British Journal of Educational Technology (2015)
doi:10.1111/bjet.12255

Using computer-assisted assessment heuristics for usability evaluations

Gavin Sim and Janet C. Read

Gavin Sim is a Senior Lecturer of Human Computer Interaction in School of Computing, Engineering and Physical Sciences at University of Central Lancashire. Dr Sim's research interest include methods for evaluating usability and user experience in a wide variety of contexts including games and educational technology. Recent papers have focused on the evaluation of games with children. Prof. Janet Read is a Professor of Child Computer Interaction in School of Computing, Engineering and Physical Sciences at University of Central Lancashire. Prof. Read's research interests include the study of digital ink interfaces for writing and the study of evaluation methods for use with children where her most cited work is the Fun Toolkit – a set of tools to help children evaluate fun. Recent work has included a study of cool as it applies to the design of teenage interfaces as well as several papers considering the ethics of participatory design with children. Address for correspondence: Dr Gavin Sim, School of Computing, Engineering and Physical Sciences, University of Central Lancashire, Preston, Lancashire PR1 2HE, UK. Email: grsim@uclan.ac.uk

Abstract
Teaching practices within educational institutions have evolved through the increased adoption of technology to deliver the curriculum and the use of computers for assessment purposes. For educational technologists, there is a vast array of commercial computer applications available for the delivery of objective tests, and in some instances, organisations have opted to develop bespoke systems to meet their individual pedagogical requirements. However, there is very little research published on the usability of these systems and on the possible usability problems that could ultimately affect users' learning or users' marks in summative tests. This paper focuses on the effectiveness of the heuristic evaluation method for the evaluation of computer-assisted assessment (CAA) systems and proposes a set of CAA heuristics for evaluating assessment tools. The results of these tests show that, with little training, novice evaluators can effectively perform an evaluation and could, using this heuristic set, identify genuine usability problems within the assessment tool. Therefore, educational technologists or software developers could use the new CAA heuristic set to aid their procurement or inform their design decisions.

Introduction

Since the mid 1980s, the UK has seen a rapid increase in the number of universities and students. In the mid 1980s, there were less than 60 universities and only 6% of 18-year-olds progressed to university, whilst 20 years later, there were 140 universities, and those attending accounted for 42% of 18-year-olds (Foskett, 2011). This increase in numbers is one of the factors that have constituted the need for new forms of assessment. Sodberg (2009) claims that assessment is a core activity in higher education. Partly in the light of increasing student numbers, but also as a result of improvements in technology, there has been a shift from traditional forms of assessment involving paper-based exams to the use of technology to deliver, administer and mark exams. Research has been conducted on the use of computers for assessment purposes since the 1970s (Morgan, 1979), and over a decade ago, it was argued that paper-and-pencil testing was obsolete and outdated compared with the latest techniques in teaching, learning and assessment (Parshall, Spray, Kalohn & Davey, 2002). Despite this, the adoption of computers into assessment practices, often referred to as computer-assisted assessment (CAA), the uptake in education has

Practitioner Notes

What is already known about this topic

- A range of e-assessment tools are used across the educational sector and industry.
- Usability problems within these tools have the potential to impact on learning and grades.
- Usability methods exist for evaluating technology, but none are specific to e-assessment.

What this paper adds

- Provides a set of heuristics for evaluating the usability of objective tests within the context of e-assessment.
- It demonstrates the applicability of the method and the heuristic set.
- The effectiveness of the heuristic set is shown, based on a corpus of known usability problems.

Implications for practice and/or policy

- The heuristics have potential to improve the usability of e-assessment tools when incorporated as part of a development lifecycle.
- By improving the usability of the tools, the heuristics have the potential to ensure student learning/grades are not affected by usability problems within the software.
- Educational technologists can review the usability of software before making it available for campus-wide deployment.

been relatively slow compared with initial expectations (Warburton, 2009). However, it is the case that more educational institutions are increasingly adopting technology to deliver curriculum, and the use of computers for assessment purposes is gradually increasing (Conole & Warburton, 2005; Crisp & Ward, 2008). Beyond educational institutions, global corporations such as Cisco, Intel and Microsoft are using computers for their accredited courses launching an initiative *Transforming Education:* Assessing and Teaching 21st Century Skills focusing upon transforming educational assessment and instructional practices (Cisco, Intel, & Microsoft, 2009). As technology continues to evolve and users' expectations increase, the effective use of technology in the teaching and learning process has become essential (Brink & Lautenbach, 2012). With these technological advancements and increased adoption of CAA within organisations, there has been a rise in the number of commercial and bespoke CAA applications that enable the construction, administration and marking of objective tests. These include Questionmark Perception, learning management systems that incorporate assessment tools into their functionality including Blackboard and Moodle, and QuizCast that is available for mobile devices. Where software does not meet the pedagogical needs of the instructor or institution, then bespoke systems are developed such as OpenMark, which is integrated into Moodle to enable diagrams to be assessed (Thomas, Waugh & Smith, 2012).

CAA products need to be designed in such a way that they are simple to use and easy to access. Ease of access and the design of products to be accessible to individuals with a range of needs are referred to as accessibility; ease of use is typically considered to be about usability. When evaluating the suitability of CAA applications, an integrated approach for testing that incorporates both accessibility and usability may be desirable. Yesilada, Brajnik, Vigo and Harper (2012) highlight the fact that there are many definitions of accessibility, with some being user centred and others presenting a wider scope including equality of access. These definitions can therefore

cover a wide range of users, including those with cognitive or physical disabilities, temporary disabilities, the aging population, and users who are excluded through lack of access to technology or poor internet connectivity. Thatcher *et al* (2003) propose that accessibility is a subset of usability; however, web accessibility guidelines are proposed within their own ISO 40500:2012 (ISO, 2012); so, at least in some cases, it can be considered independent. Yet in much of the literature, the two terms remain interrelated with Shneiderman (2000) suggesting the term "universal usability" to encompass both usability and accessibility. It has been suggested that not all accessibility problems affect non-disabled users and are therefore not within the scope of usability problems (Petrie & Kheir, 2007). In focusing primarily on usability, one assumes that most of the accessibility problems have been solved. Usability therefore is more concerned with the form and design of the software. On the other hand, usability could be approached "before" accessibility with the aim to at least make the system logically useful before ensuring its use for a more diverse range of users. Either way, usability is a key requirement for CAA.

