Computer-Assisted Assessment in Computer Science: Issues and Software

Simon Rawles
rawles@dcs.warwick.ac.uk
Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

Mike Joy
m.s.joy@warwick.ac.uk
Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

Michael Evans
Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

February 6, 2002

Abstract

As student numbers and lecturer workloads increase, traditional methods of assessment make it progressively more difficult to conduct effective assessment and provide students with detailed, specific feedback. Therefore, academic institutions are considering ways of automating their assessment methods, at least in part. This automation is realised by computer assisted assessment (CAA) software. Though CAA software has been proven in some limited situations, it remains a technology which is yet to come of age pedagogically. In this report we discuss the issues of incorporating computer-assisted assessment into university-level teaching, in particular in Computer Science, and give an overview of the software available and its capabilities.

This document may be cited as follows:

Simon Rawles, Mike Joy and Michael Evans, *Computer-Assisted Assessment in Computer Science: Issues and Software*, Research Report 387, Department of Computer Science, University of Warwick, Coventry (2002).

Contents

1 Introduction 3

2 About the reviews 3

2.1 The Website and its maintenance . . . 3

2.2 Review methodology . . . 4

3 Technical issues 5

3.1 Installation . . . 5

3.2 Technologies . . . 5

3.3 Security . . . 5

3.4 Standards . . . 6

3.4.1 The IMS Standards . . . 6

3.5 Structure of materials, reuse and integration . . . 7

3.6 Question banks . . . 7

4 Pedagogical issues 7

4.1 Scope of functionality . . . 7

4.2 Ways of taking tests . . . 8

4.2.1 Adaptive testing . . . 8

4.3 Summative testing and plagiarism . . . 8

4.4 Marking schemes . . . 9

4.5 Feedback to students . . . 9

4.6 Feedback to staff . . . 9

4.7 Question types . . . 10

4.8 Relevance to Computer Science . . . 10

4.9 Deeper learning and Bloom's taxonomy . . . 10

4.10 Student autonomy . . . 11

5 Developer and Institution 11

5.1 CAA tool producers . . . 11

5.2 Quality of interface . . . 11

5.3 Support . . . 12

5.4 Time and money costs . . . 12

6 Ideas for further work 12

6.1 Review exercise . . . 12

6.2 Topics in CAA research . . . 13

7 Conclusions 13

8 Disclaimer 14

9 Appendix: Summaries of the reviews 14

References 17

1 Introduction

One definition of computer-assisted assessment (CAA) is 'the use of computers in assessment', including examinations and assignments. CAA software can be used to automate many of the aspects of assessment, which can broadly be divided into *delivery*, *marking* and *analysis*.

Though CAA is often promoted as an antidote to rising lecturer workload as a result of increasing student numbers, there are many reasons to use CAA technology beyond this. A key application of CAA is in formative assessment, and in this setting CAA software is capable of facilitating learning by delivering tests, automatically marking and providing appropriate feedback. In addition: progress may be more easily monitored though the application of frequent tests; student self-evaluation and autonomy is encouraged; and the interactive nature of the computer may be employed to conduct new types of assessment, for example adaptive testing. The administrative benefits include the reduction in human error, integration with institutional databases and the time savings once the software has been established and the materials prepared.

This report summarises work done in collaboration with the Learning and Teaching Support Network for the Information and Computer Sciences [18], one of 24 subject centres set up to offer subject-specific expertise, best practice and information on learning and teaching in the Information and Computer Sciences. The LTSN-ICS CAA review exercise [15] is an ongoing project, the purpose of which is to investigate a range of software currently available which have capabilities of assisting the assessment process.

In this report, we summarise this work by examining both the current CAA software available and going further to consider its application in institutions of higher education. The software chosen is intended to represent a wide range of products available, from free academic software written as proof-of-concept in educational research to commercially-written virtual learning environment software intended for mass deployment in large universities. The emphasis in this exercise has been on assessment in the information and computer sciences, though many of the products reviewed have been general-purpose question-and-answer tools. We discuss software available in general terms; individual software reviews are available from the project's Website [15].

The work presented here discusses the CAA software available at the end of 2001. The project Website [15] is kept up-to-date, and the reader is invited to consult the material there for further information.

