FLEXIBILITY AND THE TECHNOLOGY OF COMPUTER-AIDED ASSESSMENT

R. D. Dowsing

School of Information Systems, University of East Anglia. UK
Email: rdd@sys.uea.ac.uk
http://www.sys.uea.ac.uk/cats

ABSTRACT

There are many different facets to flexibility in computer-aided assessment, depending on one's viewpoint. As an example, for the developer increasing the flexibility of an assessment product means increasing the development cost but also increasing the size of the potential market. For the examiner, flexibility means the ability to use assessment aids in the specific way and for the specific purpose which he/she requires. For the candidate, flexibility means being given a range of ways to answer a set of questions so that he/she can demonstrate his/her knowledge/skill to the best effect.

The higher the level of knowledge/skill to be assessed, the more difficult the assessment, the more flexible the assessor needs to be and the greater the involvement of human assessors in the assessment process. For simple types of assessment, for example, the use of multiple-choice questions, the assessment process can be almost completely automated and little human examiner involvement is required. For more complex assessment, for example, assessment of a design rather than an implementation, the candidate has many more options available and the assessment is not simply in terms of true or false but rather in degrees of correctness. At such levels computer software acts as an aid or filter to the human examiner, marking some attempts but passing attempts which are difficult to assess to the human examiner. The optimum balance, in terms of cost-effectiveness, between automatic assessment and the use of human examiners varies with time and is very sensitive to the number of candidates to be assessed.

Most of the current computerised assessors assess outcome rather than method since this is easier to automate. Techniques are now being developed which allow the method used to generate the answer to be collected and assessed. This will give the examiner additional flexibility in the assessment since learners can be assessed by outcome but professionals can be assessed by method as well as outcome. For example, in assessing IT skills a university student may be assessed for the ability to word process an essay correctly whereas a professional typist may be assessed for the efficiency of editing as well as correctness.

This paper describes the technology required to add flexibility to computer-based assessors, with examples, and shows how adding flexibility to an assessor expands the potential uses.

KEY WORDS

Computerised assessment, IT skills, skills assessment.

1. INTRODUCTION

There are three stakeholders in computerised assessment; the system developer, the examiner and the candidates. Each stakeholder has their own requirements of the assessment system and these requirements can conflict. For the developer, the least risk strategy is to build a system with as much flexibility as possible so that it can then be tailored for specific uses by different examiners. Such a system will be applicable to a wide market and thus allow development

costs to be amortised over a larger range of sales than a more specialised product. However, a general-purpose product is never as good as a specifically targeted product for a particular application and thus the developer has the difficult task of balancing the development flexibility – and hence development cost – against the range of applicability of a product.

Computerised assessors are changing rapidly, partly due to technology improvements, partly due to improvements in algorithms and partly due to the increased use and market for such assessors. Presently available assessors are primitive and there is likely to be rapid development of more flexible and sophisticated systems over the next few years. Developers must possess the ability to predict what future developments might be and build the required flexibility into current products to enable new developments to be incorporated easily and cheaply.

There are two different types of flexibility that a developer has to consider when developing computerised assessors; flexible delivery and flexible use. Flexible delivery implies that assessment software should be able to be tailored to a specific environment, for example, to run on a selection of hardware or to offer a selection of tests. It also implies that the results of tests and exercises can be integrated into the user's present mark processing and recording system. Flexible use implies that the software can be used in different types of examination or tests, for example, formative and summative tests. Flexibility thus adds to the functionality required of the assessment system and hence its size, development time and development cost. To some extent, it is possible, by building flexibility into an assessor, to produce a small number of assessors which can be customised to the exact requirements of a large number of users.

The paper concentrates specifically on the assessment of computer-based IT skills from the developer's viewpoint and illustrates the inclusion of flexibility in the development of such assessors with examples from the author's experience.

