

STRUCTURING AN ON-LINE ASSESSMENT OF STUDENTS' LEARNING

Sharon Curtis, Mary Zajicek

*Department of Computing, Oxford Brookes University,
Wheatley, Oxfordshire, UK.
sharoncurtis@brookes.ac.uk, mzajicek@brookes.ac.uk*

ABSTRACT

This paper describes an on-line assessment system (HCIQ) for HCI. The assessment questions were structured to assess understanding of HCI concepts in a way that allows for automated marking. The HCIQ system was designed so that for a large number of students, the on-line test could take place in multiple short sittings in computer labs, and in such a way that cheating was prevented.

KEYWORDS

On-line assessment, human-computer interaction, plagiarism prevention

1. INTRODUCTION

This paper describes an on-line assessment procedure for the subject Human-Computer Interaction (HCI), a computing topic that focuses on interaction and interfaces between humans and computers.

The assessment system described in this paper was designed for a second year undergraduate level course at a UK university, that involved introducing students to user-centred design. The testing of students' practical skills is carried out by group coursework, with groups of four to five students building an interface to a system. This is balanced with assessment of their understanding of HCI principles and concepts. In the past, this was carried out by means of a written exam, and students were able to show their understanding by explaining concepts and illustrating points with examples. New arrangements meant that a traditional written examination was no longer possible, and the assessment of students' understanding of HCI principles is now carried out by means of an on-line test. This paper describes the structure of the on-line test which was devised to test students' real understanding of the course material and prevent collusion or plagiarism when students are using computers in close proximity to one another.

Experience with this on-line assessment system (HCIQ) has shown that as well as the expected advantages of automated marking and reusability, this system provides good accessibility, prevents plagiarism, and students' understanding is tested in a way that has certain advantages when compared to traditional examinations.

1.1 Considerations for Automated HCI Assessment

For the assessment to be computer-marked, it is necessary either to convert paper answers to an electronic format, or have answers submitted via computer. Computer submission requires several reassurances to both students and lecturers. Students wish to ensure that their answers are correctly recorded and marked, and the marking should be verifiable in case they want to quibble (the issues are decidedly like those of electronic voting). Academic staff, in addition, wish to ensure that students are not cheating.

The most workable arrangements for the course involved holding the on-line test in computer labs, at the same time as regularly scheduled practical lab sessions. As the number of students on this course is typically large (of the order of 150) compared to the size of the computer labs (approximately 18 computers per lab), multiple practical lab sessions were necessary, with the following consequences:

- Students can typically see the screen of the person sitting next to them easily.
- As not all students have their practical lab sessions at the same time, it must be assumed that students taking the test on one day will communicate with students taking the test the following day.
- There are various cheating devices (technological and otherwise) in existence. For example, a person in another computer lab could "sniff" the screen of a person sitting in the test room, and if the questions were up on the screen, it would be possible to communicate answers via a small undetectable mechanism like a vibrating mobile phone.

With the difficulties of enforcing the ban on mobile phones or other covert technologies, one aim for the assessment system was to have test conditions that rendered illicit communication practically ineffective.

There is another substantial challenge: whilst exam questions can allow open-ended responses, automated assessment is only suitable when answers can be clearly categorized as right or wrong, as Carter (2003) points out. HCI is not a subject where the issues are black and white, as situations can be subjective to several interpretations, and thus a major issue for an HCI assessment system concerns structuring of test questions to allow for answer categorization. Inspiration from other institutions wasn't available either, as whilst there is plenty of material on the online assessment of programming in computer science, for example (Lister, 2000, & Malmi et al., 2002), there is less on other computing topics, and a search for information about automated HCI assessment was unsuccessful.

2. AN AUTOMATED ASSESSMENT SYSTEM

2.1 Question Structure

The format of question decided upon was this: for each main question, a situation is described (the *stem* of the question) along with several small multiple choice questions (MCQs) pertaining to that situation. A sample assessment question of this form is illustrated in Figure 1.