Despite the increase in software available for delivery of objective tests and the fact that many e-learning systems have integrated tools to support assessment, relative to studies investigating usability of broader educational technology environments (Berg, 2000; Nokelainen, 2006; Parlangeli, Marchigiani & Bagnara, 1999; Piguet & Peraya, 2000), there is limited research on the usability of assessment tools. Usability is an important aspect of these systems as usability problems may impact on students' learning or on their overall grades. There are a number of different definitions of usability, one definition of usability identified four dimensions that are important: effectiveness, learnability, flexibility and attitude (Shackel, 1986). These dimensions are still relevant, but the ISO 9241-11 definition is more widely adopted which defines usability as "*the extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*" (ISO, 1998). This definition refers to specific users, and within educational institutions, there are several different stakeholders involved in assessment that usability problems could have an impact on, including instructors constructing the tests, invigilators managing the tests and students completing the tests. However, for students, usability problems could impact their results and ultimately their classifications of awards; thus, this user group should be prioritised when making usability judgements about the software.

There are a substantial number of established methods for evaluating usability. Nielsen and Mack (1994) identified four methods: automatically through the use of evaluation software; formal inspection methods based on the models and formulas to identify problems; informally based on experts evaluating interfaces based on guidelines; and empirically through user studies, often referred to as user testing. Within human–computer interaction (HCI), it is widely acknowledged that user testing, through a variety of techniques including think aloud and direct observation, is the most reliable method to achieve usability in a software system (Woolrych & Cockton, 2001). However, user testing has several limitations, notably that a representative population needs to be found, and this can make user testing expensive and slow when compared with other methods. In these situations, expert methods are a viable alternative to user testing which can be used alone or can be used to compliment user testing results (Nielsen & Mack, 1994).

One of the most cited and applied expert methods is the heuristic evaluation. Nielsen and Molich (1990) first introduced heuristic evaluation in 1990 as a method for evaluating productivity software. Since the introduction of the method to the HCI community, it has become widely researched and understood and has been applied across a multitude of disciplines including educational technology. In a classic heuristic evaluation as described by Nielsen (1992), a small number of expert evaluators independently identify usability problems within a product based on the product's compliance to a number of usability principles. The experts then aggregate their individual lists of problems to form a single list of known usability problems within the system

under investigation. At this point, or while the problems are still individual, severity ratings are attached that indicate the potential impact of the problem to the end user.

Heuristic evaluations have been carried out on a wide variety of applications in different contexts, ranging from musical instruments (Fernandes & Holmes, 2002) and e-learning systems (Zaharias & Koutsabasis, 2012). To evaluate usability in CAA, heuristic evaluations remain a viable option for educational technologists to evaluate the suitably of CAA platforms, before buying and installing them. In addition, they are a viable option for software development teams to use as part of a development lifecycle for new bespoke systems or prior to any update of an existing system. A heuristic evaluation requires there to be an expert; this is the person carrying out the evaluation. Traditionally, the expert was considered to require expertise in usability, but it has been suggested that heuristics are more effective when the evaluators are double experts, that is, experts in usability but also in the domain under investigation (Jacobsen, 1998). In considering how heuristic evaluations could be useful to educational technologists, it is acknowledged that these people may not be experts in usability, and so, therefore, this paper presents a set of domain-specific heuristics for evaluating CAA software and determines whether novice evaluators (in terms of experience as usability evaluators) can use this method to elicit usability problems.

Usability and CAA

Within educational institutions, CAA has been applied in a number of different ways. Instances include adaptive testing, analysis of the content of discussion boards, automated marking and objective testing. These applications of CAA vary considerably both in terms of their design and functionality. In developing heuristics for CAA, the evaluation of objective tests is a priority as these are the most common CAA application. The focus of the research is therefore on the evaluation of usability in CAA software that delivers objective questions.

Whilst it is the case that prior to the work described within this paper there had not been heuristics for CAA, the general consideration of usability had not been overlooked by researchers working in this field. The International Standard entitled "*Information technology, a code of practice for the use of information technology (IT) in the delivery of assessment*" (ISO, 2007) offers some recommendations on aspects of usability including guidance on navigational issues with statements such as "*the procedure for this should be simple and clear*." Whitelock (2009) emphasises the importance of tests being fair and of not disadvantaging students with e-assessment procedures. Some studies have reported specific usability problems within CAA applications; Sim, Horton and Strong (2004) identified that when using multiple choice questions with negative marking applied, the inability to deselect a radio button could result in the loss of marks. Farrell and Leung (2004) identified issues associated with excessive scrolling within the Blackboard test environment and in subsequent work set out to redesign the interface to enable a split screen to address this usability issue (Farrell & Farrell, 2011). It appears, however, that when CAA software is being produced, usability is not a primary factor in determining its suitability. In a JISC project looking at advanced e-assessment techniques (Ripley, Tafler, Ridgway, Harding & Redif, 2009), usability was not a factor that was considered in the analysis of the applications. Even when usability is considered, there is concern over the way methods might be being applied. As an example, in the development of a bespoke CAA application, Lilley, Barker and Britton (2004) claimed to have evaluated the application using Nielsen's heuristics (Nielsen, 1994). On further inspection, it appears that they used 11 evaluators who independently assessed different elements of the prototype who, rather than listing usability problems and associating them with the heuristic set, rated the interface on a Likert Scale for each of the 10 usability guidelines in Nielsen's heuristic set. This resulted in aggregated scores between 3.9 and 4.5 which were used to evidence that there were no major usability problems. These numerical scores did not reveal usability problems which is the primary purpose of a heuristic evaluation.