2. GENERAL MODELS OF LEARNING AND ASSESSMENT

There are many different models of learning which have been developed over the years, some of which are discussed in Domjan (1998). A good summary of many of the well-known models can be found in Kearsley (1998). The model which many authors cite in the context of computer assisted assessment is Bloom's taxonomy, Bloom (1956). He and his committee defined a hierarchical model of learning and assessment where higher levels of the model relate to higher skills.

Level 1 Knowledge

The ability to remember and recall previously memorised information, for example, to know facts, methods, principles, concepts and procedures.

Level 2 Comprehension

The ability to grasp the meaning of material, for example, by summarising material or by predicting future trends. This level involves such processes as translation, interpretation and estimation.

Level 3 Application

The ability to apply knowledge and basic understanding to new situations using such rules, methods and principles as the situation requires.

Level 4 Analysis

The ability to break down material into its component parts, understanding the relationship between each of the parts.

Level 5 Synthesis

The ability to be able to create a new object from a set of components. This requires planning as well as analysis skills.

Level 6 Evaluation

The ability to judge the value of material based on specific criteria.

In the Bloom hierarchy, the higher the level of learning and assessment, the greater the flexibility offered to candidates in tests and exercises and the greater the flexibility required of the assessment system. Thus computerised assessors which assess higher levels of the hierarchy are more complex and more costly to produce. At the lower levels assessment returns either correct or incorrect and there is little information available to allow meaningful feedback to candidates. Consider an MCQ to test whether a candidate knows what city is the capital of France. An answer of Paris would be marked correct and anything else incorrect. For an incorrect answer the only feedback which could be given would be to give the correct answer and, possibly, an indication of why the answer chosen was wrong. At higher levels the assessment is graded for degrees of correctness and there is a considerable amount of information which can be used for meaningful feedback to candidates. Consider a question asking for a proof of a mathematical theorem. An assessor would produce an assessment based on how close the answer was to the correct answer, taking into account the method used. Feedback would consist of identifying the parts of the answer which were incorrect and feeding this information back to the candidate with explanatory comments. Thus at the lower levels exact matching algorithms are normally used whereas at the higher levels approximate matching algorithms are required which are more complex and slower. It is the use of these approximate matching algorithms which distinguishes the assessment of higher level skills from lower level skills.

Another major difference between the assessment of higher and lower level skills is the data which is assessed. Lower level exercises almost always assess the outcome of the exercise, for example, the formula typed into a spreadsheet cell as part of a spreadsheet exercise. Higher level skills can also be assessed on the outcome of an exercise but may also be assessed on the method used by the candidate to generate the outcome, for example, the sequence of key depressions and mouse clicks used. Method assessment is especially important where group-working skills are being tested.

3.0 A MODEL OF IT SKILLS ASSESSMENT

In a typical IT skills assessment, as in many other forms of assessment, the exercise/examination follows the following sequence of actions.

1. The examiner prepares the exercise and model answer(s).
2. The candidate sits exam/ does exercise.
3. The candidate's answer is compared to the model answer(s) to detect raw errors.
4. Raw errors are categorised according to the assessment criteria.
5. An assessment is generated from the error analysis.
6. The assessments is recorded for remedial help/mark generation/student competence tracking

The flexibility in the model is dependent on;

1. The amount of choice the examiner has in setting the exercise.
2. The amount of choice the candidate has in answering the exercise.
3. The method used to map raw errors to assessment errors.
4. The method of reporting the results.

Each of these is examined in more detail in the following section.

The difficulty in the assessment depends on;

1. The complexity and difficulty of the exercise set.
2. The number of equivalent correct answers.
3. The complexity of the assessment criteria.
4. The type and amount of feedback required.

4.0 FLEXIBILITY IN PRACTICE

There are many different ways in which flexibility is built into practical skills assessors. This section categorises some of the reasons and gives examples of such categories of flexibility which have been built into IT skills assessors.