Question 5			
A web design company is developing a web site for a health foods shop, and has created a prototype site. One of the company's employees explores the prototype site carefully and systematically, looking specifically at responses to any actions that potential customers might perform. For example, in response to a customer clicking on a button to put an item into a shopping basket, there should then appear a clear indication to the customer that an item has been put into the shopping basket, and which item it is. Any responses that are missing, or inadequate, are noted and reported to the design team.			
Please indicate whether the statements below are True, False, or that you Don't Know.			
	True	False	Don't Know
a) The evaluation by the employee was <i>formative</i> .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b) The evaluation involved <i>GOMS analysis</i> .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c) This situation describes an example of <i>heuristic evaluation</i> .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scoring: Correct answer = +2, Incorrect answer = -2, Don't Know = 0			

Figure 1: Example assessment question, showing grouped MCQ structure

It is felt that this format has several advantages:

- Rather than asking for explanations, which usually results in blind regurgitation of lecture notes, these questions have potential to show that students understand the concepts.

- The arrangement linking several MCQs with a single stem means less time reading and understanding the descriptions of the situations than with several separate MCQs.
- Such questions are quick to answer, as students don't have to spend time thinking of how to phrase their answers and laboriously writing them out.

In general, questions can include diagrams, and don't have to be restricted to a binary format like True/False, e.g. an alternative format might be Always/Sometimes/Never. However, each MCQ has to be carefully formulated: the description in the stem of the question must give sufficient information so that the correct answer is clear to a student knowing and understanding the assessed concepts. As is necessary with MCQs, care has to be taken to ensure that it is not possible to get substantial marks simply by guessing, and as recommended by Bull & McKenna (2004), a system using negative scoring for wrong answers was used, to ensure that the expected number of marks from guessing is suitably low. To prevent students being forced to guess if they did not know the answer, an option of "Don't Know", scoring 0, was always available.

2.2 Question Bank and Randomization

To avoid a student cheating by looking at a nearby computer or talking to students who previously sat the test, randomization of questions had to occur. The test questions were grouped into equivalence classes. Each question in an equivalence class has the same format and marking scheme, specified by a *blueprint* for that equivalence class, and this ensures that all students are demonstrably assessed at the same level.

Question Bank:	Class Test December 2004
Equivalence Class:	<i>EVAL</i> (for qns on usability evaluation)

A description is given of a usability evaluation or testing situation. Students are then asked to say, for each of three usability evaluation concepts, whether the concept is applicable to this situation.

The three concepts will include

- "formative evaluation" or "summative evaluation"
- 2 other concepts from the following list of topics taken from the lecture notes: "GOMS analysis", "cognitive walkthrough", "heuristic evaluation", "iterative design", "performance measurement" "thinking aloud protocol", "focus groups"

Answers are either True (+2 marks), False (-2 marks), or Don't Know (0 marks).

Figure 2: Example blueprint for equivalence class

For example, the blueprint for an equivalence class containing the question in Figure 1 might be that shown in Figure 2. Questions within an equivalence class are deliberately constructed to have a variety of orderings of the small MCQs, with mixed patterns of correct answers.

Each test paper is generated from a bank of questions where there are several equivalents for each question. This minimizes plagiarism problems arising from multiple runs of the class tests, and allows for later runs, in the case of students having medical reasons for delay.

2.3 A Student's Perspective of the Test

The student is given a printed test paper, and logs in at a computer, going to the web page with the answer sheet on it (this could be a different version of the web page if needed for accessibility purposes, but apart from that, the answer page is the same for everyone). The student then fills out the answer sheet both on the screen and on the paper (in case of power failure and/or for independent verification purposes), along with their student ID number, and a unique keyword identifier written on the paper. Figure 3 shows an example part of the test web page corresponding to the question in Figure 1.

A student may be able to see the screen of a nearby computer easily enough, but this will not assist cheating, as the questions are not written on the screen, but only written on the printed test papers. As the test

papers are not easily readable by a nearby student, and are individual (the students know about the randomization), it is not easy for a student to exploit the answers of a neighbouring student.

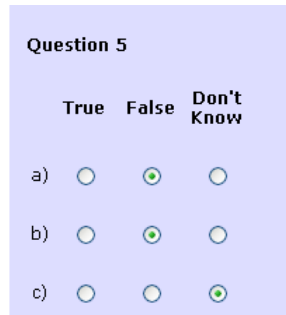


Figure 3: Section of web page corresponding to sample question in Figure 1

The student hands in his or her paper before leaving the test; no test papers leave with a student. When the student discusses the test with classmates, it will be discovered that they had different questions. These measures are crucial, to counteract the ease of a student seeing a neighbour's screen.

2.4 The HCIQ System

The Human Computer Interface Questions (HCIQ) system facilitates generating the questions, collecting the answers, and marking the results. The diagram in Figure 4 indicates the architecture of HCIQ.