Within CAA applications, there are usually different interface options, often predetermined by the software manufacturer in the form of templates. The majority of teachers, instructors and lecturers will not be experienced in evaluating the usability of an interface and will therefore not question the suitability of these default templates offered. It is widely reported that software that cannot be used intuitively can lead to an increase in the rate of errors (Coiera, Aarts & Kulikowski, 2012; Johnson, Johnson & Zahang, 2000). In the case where the software is being used to evaluate a student's knowledge or understanding, the last thing that is needed is errors in the interaction; these errors could be detrimental to the user's grades. If software is not intuitive and it needs some practice, then interaction errors will correlate with experience. In a typical student cohort, there will be variability in the computer experience of the users; some users will have experienced CAA in their schools and colleges, others may be using CAA software for formative assessment, and others may use CAA for their first time in a summative assessment, in which case, any difficulties in use could be potentially quite serious. Sim *et al* (2004) suggest that one solution to this might be to give the students earlier exposure to the interface before commencing summative assessment. This is only a partial solution as the interface may alter slightly for summative assessment compared with formative assessment; for example, there may be more security features and some time dependence. This suggests that the usability of a CAA environment may alter depending on its context of use, and this should be considered when performing the evaluation.

CAA heuristics

Heuristics for evaluating software can be general purpose or domain specific. The rationale for the development of domain-specific heuristics centres upon the perceived or found ineffectiveness of existing heuristic sets for inspection of the domain under investigation. Nielsen's heuristics were used for the evaluation of Questionmark Perception and WebCT (Sim, Read & Holifield, 2006), and despite the fact that usability problems were revealed, the heuristic set was found to be ineffective, as problems were not mapped to heuristics or given a severity ratings. Therefore, using a qualitative research approach, a set of heuristics was synthesised based upon a corpus of usability problems associated with CAA (Sim, Read & Cockton, 2009). Thematic analysis was then used to identify the core themes from the corpus, and these were then used to create the heuristic set presented in Table 1. This is the heuristic set that is evaluated in this paper.

It is important that the CAA heuristics are not used without the accompanying description as this provides additional contextual clarification to the evaluators. The descriptions present some examples of the known issues that can arise within CAA applications derived from the corpus of usability problems.

To accompany the new heuristic set, it was decided that a severity rating scale applicable to CAA was necessary. When Nielsen first proposed the heuristic evaluation method, it was accompanied by a severity scale that was used to rate problems based on prioritising which would need fixing in the next iteration of development (Nielsen & Molich, 1990). In this instance, a problem that is essential to be fixed before release is given a high severity rating. This approach to the severity of usability problems is very design centred; for CAA, in order to assist the designers in determining what should be given a high priority, a severity scale focusing on the consequences of the problem to the end user, within the context of assessment, was preferred. Therefore, a scale was designed based on the impact the problem could have on test performance, rather than on the cost to fix, and this scale ranged from "dissatisfied" to "certain." As an example, in the study that derived the heuristic set described above, many issues initially identified as problems were based on personal preferences and would not have had any real adverse consequences for the user. The problem "The order of the buttons previous, next, flag is not right—should be next, previous and flag" would not affect the user's ability to navigate nor impact on test performance and therefore was coded, with this severity scale, as "dissatisfied."

Table 1: Heuristics for CAA

No.	*Heuristic*	*Description*
1	Prevent errors and the ability to recover	Prevent errors from affecting test performance and enable the student to recover from mistakes.
2	Allow user control and freedom	The test should match real-world experience, eg, chance to review and edit.
3	Ensure appropriate help and feedback	System feedback should be clear about what action is required. For complex actions, help should be provided.
4	The user should be capable of navigating within the application, and terminating the exam should be intuitive	Navigation should be intuitive enabling the user to identify where they have been, where they are and where they want to go. Options to exit should be identifiable.
5	Ensure appropriate interface design characteristics	Interface should match standards, and design should support user tasks.
6	Means of answering question should be intuitive	Clear distinction between question styles and the process of answering the question should not be demanding. Answering the question should be matched to interface components.
7	Prevent loss of input data	When answers are input, the data should not be lost or corrupted.
8	Accessing the test should be clear and intuitive	Students should not encounter any difficulty in accessing the test.
9	Use clear language and grammar within questions and ensure the score is clearly displayed	Text should be grammatically correct and make sense. It should be obvious to students what the score is for a particular question and the scoring algorithm applied (eg, if negative marking is used). Question feedback should assist the learning process.
10	Design should inspire trust and not unfairly penalise	Students should feel confident that the system will not fail. Ensure test mode does not impact on fairness and performance within the test. For example, it should be clear if marks would be lost for incorrect spelling.
11	Minimise external factors which could affect the user	Ensure that there is minimal latency when moving between questions or saving answers. Also, ensure delivery platform is secure and robust.

The unacceptable consequences scale is presented below:

- Dissatisfied: a student could be dissatisfied, but this is unlikely to affect their overall test performance—rating = 1.
- Possible: there is a possibility that the problem may affect a student's test performance—rating = 2.
- Probable: it would probably affect a student's test performance—rating = 3.
- Certain: it would definitely affect students' test performances—rating = 4.

Research questions

The heuristics presented in this paper were synthesised using an evidence-based approach (Sim *et al*, 2009). Despite this, there still remains uncertainty over their effectiveness as they had not been used, post-design, to perform a heuristic evaluation. Despite it being the case that the heuristic evaluation method is deemed more effective when double experts are used (Jacobsen, 1998), as outlined earlier, many educational institution or software development companies may not have sufficient number of staff who will fulfil this criteria; thus, novice evaluators may be required. Based upon this, the following research questions were formed:

- Can novice evaluators find real usability problems using the CAA heuristics?
- Are the proposed heuristics effective for the CAA domain?