4.1 FLEXIBILITY IN THE MODEL

4.1.1 Flexible Question Setting

In general, many examiners wish to have the ability to set their own exercises, rather than select from the set of exercises provided. Providing the examiner with the tools to set their own exercises can be problematic for the producer of an automated assessor. The reason for this is that the difficulty of assessing IT skills exercises depends to a large extent on the amount of interaction between individual errors and this interaction between errors can be reduced to some extent by careful exercise design. If examiners or tutors are given the flexibility to generate their own exercises there is a danger that they may increase the difficulty of assessment and reduce the computerised assessors accuracy. Providing automated setting aids that alert examiners to bad practice can reduce this possibility.

Example – WordTask Tutor's Module, Dowsing *et al*. (1996)

> The WordTask word-processing assessor contains a Tutor's Module as part of the suite of assessment programs. This module allows a tutor or examiner to customise the assessor to his/her needs. One of the functions provided allows the tutor to add new exercises to the exercise set as well as determining the amount of feedback to the student and the amount and type of reporting. This module does not include exercise-vetting checks to warn the tutor of incipient assessment problems although a development of this program for professional examination does.

4.1.2 Flexible Answers

The result of an IT skills exercise can be assessed either by assessing the outcome, that is, the final document produced, or by assessing the method used to generate the result. If the final document is assessed then the candidate has the flexibility to use whatever method he/she wishes to generate the correct result. If the method used is assessed then the candidate has less flexibility. Many skills exercises are assessed by outcome and hence the candidate has the flexibility to use whatever method he/she wishes to generate the required result.

Most IT tools provide the user with many equivalent functions to perform the same action, for example, there are many different ways of centring a heading using a word processor – centring command, tab, spaces. In some instances it is the appearance of the output of the test on paper which is assessed and thus any of the function combinations which produce the correct appearance are acceptable. In other cases, especially where several people co-operate in the production of a document, the method by which an effect is produced is important. For example, changing all the headings in a document to a different font with different attributes is simple if styles have been used but difficult if they have not. In such cases the method by which the effect has been generated should be assessed. This involves collecting and assessing the event stream which is the sequence of actions invoked by the user to generate the effect. Collecting the event stream is more difficult that collecting the result of the test but it allows extra analysis

to be performed which may be required by the assessment criteria. Thus collecting and analysing the event stream gives greater flexibility but is less general and requires a substantially greater development effort.

Example – Spreadsheet Assessor based on Event Stream Analysis

A spreadsheet assessor has been developed which collects the event stream generated during a spreadsheet exercise. Analysis of the event stream is performed to determine the sequence of actions the candidate has taken. At present this information is only used to help disambiguate errors in the final output but analysis of the event stream could be performed to assess method. This assessor is more complex that the equivalent outcome-based assessor resulting in higher development costs.

4.1.3 Flexibility in the Assessment Criteria

The method of assessment used in computerised IT skills assessors is to generate raw errors from a comparison of the candidate and model answer. The raw errors, which are simple differences between the two documents, are processed to generate the higher level errors which are referred to in the assessment criteria. The assessment is determined by applying the assessment criteria to the number and type of higher level errors. Whilst the technique for identifying raw errors can be applied to virtually all types of skills assessment, the processing of raw errors to higher level errors is driven by the assessment criteria. To make an assessor applicable to a range of different exercises, set by different institutions, requires the raw error processing to be made flexible.

Example – Assessing professional word processing examinations

We have built an assessor for a professional Examination Board where the conversion of raw errors to higher level errors is rule driven. The assessment criteria are transformed to rules which are used to drive the transformation of the raw errors to higher level errors. Changing the assessment criteria requires changes to the rule set to reflect the new criteria.

4.1.4 Flexibility in Reporting

An assessor is not usually a stand-alone piece of software; it has to interface to a management and administrative system which records details of student progress, achievement and competence. Different users have different requirements in this respect and it is important that an assessor is built with as much flexibility in its output interface as possible.

