Formative Automated Computer Testing (FACT)

Nicoll Hunt, Janet Hughes and Glenn Rowe

Address for correspondence: Department of Applied Computing, University of Dundee, Dundee DD1 4HN, UK

Abstract
This study describes the context for the development of a tool to formatively assess information technology skills of students. The tool provides a reliable and valid assessment of word processing competency, utilizing automation to apply the test instrument via the Microsoft Office package. Tests can be designed directly by tutors, and delivered via a network. Evaluation of the tests suggests that immediate automated testing is preferred by students compared to a traditional written test. There is evidence that the tool improves the IT skills of its users, whereas a traditional written test has no such beneficial effect.

Introduction and background
In the United Kingdom, the Report of the National Committee of Inquiry into Higher Education in 1997 recognised the need for Higher Education institutions to provide a programme for all students which included information technology (IT) as one of a set of key skills. The Quality Assurance Agency was established as one mechanism for guiding institutions towards acceptable specifications of knowledge, understanding and skills for different programmes. The importance of information technology is demonstrated by its inclusion within the “transferable skills” component of subject benchmark statements published by the QAA. The set of information technology skills required tends to evolve in response to computing hardware and software developments, the changing set of skills brought by new entrants from school experience, and the expectations of tutors and professional bodies (Martin, 1997). Additionally, distinct disciplines have subject-specific requirements of IT. Nevertheless, the core components of most programmes include skills in word processing, email, information retrieval from databases and the Internet, and competence in the basic use of a personal computer. These skills are taught either by means of discrete IT classes or by embedding the teaching of IT skills at relevant points within subject-specific classes. The efficacy of the different approaches has been a popular topic for researchers (for example, see Martin, 1997). Less well researched is the most appropriate and effective method of assessing such skills.
Assessment
Although there is recognition that assessment serves an important role in learning, benchmark statements often prescribe what is to be assessed but not why or how it is to be assessed (Holroyd, 2000). Assessment can serve various purposes, including to assist learning and to ascertain if a specified standard has been reached. The first of these may address different needs, such as indicate to the student how he or she is progressing, to give the opportunity to rehearse, to motivate the student to learn and to improve, or to diagnose what mistakes were made, how or why mistakes were made, and to identify areas where further support is desirable. The second provides evidence for progression within a degree programme, and verifies that students have IT skills sufficient for successful subsequent study. Indeed many tertiary education institutions include the European Computer Driving Licence (ECDL) qualification, which provides evidence of basic skills across seven areas of IT, as a progression requirement for undergraduate students, for example in the medical disciplines.

It is still commonplace for assessment of IT to centre upon certification. Frequently, proficiency in the use of information technology is assessed by written examinations in which a student’s knowledge of jargon is tested rather than his or her practical abilities using an information technology application. In such instances, a written examination may not be a valid source of evidence of IT skills, and the reliability of such an assessment is questionable. Alternatives are sought which guarantee reliability and are more evidently valid. The potential of computer-assisted assessment to provide such features guarantees its continued popularity.

Automated assessment
Certain commercial software packages for automated assessment of IT skills test generic skills in a reliable summative style—but are prohibitively expensive for the majority of Higher Education establishments to use. Several automated test tools only use a simulation of well-known information technology office applications. Packages which use a "real" IT environment are notoriously less robust. Focussing upon assessing for ECDL, such IT assessment packages do not provide the facility for assessors to modify questions to produce a customized test. More general-purpose packages that do provide the flexibility to create new tests typically centre upon the use of multiple-choice questions or structured questions. The educational value of such assessment instruments remains controversial. "The primary criterion for selection of a question type... should not be the ease with which the response to the question gets evaluated by the computer but rather the type of learning the question is designed to assess" (Gibbs and Peck, 1995). Certainly for information technology assessment, multiple choice and structured questions do not provide authentic performance assessment which requires "an active demonstration of the knowledge in question, as opposed to talking or writing about it" (Biggs, 1999).

Formative assessment
Although computer-assisted assessment is becoming common in Higher Education, it is typically used for summative rather than formative purposes, and is paper-based...
rather than practical, even in the domain of information technology. Mooney et al. (1998) observed that many Computer Based Learning packages do not provide the learner with constructive feedback and the correct answer(s), and thereby such packages “fail to fulfil their potential”. Rountree (1987, 24) argues that: “feedback, or ‘knowledge of results’, is the life-blood of learning”, enabling a person to identify strengths, weaknesses, and showing how to build on the former and improve upon the latter. A similar observation is made by Laurillard (1993, 61): “Action without feedback is completely unproductive for a learner.” Her description of the learning process includes the iteration of action, feedback which comments about that action, reflection upon the feedback to identify suitable amendment to the action, and the consequential elicitation of further feedback.