2 About the reviews

2.1 The Website and its maintenance

The Website is seen as the main product of the research, and is intended to represent the current situation in CAA software through its reviews and associated information. The site can be accessed through the URL `http://www.dcs.warwick.ac.uk/ltsn-ics/resources/caa/reviews/`. Since there are many ways in which the information on the site can become out-of-date or superseded, the Website is maintained according to a defined 'updating methodology' which ensures the information is kept up-to-date.

Measures employed as part of this methodology include:

- general involvement with the CAA community;
- systematic accuracy checks for each review every three months;
- active interest in, and use of, feedback provided from the Website;
- provision of, and participation in, moderated discussion forums on the Website;
- a monthly Internet search activity for new CAA products;
- maintenance of associated resources, such as the glossary and tables.

2.2 Review methodology

Each of the reviews are written using a standard structure which aims to cover most aspects of the software's fitness for use. The structure is only used as a guideline, and is adapted where necessary. It should be noted that, although there are objective comparisons of the facilities available in CAA tools, many of the criteria against which the tools have been measured are subjective. On this account, the reviews should be treated with caution. In particular, comparisons in the reviews and the comparative summary tables reflect the authors' opinions.

The review structure may be summarised as follows:

Basic product information. The title and publisher of a product. The URL of the product's Website. A short 2–3 line description of the product, covering its main features or applications. The availability and licencing situation for the product. Which platforms the software runs on — including client and server requirements or plugin information where applicable.

Overview. A breakdown of the components in the software. A list of weak points and strong features of the software.

Capabilities. The types of assessment supported with reasons why they are suitable or unsuitable.

Student experiences. The ease of use of the software and how much prior IT knowledge is required to use it. Stability of the software. Whether the software has been only trialled or fully deployed. Effect on study or learning. Form of feedback and its usefulness. Degree of remote access. Results of tests for HTML correctness. General feelings about the assessment.

Staff experiences. Nature of lecturer feedback. Usefulness and appropriateness of the statistics and reporting. Degree to which an up-to-date view of the assessment situation can be presented. Extent to which the delivery of tests can be controlled. Robustness of the software. Integration with other tools the lecturer might use. Ease of use, particularly in question design. Previewing of questions. Degree of prior knowledge necessary to author, such as special markups or languages. Support for reuse of materials and question banks. Flexible specification of question forms. Effort required for upkeep of the software compared with traditional methods. General feelings on authoring, distribution, delivery, marking and administration.

Appropriateness for deployment within institutions. Upkeep requirements for staff and departments. Installation process. Effect on student recruitment and distance learning. Quality control issues. Possibilities of central collection and processing of assessment data. Whether the deployment of the software is likely to be smooth and efficient. Scalability of the technology and the kind of scale that the software is best used at.

Integration with related methods. *This section is about how well the software can be used in different learning situations.* Integration with computer-aided learning software, optical mark recognition. Application to group-based work. Types of tasks the software supports on Bloom's taxonomy. Support for staged questions — can later questions be selected from earlier ones? Randomisation of questions. Support for free text questions and the extent to which they can automatically marked. Ease of adding new question or assessment types. Ability to define and flexibility of marking schemes. Ability for students to submit binary files as responses.

Support from the originating organisation. Quality of documentation. Whether the software is actively maintained and frequently upgraded. Extent of maintenance commitment. Existence of on-site experts and their effectiveness.

Security and anti-plagiarism. Authentication. Password reqirement and security. Features in software which discourage collusion (copying from others), for example randomisation. Suitability of the software for summative assessment, summative assessment under exam conditions and diagnostic assessment. Handling of data and whether it is under an appropriate access control system. Extent to which packet sniffing is made difficult. Blocking of accesses by time or IP address. Use of encryption or digital signature techniques. Extent to which the browser can be misused. Privacy of marks, analyses, logged tests responses and other results.

Summary. A 3-4 line summary of the product, concluding the review and providing a minimum-detail judgement for those comparing products.

Additionally, the reviews are summarised in a series of tables, with rows showing different products and columns showing features of products. The tables come with their own explanations, but generally, features are shown by one filled or empty circle, in the case of features that are either present or absent, or by three circles representing a rating from 0 to 3, ranging from absence or unsuitability for a particular feature to total coverage or excellence in that feature. Tables are usually filled in after a review is complete and should reflect the judgements made in the review.

3 Technical issues

This section discusses the more operational and technical aspects of the tools.