Example – the WordTask word processing assessor, Dowsing *et al* (1996).

This assessor generates a number of different reports for the examiner. Firstly, a comma-separated value (CSV) record is appended to the result record file which contains a summary of the number and type of errors found together with the candidate details such as name and number. This information can be used by the examiner to generate a pass/fail or mark list by importing the data into a spreadsheet and entering appropriate formulae into relevant columns. We rely on the spreadsheet used being able to generate appropriate output for the student database. A marked copy of the candidate's text is produced which can be used for checking and validation by a human examiner. A log file is also produced which details the errors found and the classification method.

4.2 FLEXIBILITY IN THE ASSESSMENT

4.2.1 Human Flexibility versus Computer Flexibility

The cost and difficulty of computerised assessment is very much greater at the higher levels of the Bloom hierarchy than at the lower levels. There are numerous reasons for this but the main cause is the analysis and classification of complex interacting errors. Assessment criteria often require complex analysis of the raw errors and some of the analysis is semantics dependent. In such cases, the most cost-effective solution to assessment is often to use a mixture of computerised assessment and human examiners rather than attempting to fully computerise the assessment (Dowsing and Long 1997).

Human beings are very good at understanding complex scenarios and can be very flexible in their approach. Their main weaknesses are that they can be irrational, get tired and forget easily. A computer system, on the other hand, is far less flexible, but it applies the built-in rules logically, tirelessly and without any loss of information. Thus the crucial observation is that, at the current state of computer system development, human beings and computer systems are complementary. This has the most profound effect on the way in which computer-aided assessment software should be developed; the computer software should be regarded as a tool for human examiner and not as a replacement. A judicious mixture of machine and human effort should be used in the assessment process using the strengths and negating the weaknesses of each one.

Example – Assessing professional word processing examinations

> The original reason we developed such an assessor was to overcome the problem of having to assess large numbers of students taking such tests. Although such assessment can be – and is – performed using human assessors, it is expensive, time consuming and error prone. Human examiners use their expertise to assess errors which have not been previously seen or anticipated. Building such expertise into a computer program using artificial intelligence techniques is both difficult and expensive, hence to reduce software development costs it was decided to use the filtering technique whereby scripts which prove difficult to assess by computer are passed to human examiners. Attempts which are passed to the human assessor are annotated by the computerised assessor indicating the assessment which the automated assessor was able to make. This enables the human examiner to concentrate his/her efforts on those parts which contain errors rather than having to scan the entire document. Using this filtering technique has proved very successful in practice.
>
> The development of the system will be terminated when the cost of further refinement of the system exceeds that which can be recouped by the reduction in the use of human examiners over a defined period. In addition to marking the filtered scripts, human examiners check the accuracy of the computerised assessment by sampling computer-marked scripts.

4.2.2 Flexibility in the Type of Assessment

There are two main modes of assessment; summative and formative. These differ in the amount of feedback which needs to be generated, the type of feedback and the reporting requirements. Formative assessment is intended to help students in their learning and hence as much useful feedback as possible is required. Since the feedback is to be read by the student it needs to be presented in a suitable way, with unambiguous text which is easy to comprehend. A record of the assessment may be used to identify those students requiring remedial classes but it will not be required for grading purposes. Summative assessment requires little or no feedback although explanatory information may be recorded in case of appeals against the assessment.

Example – the WordTask Tutor's Module, Dowsing *et al*. (1996)

This word processing assessment software is supplied with a tutor's module which allows the tutor to specialise the software. This module allows the tutor to specify, at a detailed level, the amount of feedback to the candidate, from nothing to a detailed analysis of the errors made and the assessment given.