Study design

A study was devised to investigate, using the heuristic set and the unacceptable consequences severity scale, the usability of the assessment tool within the virtual learning environment (VLE) within the authors' university. This VLE (see Figure 1) was the campus edition of Blackboard, which was formally known as WebCT, and it was used for formative and summative assessment. The objective of the study was to assess the effectiveness of the domain-specific heuristics for use by novice evaluators. This effectiveness would be based upon the ability of these novices to identify usability problems that may affect the test performance of the users and by their ability to assign a heuristic to the usability problem, thus investigating whether there was redundancy within the heuristics set. If no problems were assigned to a specific heuristic, then the question of its suitability would be raised.

Evaluators

The evaluators were 32 students who were recruited from a second year undergraduate course in human–computer interaction. All the evaluators were given brief training in the form of a lecture, and each conducted a practice heuristic evaluation the week prior to the evaluation. The practice evaluation was not in the context of CAA; the students were required to evaluate a website using Nielsen's Heuristic set to gain limited experience of the process of carrying out the evaluation. Therefore, the evaluators were judged to be novices when it came to performing the evaluation of the CAA software. It is doubtful that they could be classified as double experts, experts in usability and assessment based upon the training and experience.

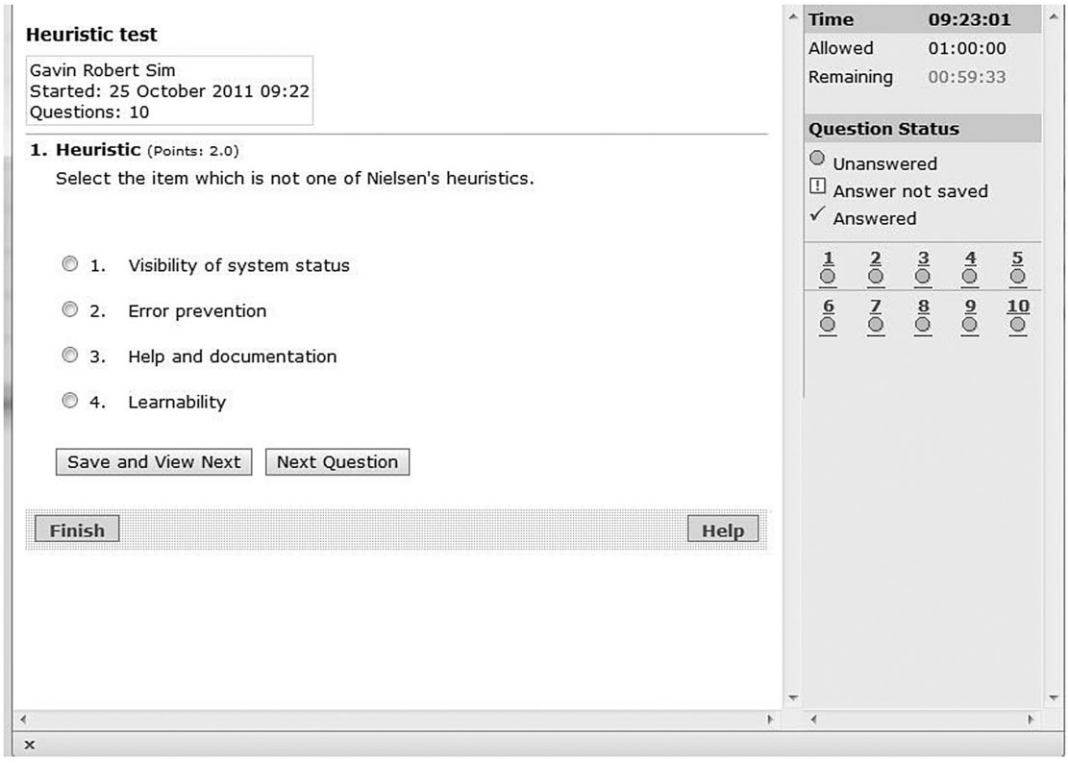

Figure 1: The assessment tool used to perform the heuristic evaluation

Question design

In order to provide a reasonable user test, it was necessary to provide a "test" environment for the evaluators. To do this, 10 questions were designed by the tutor based upon the HCI curriculum and covering topics from the course. Three of the questions were designed to be deliberately vague or have grammatical errors to determine whether the evaluators would identify these. The questions used two question styles that were known to be used for assessment purposes within computing: multiple choice (used in eight questions) and true or false (used in the remaining two questions) (Romero, Zafra, Luna & Ventura, 2013; Sim & Holifield, 2004a, 2004b). The number of questions was limited to ensure that the evaluators would have the opportunity to complete the test several times during the evaluation.

Procedure

All of the evaluators performed the heuristic evaluation in the same computer laboratory as this ensured there was little technical variability, such as monitor resolution or bandwidth. The whole evaluation was scheduled to last 1 hour, and the students were asked to:

- Log into their Blackboard account and locate the HCI test;
- Individually go through the test and answer the questions while also recording usability problems on the form provided (they could repeat the test several times);
- In groups of either three or four, merge their individual problem sets into an aggregated list of usability problems.

While completing the individual tasks, the evaluators were required to record any usability problems encountered on a form provided (see Figure 2); this was based on the evaluation forms provided in the DR-AR model (Woolrych & Cockton, 2002). In allocating usability problems to heuristics, the evaluators were allowed to categorise a single problem as a violation of multiple heuristics; this method is seen in other studies (Zhang, Johnson, Patel, Paige & Kubose, 2003).

Once individual problems had been mapped to heuristics, the evaluators were then asked to form groups of between three and four in order to merge the problems into a single list using the form provided; an example of a completed sheet is shown in Figure 3.

The evaluators were informed that problems could be merged if they were judged to be the same based on the description and also on where (in the system) they were noted. For each problem, the students had to provide a description of the problem, record the frequency of discovery within the group and assign an agreed severity rating.