Evidently feedback must be reasonably informative if a learner is to be persuaded to adjust an action, to attempt improvement. However the preparation and marking of tests alone is time-consuming, without the added time required to identify and provide suitable feedback—and information technology classes are frequently large! It may be argued that practical coursework should be the mechanism to test practical skills, and to provide feedback to students. There is regular criticism however that coursework marks tend to be higher than marks awarded in examinations, particularly in the case of computer studies (Yorke et al., 2000), and therefore an assessment instrument which could combine the benefits of coursework and examination would be welcomed.

Thus, the need for IT competence of students is widely accepted, but its assessment poses several problems for educators. How can:

• information technology tests be constructed easily which are relevant to different disciplines?
• students be given appropriate feedback to assist learning?
• assessments be task-based and realistic, yet cost-effective and reliable?

This paper reports upon the development of an IT-assessment tool which is sufficiently flexible and economical to be used within different subject areas across Higher and Further education, and which serves to inform teaching and learning rather than progression.

**Capabilities of Formative Automated Computer Testing (FACT)**

The software created was designed as an automated testing package for Microsoft Word, in the first instance. It was designed to:

1. use “real” office application software;
2. allow tutors to create their own tests;
3. supply feedback to students;
4. provide cost-effective delivery with reliable marking.

It consists of two components: a builder module which is used by a tutor to construct tests and manage a database of students and results, and a player module which delivers the test to the student, marks the responses, provides feedback and records the results in the database.
**Builder module**

The builder component is the “back-end” of the system. A tutor uses it to specify which tasks the student should perform using Microsoft Word, the test duration, and to define the marking scheme and pass mark for each of these tasks. Questions can be given a descriptive name, and outlined for the user (for example: “Forfar Dispatch Excerpt”, “You will be asked to perform a number of tasks based on an excerpt from the Forfar Dispatch”). This permits the structuring of a test hierarchically into topics with sub-questions (Figure 1). A wizard is provided to help the test creator specify the actions to be performed in each sub-question (Figure 2). Further and different types of actions could be added: the initial wizard set was programmed in accordance with tutor advice, and it can be extended to accommodate a range of objects and more challenging operations.

![Figure 1: Hierarchical test structure](image1.png)

![Figure 2: A “wizard” to assist question creation](image2.png)
When building the tests an instance of Word 2000 is used to take in information about the tasks that are to be displayed, for example to indicate a document file to be used in the text, and to specify which text or paragraphs are to be updated (Figure 3). Tests can be saved, edited, and printed. The builder component is used also to maintain the list of students authorized to attempt tests.

**Player module**

The player component provides students with a “front-end” to the system to take a test. After a successful log-in, a user may sit any of the tests available. The player module displays a question and a timer to indicate available time remaining to complete the task. As well as controlling Microsoft Word 2000 so that the student can use the actual application to attempt the task, the player module assesses if the task has been completed successfully and saves results to the database. Upon completion of the assessment the student receives immediately a performance result (marks awarded for each question and sub-question and percentage). For formative feedback, the student can view a complete breakdown of his/her results accompanied by explanations of where marks were lost and how to avoid such errors in future (Figure 4).

**Implementation of FACT**

The builder and player modules of FACT were written in Microsoft Visual C++, version 6. Although Visual C++ and the Microsoft Foundation Class (MFC) libraries provide all that is needed to build the graphical user interface and internal data handling for the program, efficient methods were needed to allow FACT to be run by many users.
simultaneously over a network, and also to allow FACT access to and control over running Microsoft Office applications such as Word. In addition, an efficient method of storing and retrieving both the tests created by tutors and the test results produced by students was required.

The networking and data storage problems were solved by using Microsoft Access as the database for storing and retrieving the tests and test results. Microsoft Access allows a network of users to use a single database simultaneously without conflict, and was therefore used as the data manager. C++ classes were developed to allow Visual C++ to communicate with Access, and to store the tests and test results automatically during a tutor’s or student’s session. Access also allows binary files such as Word documents to be stored as data fields, so it was possible to store the actual Word documents produced by students during a test, in addition to detailed records such as their answers to each question, and marks awarded. During the evaluation sessions (see below), it was observed that the FACT-Access link worked well when FACT was used simultaneously by the students in a class over a typical university network.

