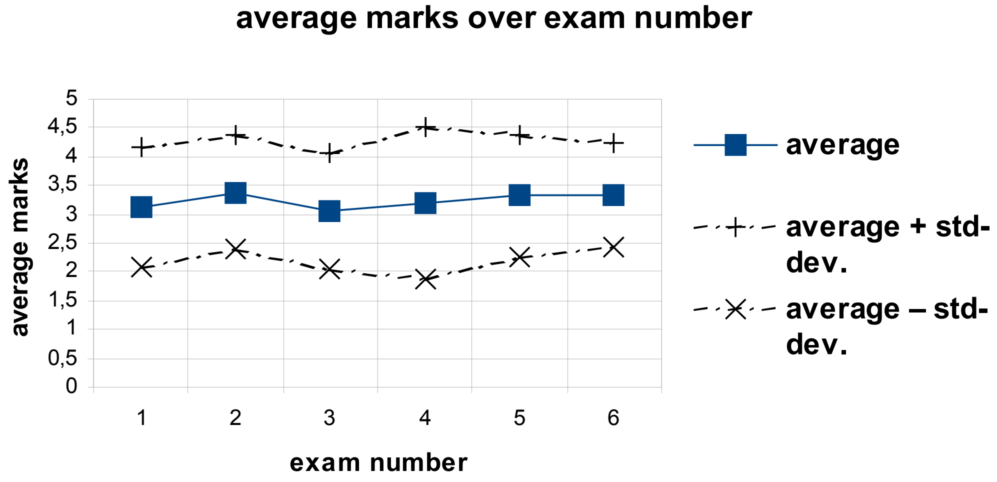

Figure 2: Six exams were conducted within six weeks from11/14/07 through 12/19/07. The average marks with upper and lower standard-deviation boundaries of the successful students (16) are shown. No trend to a better student learning outcome can be seen. A formative impact on the student learning outcome was not observable. 2,7 3,0 3,3 are equivalent to "C" in the ECTS grading scale [3].

The individual performance changes are visualised by applying linear regression lines on the individual grades over time. The slope of these regression lines gives a hint to the tendency of the learning outcome assessment of the individual participant. Figure 3 visualises the value of the slope for the candidates. Positive values indicate the tendency to reduced learning outcomes and negative values indicate the tendency to improvements in the learning outcomes. No clear tendencies are observable (see Figure 3).

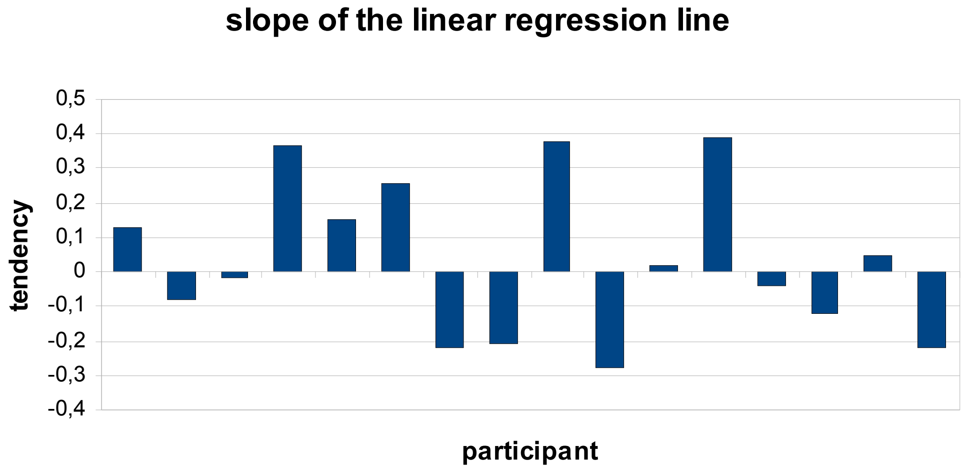

Figure 3: Slope of the linear regression line of the assessed learning outcome of the individual exam participant. Positive values indicate a trend to reduced summative assessment outcome, negative values vice versa.