3.1 Installation

Most of the products were relatively straightforward to install, though a common problem, especially in respect of software developed at academic institutions, is that software is typically not packaged completely. In particular, software which is intended to be cross-platform is too specific to particular machine configurations or operating systems, and required small edits to get working. In the case of a few products, installation was rather an involved process, taking hours for users skilled with computers to accomplish. Those products may therefore require installation by system administrators rather than the lecturers themselves.

3.2 Technologies

Many of the non-commercial products reviewed used established and often open technologies like Java [27], Perl [21] or the Apache Web server [1]. As new technologies emerge and existing technology develops, CAA software is often not updated to take advantage of these, and CAA therefore often lags technologically. The most commonly-supported operating systems on which the products run are the newer versions of Windows and the various flavours of Unix. However, a significant number of the commercial products operate as paid-for services on the Web.

Delivery technologies vary, though unsurprisingly the World Wide Web is a popular choice, allowing assessments to be easily distributed through its almost universal presence on the users' desktops. However, other products use custom-implemented programs, often as part of a client-server approach [16, 17]. Other products make direct use of third-party software. For instance, the Authorware multimedia authoring tool [20] is used to drive the TRIADS [19] assessment engine, and is directly used to edit questions and otherwise set up assessments.

3.3 Security

Security is an aspect of the products which has generally not been considered in design. For some products, this is simply because they are not intended for anything other than for purposes of self-testing. Many products, however, claim that they can be used for diagnostic or even summative testing. Since this requires the authentication of students and the storage of their results, as well as measures to prevent one student from intercepting another's submission and other communication, this could undermine the integrity of the assessment process. Encryption and other security measures are not commonly used. Provision of adequate authentication is somewhat easier, though in some cases passwords were not as closely guarded by the products as might be expected.

Security and reliability problems are inherent in some delivery technologies when applied to CAA [30]. Web-delivered CAA is common, though very few products use secure connections, leaving response data open to other users of the network. Standard browsers are designed to deliver Web documents rather than act as front ends to assessment software and so can be prone to various acts of cheating, perhaps by deliberate crashing of the software, access to unauthorised sites or other applications through the clipboard. Question Mark [23] provides a special secure browser which it claims overcomes many of these problems.

Some software does address security well, notably Nottingham University's Coursemaster system [16], which encrypts traffic between the administrator and the server, and uses a cross-checking session key system to protect students' communications with the server from packet sniffing. Other software appears to reach a given level of security, for example accepting passwords as part of the login process, but compromises the security by incorporating the passwords in the HTML source of each page served. A similar problem exists in some products in which the answers to the questions themselves can be read by simply viewing the source, but this is not typically an issue with more capable, commercial solutions.

3.4 Standards

Adoption of standards, such as the IMS's Question and Test Interoperability standard (IMS-QTI) [12], does not appear to be a priority for most of producers of the CAA products reviewed, and there is little information in most cases to suggest they are considering conformance, though a few commercial producers are recognising the standard and suggesting future software will use the standards to some extent. It appears that the range of question formats is becoming more diverse. Indeed, in some cases, IMS standards have been extended in an ad-hoc manner, suggesting the basic IMS standard is neither complete enough nor flexible enough for the needs of developers of the more complex products. This trend seems likely to continue, as IMS standards lag behind the products' capabilities which they aim to cover.

However, markup has been used to provide a structured and delivery-independent approach to a number of projects. In addition to the use of IMS standards [12], the WebMCQ [7] system is partly motivated by a wider agenda for the organisation of teaching materials based on a markup language, and Bristol University's TML system [13] has been established for several years as a usable markup language for quiz-like assessments, and is supported by the Netquest delivery system.

3.4.1 The IMS Standards

An industry standard structure [12] has been developed for the representation of question and test data, facilitating the use of these data between different products. The standard has been defined by the IMS Global Learning Consortium, comprising educational, commercial and government organisation members. This standard, the Question and Test Interoperability (QTI) specification, is defined in XML, the extensible markup language.

The standard itself, at the time of writing, is at version 1.2, a public draft specification. Its structures for delivery and response allow the construction of questions from a set of standard question types and the collection and scoring of responses. The standard is relatively large, with well over a hundred XML elements covering mainly the following:

- specifications for question-and-answer data and the packaging of such data;
- information about processing responses for assessment (scoring, feedback, etc.) and reporting these results;
- other meta-data;
- high-level presentational instructions (though not specifications for a specific user-interface);
- the definition of external APIs for products.