© British Educational Communications and Technology Agency, 2002.
The requirement that FACT should have control over a running Office application was solved by using the technique of automation. The automation feature is part of Microsoft’s Component Object Model (COM) system, and allows Windows applications access to each other's internal components. All Office applications provide a set of automation classes which may be used by external programs to control the application from outside. Using the Word automation classes, FACT can start up Word and then control its running. The Word automation classes allow external programs to determine virtually any property of the document being edited. Using automation, FACT is able to query Word as to the fonts, paragraph styles, and so on that have been used in various parts of the document, and thus determine which parts of a question a student has answered correctly.

Since all Office applications such as Excel, PowerPoint and Access support the automation feature, it would be relatively straightforward to extend FACT to provide tests in any of these packages.

**Evaluation of FACT**

A formal evaluation of the FACT package was conducted to:

1. test the network and multi-user capabilities of the software;
2. obtain users’ opinions on the usability of the software as compared to traditional paper-based testing methods;
3. assess the ability of FACT to increase the IT skills of users.

**Structure of the evaluation**

The evaluation was conducted using 23 students from one of the first year IT classes at the University of Dundee. The scope for a larger sample size was limited by the University’s policy of small classes for certain practical laboratory work. Approximately three months earlier the students had completed a level 1 course in IT which included instruction on the use of word processors. During the one-hour evaluation session, students carried out three tasks:

1. a 20-minute computer-based IT skills test using FACT;
2. a 20-minute paper-based IT skills test;
3. a written evaluation comparing the two testing methods.

The computer-based test required students to create and edit two Microsoft Word documents using actions such as changing font, applying text styles and changing paragraph indentation. The test was created using FACT’s builder module as described above. The FACT builder framework can support tests as complex as any the ECDL syllabus suggests; in this case the tasks set were not particularly demanding, in consideration of the limited time available for the test.

The paper-based test also centred on students’ knowledge of the Microsoft Word package. The written test was more qualitative than the computer-based test: it asked students to define terms used in word-processing (such as “style”, “non-printing characters” and “hanging indent”), as well as describe how certain actions are performed.
using Word. Its structure was based on level 1 examinations given to students in IT courses at the University. The test was marked manually.

The students were divided into two groups. One group of students did the written test first, followed by the FACT test, with the other group of students doing the tests in reverse order. Finally, all students completed a questionnaire comparing the written and FACT tests. Respondents were asked to state positive and negative aspects of both tests, which method they preferred and which they found more demanding, giving reasons.

In addition to the tests and questionnaire, the running of the FACT system over the departmental network was monitored to judge its performance whilst several users ran the program simultaneously.

**Results of evaluation**

*Networking and multi-user capabilities of FACT*

No problems were encountered with the FACT software during the running of the testing session: it appears to be robust for moderate network usage (around 15–20 simultaneous logins).

*Users’ comparisons of written and FACT testing methods*

The overall response from the students was that they preferred the FACT test to the written test. Twenty-two of the 23 students in the evaluation session preferred the FACT test to the written test. The reasons given largely referred to the facts that no handwriting was required, that on-line help was available during the test, and that the computer-based test took less time. They also appreciated the ability to use Word directly from within the test, which was perceived as a more natural testing environment than having to describe Word’s capabilities on paper.

It is interesting to note that students did not prefer the FACT test because it was perceived to be easier than the written test. They found the computer-based method to be a fairer way of assessing IT skills, since the package allowed them to use directly the application on which they were being tested.

The students’ responses to the question as to whether the FACT or written test was more demanding were more varied. Seventeen of the 23 students (approximately 75%) found the written test to be more demanding, but not because the questions on the written test were perceived to be more difficult. The main reasons given were that students found it more difficult to explain actions rather than simply to perform them. The availability of on-line help was also cited by many students as a factor in finding the FACT tests less stressful. The six students who found FACT to be more demanding named the countdown timer as the main reason. The added pressure of being constantly reminded how much time they had remaining was distracting to them.