4.2.3 Flexibility in Mode of use

There are two modes of operation of assessors; on-line and off-line. Off-line assessors collect the output and/or the events generated during an exercise and assess them after the exercise is complete. On-line assessors collect the assessment data whilst the candidate is undertaking the test and produce an assessment at the end of the test. On-line assessors tend to be used for formative assessment and off-line assessors for summative assessment. Technically, the difference between the two types is minimal; it is simply the difference between the assessment software being integrated into the IT tool or being a stand-alone program. In the case of off-line assessment, the results of an exercise have to be stored in files and the ease of off-line assessment is mainly determined by how easy it is to recover the relevant information from the files. For example, there is no 'standard' file format for spreadsheets so a spreadsheet assessor has to be able to read and assess a number of different file formats. If the on-line assessor is a simulation then it is relatively easy to obtain the required information from the exercise since the code is under the control of the developer but if the test uses a commercial IT tool then obtaining the required information is usually more difficult. For flexibility an assessor should be able to work in both on-line and off-line modes.

Example – the WordTask word processing as*sesso*r, Dowsing et al. (1998)

Two copies of this assessor were developed; one is based on the use of a generic word processor which we have developed from the HighEdit (1997) software library and the other is an assessor which assesses the Rich Text Format (RTF) output from any standard word processor. On-line assessment uses the generic system whilst off-line assessment assesses the RTF output. In fact, it appears that people often prefer to use the RTF version even for on-line assessment since fast response can be obtained and it also allows the candidate to use a commercial word processor.

4.3 FLEXIBILITY IN THE IMPLEMENTATION

4.3.1 Flexibility in the Assessment Environment

Assessment software has to be flexible in terms of the environment in which it can be used. This includes the type of computer, the resource requirements, the operating system and version and the networking environment – standalone, local area network or wide area network. Networking brings with it the flexibility that the candidate may take tests at a remote site but this raises the problem of authentication. Passwords are not enough since users can collaborate and there is no guarantee who is taking the test. There are no obvious solutions except to use a human invigilator. *Java,* (Gosling et al 1997), is becoming the language of choice for flexible implementation since it can be ported to different platforms with relative ease. For formative assessment it is useful to have versions which run over the Internet but it is not clear whether this is necessary or desirable for summative assessment. Most Examinations Boards are very conservative and will not consider remote summative assessment at present. They run the assessors on their computers at their site and candidate's attempts are transported, in many cases electronically, to that site. In such cases portability is not a significan*t* requirement.

Example – a version of WordTask for Schools

This version of the word processing assessor looks on a network file server – at a user-specified address – to find if there are test files to be run. If so, it loads the network files into a local directory which may also contain local files. This makes distribution of new tests simple, allows students to be given the same or different tests and provides a unified network and stand-alone version of the program. If a network file server exists, the results are concatenated to a named file on the server; otherwise the results are placed on the local file store.

4.3.2 Flexibility in the Implementation

Two different approaches to providing assessment software for IT skill tests are possible. One approach is to simulate the IT tool, instrumenting the simulation to collect whatever information is necessary for the assessment. This gives the examiner full control so that the candidate can be prevented from taking unwanted actions, for example, corrupting the file store on the computer. The second approach allows the candidate to use 'standard' IT tools which makes the test more realistic but allows the candidate to take any actions allowed by the real tool, for example, to copy a file with pre-recorded answers. Whilst the first approach appears to provide a better test environment, it is more costly and suffers from being much less flexible. The problem is that the simulation has to keep pace with the rapid evolution of real IT tools. The simplest development strategy is to use the real tools and to take precautions to attempt to prevent unwanted user actions. Such precautions involve encrypting information so that it cannot be read with standard editors, setting protection so that the user does not have the privilege to change the file contents and providing flexible tests so that the exact questions in a test cannot be known in *a*dvance.