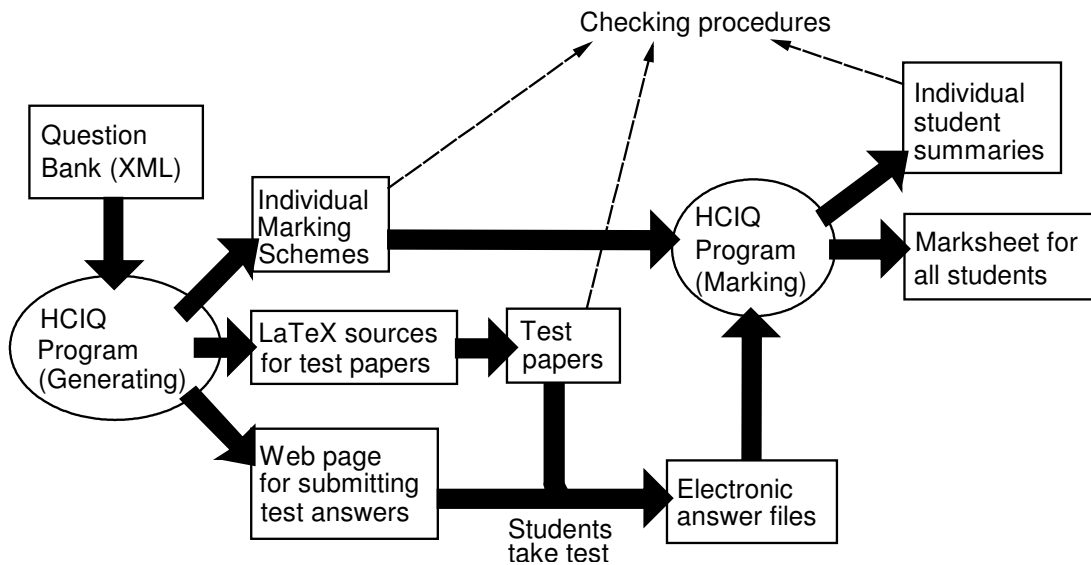


Figure 4: The architecture of HCIQ

The questions are written in (or converted to) XML format, including marking schemes. The HCIQ program is capable of reading the XML files from a question bank, checking the format and marking schemes of questions to make sure that they are sensible, and producing the test materials:

- LaTeX source for each of the individual test papers;
- electronic files containing the marking schemes for the individual test papers;
- an HTML file containing the online answer page.

The individual test papers are made up with one question chosen randomly from each equivalence class, ordered in a way specified by the test organiser. The papers are generated in LaTeX from a template file

providing the rubric. A file of *keywords* (common easily-spelt words like "rose" or "lily") supplies unique identifiers for connecting printed papers and submitted answers to their marking schemes. If necessary, generated LaTeX files can be edited to comply with specific accessibility requirements for individuals. There are separate scripts for converting all the LaTeX files to PDF files and printing them.

The web page for submitting answers is generated from an HTML template; HCIQ also generates Javascript within the web page, to check for and prevent errors when students type in their student ID and keyword information. The web page is generated without explicit style information, but instead includes information that enables the use of a separate CSS style sheet. This means it is easy to precisely customise the appearance of the page, for example if individual accessibility requirements necessitate it.

It is convenient for an invigilator of the test to be able to instantly see at a glance that the student really is doing the test, and is not cheating by looking at other information available via the computer. This can be achieved by using a CSS style sheet that produces a distinctive recognizable appearance of the answer page. In practice, we found that one way to make the appearance of the page instantly recognizable is to colour the background of odd-numbered questions one pale colour, and the background of even-numbered questions a different pale colour. This also helps students to easily visually distinguish which question is which on the web page.

A CGI script (written in Perl) handles the submission of students' answers via the web page. HCIQ then reads those answers and marks the tests, producing individual reports (useful for providing detailed feedback to students) and a marks summary for the whole class, which can be easily be imported into a spreadsheet.

In the case of students quibbling with their marks, the electronic marking process can be verified from the paper copies. A selected sample of the papers should also be checked by hand, from the printed papers and the electronic submissions, for quality assurance purposes.

It is planned that eventually HCIQ will be able to generate self-test web pages from a practice test question bank, for student revision purposes.

2.5 Student Preparation

2.5.1 Pilot Test

Since the environment for the on-line class test might be unfamiliar to students, and the exact format of test and nature of the questions almost certainly will be unfamiliar, in order to help them to revise more effectively for the test, a pilot version of the test was provided, several weeks in advance of the real test. The pilot bank necessitated two separate banks of questions, one for the pilot test and one for the real class test.