3.4 Summary of the first part

The OX system, based on the multiple-choice technique for the summative assessment of student learning outcome, works reliably with low time and effort for the lecturer and many benefits for the exam participants. A formative impact on the student learning outcome could not be observed. A further improvement by adapting formative assessment algorithms such as MarkIT [11] with rich client interfaces for lecturer as well as exam participants for short essay items is proposed.

4 Formative assessment of student learning

Whilst checking student progress via summative assessment is in many cases the accepted practice, particularly where large numbers of students are involved, good teachers know that deep learning [6] cannot readily be assessed by objective testing – T/F or multiple-choice tests. Deep conceptual learning is usually associated with formative evaluation where students are required to reflect on a lesson (some learning experience) and in so doing are able to achieve deeper insight into the issue or problem under consideration.

To ascertain the achievement of deep learning one needs evidence of synthesis in addition to recognition, and this is best checked by evaluating the written or spoken reflections of learners. Prima facie, deep learning does not lend itself to automated assessment as readily as surface learning, and yet there is an urgent need to accomplish this in cost-effective and efficient ways. Yasuko [12] has recognised this need to evaluate deep learning and has proposed an analysis of student assignment material to determine the connection between the "transition of the level of comprehension" and deep conceptual learning. Recent advances in essay grading or essay scoring technology [9,11]) have permitted educators to seriously contemplate embedding automated assessment of written material into their courses and Learning Management systems. For example, D'Mello et al. [4] at the University of Memphis have developed "a fully automated computer tutor that simulates human tutors and holds conversations with students in natural language" named AutoTutor. Dreher, Scerbakov, and Helic [5] used the WBT-Master Learning Management system which was endowed with assignment assessment and feedback support functions, but this still required the evaluator to read through hundreds of assignments. Both these systems make significant efficiency gains over pure manual grading systems but can be further improved when it comes to interactive feedback and formative evaluation.

Heinrich and Lawn [8] had the goal of automating formative assessment, although they were only able to realize assistance in document management, and very importantly some provision for feedback annotations and marks in their MarkTool system. Guetl, Dreher and Williams [7] proposed a system called E-TESTER to handle the auto-generated questions and automatic answer assessments and are currently working on further developments to progress this idea.

The essential requirement for automated dynamic formative evaluation, apart from reliability and validity of course, is computational efficiency – that is speed. Computational linguistics requires substantial computer resources and unless the answer-text can be processed within fractions of a second (perhaps up to 1 or 2 seconds at most) such algorithms cannot be used in a Learning Management environment to assess student learning on-the-fly. The algorithm used in the MarkIT automated essay grading system [11], named Normalised Word Vector or NWV [10] is able to perform its computations within such timeframes, and therefore lends itself to the proposed use – that is to provide dynamic formative evaluation and feedback of students written reflections for the purpose of ascertaining and promoting deep learning.

NWV is a computationally efficient algorithm for parsing text and representing document semantics using vector algebra techniques. The vector representation is based on a *normalised* subset of words in a thesaurus thereby reducing words to matching *root words* [10].

5 Formative Assessment of Learning in OX

Given the modularity of OX's overall distributed architecture, it would make OX a logical platform for hosting a component for the automated assessment of formative learning. This way, this component can benefit from many of OX's general features such as authentication, tested UI, resilience to error, and legal security. In fact, it would be straightforward to design exams that profit from both the summative and the formative learning modules, e.g. by checking some basic factual knowledge prior to assessing deep learning.

The authors plan to integrate automated assessment of formative learning in OX. In a first step, the system is expected to concentrate on exams in English and German, though other languages may be covered at a later stage. The remainder of this section covers the current plans for this work:

5.1 A New OX

For the student's perspective much of OX's interface would remain unchanged. Probably the most noteworthy addition to their view of the UI would be the addition of text boxes for free text entry.

From the examiner's point of view, however, the system would show a quite different face. In particular, it would need tools to:

1. Create exams that can also contain free text questions
2. Manually grade exams to create a sufficiently large sample to extract a NWV to grade this exam (cf. [6], [11])
3. Extract the NWV and visualize its parameters
4. Automatically grade the exam based on the previously established NWV
5. Visualize significant deviations between manually and automatically assigned grades

Whereas tools 1, 2 and 5 are rather straightforward to develop based on the OX set of libraries, tools 3 and 4 require either the integration of the current MarkIT algorithm or to re-implement it anew(cf. above).