*The effect of FACT testing on users’ IT skills*

Some judgement of the efficacy of the FACT system in improving the IT skills of users can be gained by examining the test results in the evaluation sessions. If FACT has an
effect on these skills, this should be visible as a statistically significant difference between the mean scores in the written tests in the two groups of students. Those who did the written test before the FACT test should have a lower mean than those who did the test afterwards. The results are shown in the table:

As can been seen from the table, there is a difference of 17.6 in the means of the two groups, with those students doing the written test after FACT having a higher average mark. A two-tailed t-test was performed on the data to assess the significance level of the result. The value of 0.03 quoted in the table signifies that there is a probability of 0.03 (3%) of the two sets of data being drawn from the same distribution. In other words, there is a 97% chance that the results indicate a significant difference between the two groups, and that the results cannot be explained by chance.

However, since the two groups consisted of different students, we need to take account of the different abilities of the students in the two groups. To do this, we obtained the final examination marks obtained by the students in their first year IT course. These data are shown in Table 2.

We can see that the second group of students scored slightly (5.7%) higher in their IT course. The t-test for these data shows that there is a 75% chance that these results are significant (that is, due to a real difference in ability between the two groups, assuming that the marking procedures were uniform), so we can compensate for this by hypothesizing a difference in the mean of 5.7% between the two groups in our written test data shown in Table 1, and then redoing the t-test. The new t-test tells us that if there is a difference of 5.7 in the means of the distributions from which the two sets of data are drawn, there is an 87% chance that the differences seen in Table 1 are significant. Although the significance level has dropped from the 97% in the original analysis, it is still a strong indication that FACT is having a positive effect on students’ IT skills.

Table 1: Scores on the written test

<table>
<thead>
<tr>
<th></th>
<th>Written test before FACT</th>
<th>Written test after FACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean score (%)</td>
<td>44.6</td>
<td>62.2</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>18.1</td>
<td>17.1</td>
</tr>
<tr>
<td>t-test</td>
<td>0.03</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Final examination marks in first year IT course

<table>
<thead>
<tr>
<th></th>
<th>Written test before FACT</th>
<th>Written test after FACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean score (%)</td>
<td>58.7</td>
<td>64.4</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>11.3</td>
<td>11.2</td>
</tr>
<tr>
<td>t-test</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>
Finally, we can examine the scores obtained by the two groups of students on the FACT test.

<table>
<thead>
<tr>
<th>Written test before FACT</th>
<th>Written test after FACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean score (%)</td>
<td>77.9</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>15.4</td>
</tr>
<tr>
<td>t-test</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Here, we can see that there is much less difference between the two groups, which is confirmed by the t-test which states that there is a 59% chance that the differences between the groups are significant. However, since the difference between the means for the two groups is fairly small, it would seem that taking a written test before taking the FACT test has very little effect on the results of the computer-based test.

From this analysis, there is evidence to support the claim that the FACT testing system does improve the IT skills of its users, but that sitting a traditional written test does not have such a beneficial effect.

**Discussion**

The automated assessment tool described provides a method of reliably determining the practical competences of students in their use of an Information Technology application, specifically word processing. Tests are created from the functions and data types of the Microsoft Word type library: this permits the use of the word processing application itself within the test, thereby providing meaningful assessment. The task wizard concept used to construct questions is versatile. The wizard set can be extended to allow the assembly of a variety of task types with a range of complexities, in order that students are assessed at an appropriately demanding level. The techniques used can be applied also to other typical office applications such as the use of spreadsheet software and presentation software.

Students’ responses suggested a preference for the automated test, which provided instant non-adversarial judgement and informative comment about areas of error and methods to prevent further such error, rather than a written test. Moreover, there is evidence that use of the automated testing system improves the IT skills of its users, whereas the traditional written test does not have such a beneficial effect. In a geographical context, Charman and Elmes (1998) also showed that computer-based assessment could be effective in delivering improved student performance as well as being popular with students. These findings contrast with the work of Lee and Weerakoon (2001), whose study also found an overall difference in students’ performance using two types of test (computer-based and paper-and-pen-based), but in favour of the paper-and-pen-based test. However that study considered only multiple-choice questions, where the mode of delivery of the assessment was less important than the validity of
the test items. The present study suggests that not only are automated assessment and formative assessment types compatible, but that automated assessment may support improved performance in a subsequent decontextualised assessment involving a declarative paper-and-pencil test. Thelwall (2000) suggested that “computer based assessment can enhance the value of education in certain circumstances, either because it is intrinsically better than paper assessment or because the assessment would be impractical without it”. The tool outlined in this paper succeeds because of both reasons: it provides a motivating, formative form of IT assessment which is an authentic and practical to implement reliably on a large scale in an IT laboratory.

References