Problem found (write a single problem in the space)	Heuristic(s) violated	Where it was violated		How was it found	Severity Rating
		Task	Location		
No visible icons to forward / backward the questions.	Accessing the test should be clear and intuitive	☐ Accessing the test ☑ Navigating within the test ☐ Answering the question ☐ Finishing the test	Within each question.	☐ Scanning for problems ☑ Systematically searching for problems ☐ Trying to force errors ☐ Following users task	2
When re-opening the test, the user shouldn't be able to do the test again.	Design should inspire trust and assist in fairly forgiving. Error prevention	☑ Accessing the test ☐ Navigating within the test ☐ Answering the question ☑ Finishing the test	Within the test.	☐ Scanning for problems ☑ Systematically searching for problems ☑ Trying to force errors ☐ Following users task	3

Figure 2: Completed individual evaluators sheet

Merged Problems

Problem	Frequency	Severity
Navagational buttons no previous questions buttons	2	2
Not clear whether user should click 'Save + view next' or 'next question'. Dialogue box appears which wastes time for user who has set time for test	1	3
Shouldn't be able to restart test but doesn't overright the previous exam result	2	3
Spelling error on test title	1	2
Due to poor layout and grammar, Q4 is difficult to understand	1	3
The 'help' button isn't relevent to using + navigating the test	1	3
multiple correct answers for Q 3 which wasn't made clear	1	4

Figure 3: Completed group evaluators sheet

Analysis

Across the study, a total of nine aggregated lists of problems were produced, along with the individual evaluators forms. The aggregated forms were analysed by the first author and an educational technologist to verify that the problems predicted were not false positives. The heuristic evaluation method is an inspection method where evaluators are required to predict problems that they think users will encounter; this means that on occasions, usability problems can be included in the aggregated list that are not real problems in that a user would not have a problem, hence the phrase "false positives." To understand the effectiveness of the evaluators, the aggregated lists were compared with an earlier corpus of known usability problems that were derived to synthesise the original heuristic set (Sim *et al*, 2009). This corpus consisted of 34 usability problems synthesised from an initial corpus of over 300 usability problems that were systematically merged and filtered to produce the final corpus. If the reported problem from the current study was found within the corpus, then it was assumed to be a "real problem"; otherwise, the problem was further analysed by re-examining the software to determine whether it is a viable problem or whether it was indeed a false positive.

The second stage involved the aggregation of the groups' forms (nine set of problems) into a single list of usability problems. This was performed to understand how effective the groups, as novice evaluators, were at finding and reporting problems.

Results

The results are presented in three sections. The first is the individual evaluators' results, followed by the group's performance, and finally, the aggregation of the groups' data is presented.

Individual evaluators' performance

The individual evaluators collectively identified 113 problems before their individual sheets were aggregated within their groups. Of these, 91 (80%) were matched to an appropriate heuristic, and 22 were not assigned to any heuristic. However, of the 32 evaluators, five did not classify any of their problems to a heuristic, suggesting they did not necessarily understand the process, the forms or they felt the heuristic set were not suitable. There were two individuals who only partially assigned their problems to a heuristic; these accounted for four of the 22 problems and are reported below:

- The answer is in the title of the question
- Having 5 tests open at once
- Inconsistent use of backspace goes back a question but only on certain ones
- Save and view next, next question unclear which one to press.

Upon further analysis, it would appear that these problems could have been matched to one of the CAA heuristic; for example, the last two problems would clearly be a breach of Heuristic 4: *the user should be capable of navigating within the application, and terminating the exam should be intuitive*.

The number of problems allocated to each heuristic is shown in Table 2; the heuristic number corresponds with the description in Table 1.

It was possible for the evaluators to assign a problem to more than one heuristic; for example, one evaluator stated that the backspace will make the webpage expire but not on all pages. This was claimed to have violated Heuristic 1, *error prevention and recovery*, and Heuristic 7, *prevent loss of input data*. These two heuristics might be judged to be similar, but the first focuses upon generic attributes of the software, while the later focuses the evaluator upon the questions itself as the consequences of errors here could affect test performance. It might be the case that any violation of Heuristic 7 could also be attributed to Heuristic 1 as they are both addressing issues associated with data recovery. Overall, there was no redundancy in the heuristic set in so far as each heuristic had unique problems associated with it, suggesting that the heuristics offer adequate as well as essential coverage of the domain.

Group problems

In total, 56 problems were reported by all the groups, and the average number of problems per group was *Mean* = 6.2, *SD* = 1.85. Groups B and E identified the lowest number of problems with 4, while group D had the highest with 9 (see Table 3).

In the table matched to corpus are problems the group reported that were identified in the established corpus of 34 usability problems; unique are new problems that might be specific to this test, and false positives are inaccurate predictions. Overall, 84% of the group problems, 47 of the 56, matched known usability problems identified in the corpus. For example, issues with

Table 2: Frequency of problems matched to a heuristic

	Heuristic number											
	NA	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*
Number of problems	22	17	13	18	10	5	9	9	8	15	4	4

Table 3: Total number of problems found by each group and the number of problems that matched the corpus

Group	*Problems found*	*Matched to corpus*	*Unique*	*False positives*
A	8	5	2	1
B	4	4	0	0
C	5	4	0	1
D	9	7	0	2
E	4	4	0	0
F	5	4	0	1
G	6	5	0	1
H	8	8	0	0
I	7	6	0	1

navigation have previously been identified, and this was also reported in this study with the use of two buttons, one with the label *Save and View Next* and the other with *Next Question*. Users were unsure which button they should be pressing as having two buttons appear is rather unnecessary for the same or similar functionality.

The groups identified five unique problems that had previously not been identified in the corpus or were unique to characteristics of this test; the problems are as follows:

- Had five exams open at once;
- Inconsistent use of back button, can go back but only on certain questions;
- Answer to question 4 is in the title (reported by two groups);
- Title inconsistent;
- Capital letters on some questions and not others.