Example – a File Management test, RSA (1995)

In a file management test developed for an Examination Board, the original software simulated an operating system environment and the candidate had to perform a series of actions which involved modifying the file structure and contents. The problem with this assessment software was that it was difficult to keep up with the evolutions of the 'real' software which was being emulated. As a result a new version of the software has been developed which uses the actual software to perform the test and assesses the actions by examining the state of the filestore at the end of the test. The test is encrypted, as are the results, but the candidate can erase/corrupt these files, even though they are 'hidden'.

4.3.3 Coping with Flexibility in the Model

An assessor can be made flexible by providing all the variables which control the flexibility as data in the form of encrypted files, that is, make the assessor data-driven. The user can tailor the software to their particular requirements by editing the information in the encrypted files using programs provided by the developer. In order to do this it is necessary to make the algorithms used in the assessment data driven, that is, to separate out the data from the code.

Example – professional word processing assessor

The complexities of this assessor arise due to the difficulty in categorising the raw errors obtained from the comparison of the model and candidate answers into the categories required by the assessment criteria. We have used a rule based system for the categorisation since this technique allows us to gradually increase the complexity of the assessment as we automatically assess more of the criteria. The software is data driven and the data is provided in encrypted form in several files so that this complex assessor can cope with changes in assessment criteria**.**

5.0 CONCLUSIONS

For a developer, flexibility in a product is desirable in order to cope with changing technology and user requirements. Flexibility allows the developer to extend the market for a product and reduce the lifetime costs. The key to building flexibility into a product is an accurate prediction of future trends.

Flexibility can be built into the IT skills assessment model by allowing the examiner to design his/her own exercises, by allowing the student multiple methods of answering the exercises, by proving data-driven rule-based assessment and by supporting multiple interfaces to administrative and managerial systems.

Flexibility can be built into the assessment process by using a suitable mix of automated assessors and human examiners. Computer-based assessors and human assessors have different strengths and weaknesses and are complementary. With the present state of knowledge, computer-based assessors cannot completely replace human examiners in many forms of complex assessment. The most cost-effective and efficient method of performing complex skills assessment is to use computer-based assessors to filter the candidate's work, marking the simple cases and passing the more complex cases to human examiners.

Flexibility in the implementation of a computer-based skills assessor can be provided by careful design of the system bearing in mind the target hardware and software platforms available, the requirements for formative and summative assessment and the IT tools available to the student.

The aims of introducing computer-based assessors are a reduction in assessment time, a reduction in cost and use of expensive human resources and an improvement in the consistency of assessment. By adding flexibility to the assessment tools, they can be made applicable to a wide range of different needs, making the cost of development viable.

6.0 ACKNOWLEDGEMENTS

The author would like to thank Prof. M.R. Sleep and S. Long for useful comments on the draft of this paper. He would also like to thank the Higher Education Funding Councils of Great Britain for supporting this work through the award of a Phase 2 TLTP award and the Royal Society of Arts Examinations Board for supporting the work on professional examinations.

7.0 REFERENCES

Bloom B.*S. (ed) (1956) Taxonomy of Educational Objectives Handbook I:*Cognitive Domain, McKay, New York.

Do*mjan, M. (1998) The principles of learni*ng and behaviour, Brooks/Cole, Pacific Grove, CA.

Dowsing, R.D., Long, S., and Sleep, M.R. (1996). The CATS word processing s*kills assessor,* Active Learnin**g**, 4, 46 – 52.

Dowsing, R.D., Long, S., and Sleep, M.R. (1998). Assessing word processing skills by computer, Information Services and U**s**e, 17, 1 – 10.

Gosling, J., Joy, B. and Steele, G. (1997). The Java Language Specification, Addison-Wesley.

HighEdit (Version 5) (1997), Heiler Software, Mitlerer Pfad 5, 70499 Stuttgart, Germany.

Kearsley, G. 1998. Explorations in Learning and Instruction: The Theory into Practice Database. URL (http://www.gwu.edu/~tip/). August 1998.

RSA (1995). CLAIT DOS file management - Information Brief, RSA Examination Board**, C**o**ventry, UK.**