5.2 Multilinguality

At present, MarkIT works only for English-language texts. It leverages the Australian Macquarie thesaurus to map all English words to some 800 different concepts. It then bases its semantic analysis on these concepts.

As present it is not clear how well this choice works for very domain-specific texts with their plethora of special terms with highly specific meanings, as the Macquarie thesaurus does not cover such terms. The concrete choice of thesaurus will certainly not work for other natural languages such as German where we need language specific thesauri. Specialised thesauri can be sourced (preferable) or built, perhaps as domain specific ontologies.

We theorize that we shall need to develop tools for the

1. Automatic extraction of thesauri (clustering of related terms, potentially based on domain-specific texts)
2. Manual correction and creation of automatically extracted thesauri

Licensing commercial thesauri for a given language will complement the automatically extracted thesauri.

6 Summary

Our aim is to provide support for the onerous assessment task. On the one hand one wants to provide timely feedback to students to promote further learning- accurate, reliable, informative. On the other, one must be efficient. The automated, or semi-automated creation of assessments and subsequent feedback to students must be one possible path to achieve these goals.

The existing OX system provides many of the features needed for accurate and reliable assessments, and can be further developed and built upon to integrate recent developments in Automated Essay Grading and conceptual analysis, thereby extending the life of OX via its new incarnation NewOX.

Acknowledgement

We wish to thank Andreas Orth for his support with software development, bug fixing and conducting of exams.

References:

[1] http://www.w3.org/XML/

[2] Gronlund, Norma E. (2003), Assessment of Student Achievement, Boston: Prentice Hall

[3] http://ec.europa.eu/education/programmes/socrates/ects/index_en.html

[4] D'Mello, S., Craig, S., Witherspoon, A., Sullins, J., McDaniel, B., Gholson, B., & Graesser, A. (2005) The Relationship between Affective States and Dialog Patterns during Interactions with AutoTutor. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005* (pp. 2004-2011). Norfolk, VA: AACE. http://www.editlib.org/index.cfm

[5] Dreher, H., Scerbakov, N., & Helic, D. (2004). Thematic Driven Learning. *Proceedings of E-Learn 2004 Conference*, pp2594-2600, AACE. Washington DC, USA, November 1-5, 2004.

[6] Dreher, H. (2006) Interactive On-line Formative Evaluation of Student Assignments. *Journal of Issues in Informing Science & Information Technology*, Vol 3, 2006. http://informingscience.org/proceedings/InSITE2006/IISITDreh235.pdf

[7] Guetl, C., Dreher, H. & Williams, R. (2005) E-TESTER: a Computer-based Tool for Auto-generated Question and Answer Assessment. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005* (pp. 2929-2936)., October. E-Learn 2005 http://www.editlib.org/index.cfm

[8] Heinrich, E., & Lawn, A. (2004). Onscreen Marking Support for Formative Assessment. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004* (pp. 1985-1992). Norfolk, VA: AACE. http://www.editlib.org/index.cfm

[9] Williams, R. (2001) Automated Essay Grading: An Evaluation of Four Conceptual Models in Kulski, M. & Herrmann, A.(editors) (2001), *New Horizons in University Teaching and Learning: Responding to Change*, Curtin University of Technology, Perth, Australia. http://lsn.curtin.edu.au/tlf/tlf2001/williams.html

[10] Williams, R. (2006) The Power of Normalised Word Vectors for Automatically Grading Essays. Paper to be presented at InSITE 2006, Manchester, England, June 25-28. http://2006.informingscience.org

[11] Williams, R. & Dreher, H. (2004) Automatically Grading Essays with Markit©. *Issues in Informing Science and Information Technology,* Vol. 1, pp693-700. http://articles.iisit.org/092willi.pdf

[12] Yasuko, N. (2004) Innovation of the Formative Assessments Approach using WBT. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004* (pp. 453-460). Norfolk, VA: AACE. http://www.editlib.org/index.cfm