The problem *had five exams open at once* is unique, and despite the fact the corpus was synthesised based upon data of over 300 individuals, no one actually attempted to do this. The severity of this issue is very much context specific; if the test was for formative purposes, then it would not have major implications as it is possible that the user could attempt this several times. Within a summative context, problems would arise if the user attempted to answer the questions in different versions as only one test can be submitted. If a user then attempted to submit the test again, the message "*Test already completed. Click OK to review results of last attempt*" would be displayed; it is however unlikely this scenario would occur. This problem might be addressed through Heuristic 11 as this deals with the robust nature of the delivery platform and security issues which clearly violates having five exams open at once. The last 3 points identified are issues associated with the construction of this particular test and were identified as breaching Heuristic 9: *use clear language and grammar within questions and ensure the score is clearly displayed*. Although these three problems were not part of the original corpus, they were judged to viable issues; however, whether *title inconsistent* and *capital letters on some questions and not others* would have an impact on test performance is questionable.

In addition to the unique problems, there were two problems that were judged to be false positives, these were as follows:

- Not is not bold;
- Heuristic spelt wrong in the title (reported by two groups).

The first problem might be judged to be good practice to highlight the word "not," but it does not change the meaning of the sentence and would not impact on test performance. The second point would not have an impact on test performance and did not cause the users any difficulties during the study; within the Blackboard environment, heuristics was spelt wrong in the title of the test

Table 4: Frequency of problems identified by groups

	Number of groups who identified the problem								
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Problems identified	9	0	3	3	1	0	1	1	0

as it was spelt *Hueristics*. Despite this spelling mistake, none of the students failed to access the test, and therefore, it was judged to be a false positive.

Merged group problem set

The group problems (as seen in Table 3) were then aggregated into a single list of problems with the false positive removed at this stage. From the 56 problems, an aggregated list of 18 problems was derived; for each problem, the frequency of discovery (in terms of groups) is displayed in Table 4.

There were no problems that were identified by all nine groups; the problem with the highest frequency of discovery related to navigational problems, *no clear way to go back to previous question*, was identified by eight groups. Within the software, in order to go back to previous questions, it is necessary to click on the question number in the right panel, yet within the interface, there is a next question button but not a previous question option. The quality of the help provided was identified by seven groups who stated that *Help* was not related with how to answer questions; thus, it was deemed not very helpful. However, 50% of problems were identified by a single group; therefore, the effectiveness of a small number of evaluators using the heuristics could be questioned.

Discussion

In this study, all the evaluators with minimal training found usability problems within the Blackboard assessment tool, and the majority were able to match their problems to a heuristic. Of the 32 participants, five did not match any of their problems to a heuristic; however, this might have been attributed to the fact they did not attend the training the week before the study, they did not understand the process, they were unmotivated or they could not find a suitable heuristics for any of their problems. Despite this, the problems that they did identify were judged to be real problems; there were only two false positives (in one instance, two people reported the same problem) reported out of 113 reported problems by individual evaluators. Overall, it would appear that novice evaluators using the heuristic set could identify genuine usability problems within the Blackboard test environment.

Each of the heuristics had at least one problem classified to it. There was no redundancy in the heuristic set, suggesting that they adequately represent the domain. An alternative view to redundancy is presented by Zaharias and Koutsabasis (2012) who defined redundancy as the degree to which usability problems appear relevant to more than one heuristic, suggesting that if the problem fits more than one heuristic, then they are not distinct enough. The evaluators using the CAA heuristics could state that a problem violated more than one heuristic, and of the 113 problems, 15 were classified to more than one heuristic. However, within the heuristic evaluation method, evaluators can interpret problems differently, and it is quite feasible that problems could violate multiple heuristics within the CAA set and other heuristic sets. For example, the problem *closed the window and unable to continue* could easily be classified to Heuristic 8, *accessing the test should be clear and intuitive*, or Heuristic 1, *prevent errors and the ability to recover*. Usability is normally associated with problems users encounter; thus, at an abstract level, the majority of problems are associated with errors and the ability to recover, and therefore, problems could be

classified to Heuristic 1 in the CAA set. Error prevention is also one of Nielsen's (1994) heuristics, and again, it is quite feasible that problems could be classified to multiple heuristics within this set. It is more important that real problems are identified and mapped to suitable heuristics than refine and possibly expand the heuristic sets to limit any possible overlap that could therefore make the method more complex. This view is supported in the literatures as Paddison and Englefield (2004) suggested that the number of heuristics provided should be limited in order to not overwhelm the evaluators, and Paavilainen (2010) highlighted the problem that large heuristic sets are difficult to use.

Zaharias and Koutsabasis (2012) suggested that an even distribution of problems within the heuristic set is an important criterion for determining the effectiveness of the set. However, there was an uneven frequency distribution among the heuristics with the highest number of problems associated with Heuristic 3, *ensure appropriate help and feedback*, which had 18 problems classified to it. In contrast, Heuristic 10, *design should inspire trust and not unfairly penalise*, and Heuristic 11, *minimise external factors that could affect the user*, both only had four problems. Within the context of CAA, the majority of the interaction centres upon the answering of questions and navigation between the questions; therefore, you would expect an uneven distribution. This could also be attributed to the design of the test; issues associated with Heuristic 10 tend to centre upon issues concerning negative marking and questions being marked wrong in free text questions due to spelling, and these marking algorithms and question styles did not form the basis of the test provided. Overall, it would appear that the novice users could find problems, could interpret the heuristics and could classify the problems to suitable heuristics.

In total 84% of groups problems were matched to the existing corpus, and there were only two problems (3%) that were judged to be false positives, which further supports the effectiveness of the heuristic set within the domain. Coverage is one of the constructs that has been used to determine the effectiveness of heuristic sets (Paddison & Englefield, 2004; Zaharias & Koutsabasis, 2012) and is concerned with the extent to which heuristics adequately represent the domain being investigated. Ideally, the heuristics would offer a high coverage of all usability problems within the domain. In this study, after the problems had been merged into a single list, 18 problems remained, and of these, 17 matched the original corpus (of 34 problems) suggesting a high level of coverage.