The pilot test was not only advantageous for students, but also enabled the authors to check the usability and accessibility of the test system. Furthermore, since the students had just learnt about usability testing (an HCI topic), then the pilot test also afforded students the opportunity to participate in a real live usability test! The pilot test was given on the first half of the course material, and students also filled out a questionnaire about the system, which provided valuable usability information on the computer interface, both for usability and accessibility issues. The pilot test was also crucial for obtaining approximate timings of how long students were taking per question, in order to get the length of the test correct for the real test.

2.5.2 Revision

Students were provided with two sample papers (PDF documents in the same format as the printed test papers) as a form of revision before the e-assessment. They were provided with sample answers for one paper with a detailed rationale of why the answers were right. For each equivalence class used on the final e-assessment, there was at least one question on one of the sample papers matching the blueprint of that equivalence class. Tutors also spent some time during lectures going over a sample paper.

3. EVALUATION OF THE HCIQ E-ASSESSMENT

3.1 Implementation Choices

In choosing how to implement the assessment system, existing languages for representing MCQs were considered. QML (QML, 2004) is one such XML-based language, with clear simple question specifications, but this cannot group MCQs together. The “IMS Question & Test Interoperability Specification” (IMS Global Learning Consortium, 2005) is suitable for describing such grouped MCQs, however it is complex, and time constraints dictated using a simpler structure. However it is estimated that it will be straightforward enough to translate the questions to meet this specification in the future, should this prove useful, as the IMS specification is also XML-based and there are many XML code libraries to assist with translation from one format to another.

Although the creation of a new e-assessment system represents a significant effort, especially for such a relatively simple test, there are several reasons why we've opted to do this "in-house", rather than using a pre-existing system such as WebCT:

1. The grouped equivalence class structure with a single stem for several MCQs is non-standard and we have not found a system that can present such questions in a user-friendly way.
2. Systems such as WebCT typically show the test questions on the screen; even if the choice and ordering of the questions is randomized, a student can still copy some answers by looking at a screen of a nearby student. We are not aware of a system that can support having printed test papers with individual marking schemes, matching an online answers-only form, in a highly usable way.
3. The HCIQ system, being in-house, is more controllable through direct access to all parts of the system. This makes it possible to make ad hoc adjustments, rather than being constrained to work within whatever the given system makes available, or being forced to update things when an e-learning environment authored by someone else gets updated to a new version.

In summary, rather than adapting to the limits imposed by already available technology (the technology tail wagging the pedagogic dog), the HCIQ system is the result of being determined to provide what is needed in practice for good automated HCI testing, and forcing the technology to help, not hinder. The HCIQ system demanded considerable initial effort, but provides a system that is re-usable for many years to come, and can be customized to assess courses in other computing subject areas.

3.2 Accessibility

This was easily catered for by the pilot test, which, as it was held several weeks in advance of the real test, gave plenty of warning about potential problems. Students filled in questionnaires concerning readability, colours (of printed test papers and the web page), and font styles & sizes. Students who had access problems in the pilot test were contacted to make sure that arrangements for the real test would be suitable for their needs. A few students had had difficulty reading an 11pt font size on the written paper, and this was increased to 12 pt for the real test.

Although the HCIQ system produced HTML files that could have been used with a variety of CSS stylesheets in order to customise views of the answer pages for students, in practice all of the students were happy with the web page already provided, and none needed special arrangements for the web page.

One particular case of note was that of a motor impaired student who was in the class. For her, written exams require too much writing, and thus she requires an amanuensis to write her answers down for her, which can be problematic as some people have difficulty interpreting her speech. This student was given her usual amount of extra time for the HCIQ assessment and was able to complete it completely unaided. Other students with special needs were given similar extra time allowances.

3.3 Test Results

Compared to a traditional written examination, it was felt that the test produced reasonable results: there was a similar average and standard deviation to exam results from previous runnings of the course. It was felt that the questions did test students' understanding thoroughly, in many ways better than a traditional exam.

Students for whom English is a 2nd language found the test easier to answer than a traditional exam, because answering questions did not involve writing. Analysis of test results showed that whilst non-native English speakers performed slightly worse on the test than native English speakers, this pattern was also true of the coursework, and the average performance on the test relative to the coursework was the same for both groups.