There is concern about the number of unique problems identified by groups, with nine of the 18 problems only revealed by a single group. This is a potential issue as it questions the number of evaluators that would be required to adequately perform a thorough evaluation of the software to maximise problem discovery. However, this could be perhaps associated with the time scale for the evaluation, in the fact that it was performed in a single hour, allowing the evaluators to only interact with the application for about 30 minutes. The evaluators would have only been able to complete the test once or twice at the most; thus, a number of issues may have been missed. In the study by Zaharias and Koutsabasis (2012) who evaluated an e-learning course using heuristics, the process lasted between 4 and 5 hours; thus, a more comprehensive evaluation could be performed within this time frame. Kientz *et al* (2010) used heuristics to evaluate persuasive health technologies, and after an initial 15 minutes of exploring the technologies, the evaluators were allowed unlimited time to perform the evaluation; they were also paid which would add incentive. If the evaluation session was longer, it is quite possible that the number of unique problems may have been reduced. It could be worthwhile to have evaluators perform a brief training session, this could serve two purposes: the first, ensuring the evaluators understand the process and can find and match problems to the heuristic set, and the second, if a large number of evaluators are available, then this could be used as selection criteria to establish the most effective evaluators to overcome the variability in evaluator's performance.

The heuristic set also does not cover or incorporate accessibility guidelines or recommendations. Accessibility problems may go undetected if the system is evaluated solely on the basis of the heuristic set that is designed to detect usability problems, which differ from accessibility problems (Petrie & Kheir, 2007). Even within the context of usability, different evaluation methods yield different results (Tan, Liu & Bishu, 2009), and even the same website with teams using the same evaluation method but different protocols reported different problems (Molich, Ede, Kaasgaard & Karyukin, 2004). Therefore, the heuristic set proposed in the paper could be used as part of a systematic evaluation of the suitability of the software from both a usability and accessibility perspective.

Conclusions

This paper aimed to examine whether novice evaluators with very little training could use the CAA heuristics to perform an evaluation of the assessment tool within Blackboard. The results showed that all the evaluators could successfully identify real usability problems within the system. Metrics were used to identify the effectiveness of the heuristic set based on a corpus of known usability problems. After aggregation of the group's problem sets, 18 problems out of a possible 17 matched the reduced corpus suggesting they were effective.

It is recommended that training be provided to the evaluators to ensure that they are experienced in the method; this could simply be carrying out an evaluation of an existing system. This would enable a decision to be made as to who were suitable evaluators as in this study, it was shown that a small number may not be sufficient to uncover the majority of problems. However, this may further be addressed by ensuring sufficient time is provided to enable the evaluators to perform a thorough evaluation.

Overall, this study has shown that the heuristic evaluation method and the proposed CAA heuristics can effectively be used to identify usability problems within a CAA application with novice evaluators. Educational technologists could use the heuristic set to inform their decision making on the suitability of assessment tools before making them available for academic use. In addition, software developers could use these as part of an iterative design methodology to inform design decisions when developing new or modifying existing CAA applications.

Further research is needed to establish the effectiveness of the heuristic set in other contexts of use, for example, the delivery of objective tests via tablet devices. Additional research is required to investigate the effectiveness of the new severity scales to ascertain the inter-rater reliability. This is important to enable development teams to make accurate informed decisions on how to prioritise fixing of the usability problems detected.

References

Berg, G. A. (2000). Human-computer interaction (HCI) in educational environments: implications of understanding computers as media. *Journal of Educational Multimedia and Hypermedia*, *9*, 4, 347–368.

Brink, R. & Lautenbach, G. (2012). Electronic assessment in higher education. *Educational Studies*, *37*, 5, 503–512.

Cisco, Intel, & Microsoft (2009). *Transforming education: assessing and teaching 21st century skills*. Retrieved March 12, 2012, from http://download.microsoft.com/download/6/E/9/6E9A7CA7-0DC4-4823-993E-A54D18C19F2E/Transformative%20Assessment.pdf.

Coiera, E., Aarts, J. & Kulikowski, C. (2012). The dangerous decade. *Journal of the American Informatics Association*, *19*, 1, 2–5.

Conole, G. & Warburton, B. (2005). A review of computer-assisted assessment. *ALT-J, Research in Learning Technology*, *13*, 1, 19–33.

Crisp, V. & Ward, C. (2008). The development of a formative scenario-based computer assisted assessment tool in psychology for teachers: the PePCAA project. *Computers & Education*, *50*, 4, 1509–1526.

Farrell, G. & Farrell, V. (2011). *Online assessment: splitting the screen to be seen*. Paper presented at the 23rd Australian Computer-Human Interaction Conference, Canberra.

Farrell, G. & Leung, Y. (2004). *Comparison of two student cohorts utilizing blackboard CAA with different assessment content: a lesson to be learnt*. Paper presented at the Computer Assisted Assessment Conference, Loughborough.
Fernandes, G. & Holmes, C. (2002). *Applying HCI to music related hardware*. Paper presented at the CHI 2002, Minneapolis, Minnesota.
Foskett, N. (2011). Markets, governments, funding and the marketization of UK higher education. In M. Molesworth & R. Scullion (Eds), *The marketisation of higher education: the student as consumer* (pp. 25–38). Abingdon: Routledge.
ISO. (1998). *Ergonomic requirements for office work with visual display terminals (VDTs)—part 11: guidance on usability*. ISO 9241-11.
ISO. (2007). *Information technology—a code of practice for the use of information technology (IT) in the delivery of assessments*. pp 39. ISO/IEC23988.
ISO. (2012). *Information technology—W3C Web Content Accessibility Guidelines (WCAG) 2.0*. ISO/IEC40500.
Jacobsen, N. E. (1998). *The evaluator effect in usability studies: problem detection and severity judgements*. Paper presented at the Human Factors and Ergonomics Society 42nd Annual Meeting, Chicago.
Johnson, C. M., Johnson, T. & Zahang, J. J. (2000). *Increase productivity and reducing errors through usability analysis: a case study and recommendations*. Paper presented at the Proceedings of the AMIA Symposium, 394–398.
Kientz, J., Choe, E. K., Birch, B., Maharaj, R., Fonville, A., Glasson, C. *et al* (2010). *Heuristic evaluation of persuasive health technologies*. Paper presented at the IHI 10, Arlington.
Lilley, M., Barker, T. & Britton, C. (2004). The development and evaluation of a software prototype for computer-adaptive testing. *Computers & Education*, *43*, 1, 109–123.
Molich, R., Ede, M. R., Kaasgaard, K. & Karyukin, B. (2004). Comparative usability evaluation. *Behaviour & Information Technology*, *23*, 1, 65–74.
Morgan, M. R. J. (1979). MCQ: an interactive computer program for multiple-choice self testing. *Biochemical Education*, *7*, 3, 67–69.
Nielsen, J. (1992). *Finding usability problems through heuristic evaluation*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, Monterey.
Nielsen, J. (1994). *Enhancing the explanatory power of usability heuristics*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence, Boston.
Nielsen, J. & Mack, R. L. (1994). *Usability inspection methods*. New York: John Wiley & Sons.
Nielsen, J. & Molich, R. (1990). *Heuristic evaluation of the user interface*. Paper presented at the SIGCHI conference on Human factors in computing systems: empowering people, Seattle.
Nokelainen, P. (2006). An empirical assessment of pedagogical usability criteria for digital learning material with elementary school children. *Educational Technology and Society*, *9*, 2, 178–197.
Paavilainen, J. (2010). *Critical review on video game evaluation heuristics: social games perspective*. Paper presented at the FuturePlay, Vancouver.
Paddison, C. & Englefield, P. (2004). Applying heuristics to accessibility inspections. *Interacting With Computers*, *16*, 2, 507–521.
Parlangeli, O., Marchigiani, E. & Bagnara, S. (1999). Multimedia systems in distance education: effects of usability on learning. *Interacting With Computers*, *12*, 1, 37–49.
Parshall, C. G., Spray, J. A., Kalohn, J. C. & Davey, T. (2002). *Practical considerations in computer-based testing*. New York: Springer.
Petrie, H. & Kheir, O. (2007). *The relationship between accessibility and usability of websites*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, San Jose.
Piguet, A. & Peraya, D. (2000). Creating web-integrated learning environments: an analysis of WebCT authoring tools in respect to usability. *Australian Journal of Educational Technology*, *16*, 3, 302–314.
Ripley, M., Tafler, J., Ridgway, J., Harding, R. & Redif, H. (2009). *Review of advanced e-assessment techniques* (pp 1–28). Bristol: JISC.
Romero, C., Zafra, A., Luna, J. M. & Ventura, S. (2013). Association rule mining using genetic programming to provide feedback to instructors from multiple-choice quiz data. *Expert Systems*, *30*, 2, 162–172.
Shackel, B. (1986). *Ergonomics in design for usability*. Paper presented at the HCI 86 Conference on People and Computers II, Cambridge.
Shneiderman, B. (2000). Universal usability. *Communications of the ACM*, *43*, 5, 85–91.
Sim, G. & Holifield, P. (2004a). *Computer assisted assessment: all those in favour tick here*. Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications, Lugano.
Sim, G. & Holifield, P. (2004b). *Piloting CAA: all aboard*. Paper presented at the 8th International Computer Assisted Assessment Conference, Loughborough.

Sim, G., Horton, M. & Strong, S. (2004). *Interfaces for online assessment: friend or foe?* Paper presented at the 7th HCI Educators Workshop, Preston.
Sim, G., Read, J. C. & Holifield, P. (2006). *Using heuristics to evaluate a computer assisted assessment environment*. Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications, Orlando.
Sim, G., Read, J. C. & Cockton, G. (2009). *Evidence based design of heuristics for computer assisted assessment*. Paper presented at the 12th IFIP TC13 Conference in Human Computer Interaction, Uppsala.
Sodberg, U. (2009). A research review of e-assessment. *Assessment & Evaluation in Higher Education*, *37*, 5, 591–604.
Tan, W.-S., Liu, D. & Bishu, R. (2009). Web evaluation: heuristic evaluation vs. user testing. *International Journal of Industrial Ergonomics*, *39*, 4, 621–627.
Thatcher, J., Waddell, C. D., Henry, S. L., Swierenga, S., Urban, M. D., Burks, M. *et al* (2003). *Constructing accessible web sites*. San Francisco, CA: Glasshaus.
Thomas, P., Waugh, K. & Smith, N. (2012). *Automatically assessing free-form diagrams in e-assessment systems*. Paper presented at the STEM Annual Conference 2012: aiming for excellence in STEM learning and teaching, London.
Warburton, B. (2009). Quick win or slow burn: modelling UK HE CAA uptake. *Assessment & Evaluation in Higher Education*, *34*, 3, 257–272.
Whitelock, D. (2009). Editorial: e-assessment: developing new dialogues for the digital age. *British Journal of Educational Technology*, *40*, 2, 199–202.
Woolrych, A. & Cockton, G. (2001). *Why and when five tests users aren't enough*. Paper presented at the IHM-HCI, Toulouse.
Woolrych, A. & Cockton, G. (2002). *Testing a conjecture based on the DR-AR model of usability inspection method effectiveness*. Paper presented at the 16th British HCI Group Annual Conference, London.
Yesilada, Y., Brajnik, G., Vigo, M. & Harper, H. (2012). *Understanding web accessibility and its drivers*. Paper presented at the International Cross-Disciplinary Conference on Web Accessibility (W4A '12). ACM, New York.
Zaharias, P. & Koutsabasis, P. (2012). Heuristic evaluation of e-learning courses: a comparative analysis of two e-learning heuristic sets. *Campus-Wide Information Systems*, *29*, 1, 45–60.
Zhang, J., Johnson, T. R., Patel, V. L., Paige, D. L. & Kubose, T. (2003). Using usability heuristics to evaluate patient safety of medical devices. *Journal of Biomedical Informatics*, *36*, 1, 23–30.