3.3 Students' Perceptions

When comparing the on-line assessment with the standard written exam, students much preferred the HCIQ assessment. Nobody said that they preferred an exam. Typical comments were:

'Very easy to use and navigate'

'This is a very good way to test this module. This type of test makes the whole test process more interesting and less stressful. You are able to take your time and not much writing involved, which is always a good thing.'

Most negative comments from students concerned aspects of implementation rather than the test itself:

'Good idea, though defs keep filling in both ie paper and computer for a while just in case!'

'Submit button at the top not needed, could result in people submitting answers accidentally'

'I really don't like the negative marking system; I know that won't change, but I just wanted to say that'

Many students complained at first about the negative marking, possibly because they are used to being able to guess and get some free marks! Having a pilot test was useful as it allowed students to change their test strategy and make appropriate use of the 'Don't know' option that is always on offer and carries 0 marks. In our view the marking scheme encouraged students to think more about the answers rather than guessing.

3.4 Lecturers' Perceptions

We found the book "*Blueprint for computer-assisted assessment*" (Bull and McKenna, 2004) very useful for orientating ourselves to the world of automated assessment and also rather inspiring as we took on this new venture. We found that it provided a sound introduction to the common types of MCQs, together with the main considerations and pitfalls involved.

Setting questions for electronic marking required a different way of thinking about assessment. We thought more about what we wanted the students to **know** and **understand** at the end of the course. Once the blueprint for each equivalence class was established, a significant amount of creativity was required to come up with scenarios that tested students' understanding of the HCI principles and procedures examined by the particular class. Although this process involved a lot of hard work, it was far more enjoyable than marking written exams and writing questions for exams, which can easily be too tied to the recreation of lecture notes.

We spent a considerable amount of time attending to all the details that ensured the smooth running of the e-assessment process. This paid off, in the sense that when the e-assessment took place we were not required to spend any time rectifying the effects of nasty surprises.

We are now in possession of a considerable question bank of well-designed, effective and re-usable HCI questions. On future occasions far less time will be required to set up the test. As for the marking, the biggest problem with the marking for 139 students being automated and taking three minutes rather than three days, was trying not to look smug when our colleagues had high marking piles and we didn't.

We felt that the process of setting up the e-assessment had a positive effect in that it encouraged us to re-think the syllabus, what we considered important to teach about the subject, and how we could do it better. The test addressed the assessment of students' knowledge and understanding, corresponding to the two lower levels of Bloom's taxonomy (1956), and the practical group coursework involved design with application of HCI principles, so it was felt the two types of assessment complemented each other well.

4. CONCLUSION

This approach towards e-assessment for the subject of HCI in an automated way is, we believe, novel. It is hoped that this paper can provide ideas for other educators looking to assess similar subjects that also have issues that are not so black and white when it comes to answers to test questions.

This system for e-assessment was very successful and well worth all the effort put into it. With its capability for future re-use, good accessibility, plagiarism prevention, and above all, suitable HCI testing, we feel it is a good example of using a combination of technologies to provide the support best suited to the assessment, rather than trying to fit the assessment to an off-the-shelf e-learning environment.

ACKNOWLEDGEMENTS

Thanks are due to Charles Bryant, who wrote the Perl script for saving the web page submitted answers, and also to Christophe Restif and Esmyr Koomen, who helped test the system before the students did.

REFERENCES

- Bloom, B.S.(Ed) (1956) *Taxonomy of Educational Objectives Handbook 1: Cognitive Domain*. New York: Longman, Green & Co.
- Bull, J., McKenna, C., (2004) *"Blueprint for computer-assisted assessment"*. Routledge Falmer.
- Carter, J. et al (2003) "How shall we assess this?" *ACM SIGSE Bulletin*, Vol 35, Issue 4
- IMS Global Learning Consortium (2005) Question & Test Interoperability Specification.
<http://www.imsproject.org/question/> (last accessed 18-02-05)
- Lister, R. (2000) On blooming first year programming, and its blooming assessment *Proc.Fourth Australasian Computing Education Conference (ACE2000)*, Melbourne. pp 158-162.
- L. Malmi, A. Korhonen, and R. Saikkonen (2002). Experiences in automatic assessment on mass courses and issues for designing virtual courses. *In Proceedings of The 7th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE'02*, pages 55-59, Aarhus, Denmark. ACM.
- QML (Questions Markup Language) (2004) <http://cnx.rice.edu/content/m10140/latest/> (last accessed 18-02-05)
- WebCT (2005) <http://www.webct.com/>