IMS have also developed QTILite, a cut-down version of the main specification which, amongst other things, only deals with multiple-choice questions.

The area of test delivery is defined in its own standard [11], covering the selection and ordering of questions. Small sets of questions may be defined and a random number of questions chosen from each bank. These can be presented in order of appearance in the question bank or in an arbitrary order, and obligatory questions can be defined. Questions may also be defined to require others (*overlap inclusion*) or to exclude others from being delivered (*overlap exclusion*). In its present form, the standard is limited to selection of questions based on random selection rather than as a result of previous responses, though a number of scenarios in which this takes place have been identified for future study. Included in these are assessment methods such as *computer adaptive testing*, in which difficulty is adjusted based on the current estimate of the examinee's ability, or simulated cases in which, for example, a medical situation is described and questions lead examinees through different situations that may occur based on their responses. However, it is still the case that the conditional selection of questions based on outcome response is omitted from the standard. The ability to define preconditions and postconditions for sections and items is also left for further study and inclusion in future versions of the standards.

3.5 Structure of materials, reuse and integration

The structure, or otherwise, of learning and assessment material has an impact on how it is used, and perhaps more importantly, reused. The development of structures and markup languages for such materials has been the subject of several projects, each with divergent results [28]. The interest in structure and reuse prompts the question of to what extent organisation of materials exists in CAA software. A related issue is to what extent are assessment and learning materials integrable into presentation media, such as the Web.

There appears to be little provision for reuse and integration in most of the software, the exception being the Virtual Learning Environments (VLE), where it is often the case that learning and assessment materials have some concept of structure in that they are presented on separate pages or modules. Most products which provide a basic level of material organisation allow questions to be stored and reused in new assessments through reference to allocated ID numbers. The IMS-QTI standards and standards associated with WebMCQ [7] detail a multilevel hierarchy, in which there are levels for courses, blocks, assessments, sections and questions. Meta-data is also well-developed in the standards, though both the concept of hierarchy and meta-data is often inadequately represented explicitly in products themselves.

3.6 Question banks

There has been much interest in a question-bank approach to assessment for CAA software. That is, software which facilitates the creation and use of a set of questions, either shared between assessments or between lecturers. This concept does not feature much in the tools reviewed, perhaps because of the relatively low penetration of standards into the products. A significant problem with the process of testing using CAA is the building of questions, which is time-consuming and surprisingly hard to do well, and the scope to organise questions, and ultimately to share them, would encourage the creation of a resource to be used year-on-year and to share with colleagues.

4 Pedagogical issues

This section considers the tools with respect to how they are deployed for use in education.

4.1 Scope of functionality

The intended capabilities and settings for the products vary significantly. Some products, for example the TACO system [25], are intended only as simple multiple choice Web interfaces for inclusion in Web-delivered coursework material. Others are intended as fully-featured question-and-answer systems addressing many types of testing, and there is a wide spectrum of functionality between these.

Most products reviewed are reasonably focussed on computer aided assessment, but some also address other areas that can be covered by the tools. Other products feature computer-assisted assessment only as part of a wider product. Commercial products seem to be gravitating towards a Virtual Learning Environment, intended as a 'virtual university in a box'. Consequently the focus shifts from CAA towards a broad general-purpose educational solution, losing many of the advanced capabilities which course organisers may require. Some CAA products are focussed around particular academic goals rather than simply assessment; for example the TML system [13] attempts to establish a markup for question-and-answer data from the days before IMS standards had become well-defined.

Typically, commercial products tend to be rather too general, favouring a "one-stop-shop" solution rather than something derived from real pedagogical requirements. In contrast, academic projects are often developed from the specific requirements of the job. A small number of products are developed from academic spin-off companies, and to some extent have the advantages of both commercial and academic practice.

4.2 Ways of taking tests

These are relatively basic, and generally lack any form of conditional questioning. This significantly limits their usefulness for formative testing, an area in which the use of a computer would seem particularly appropriate. Some products deliver using a single page with a sequence of questions and only allow for setting simple sets of questions.

4.2.1 Adaptive testing

Though the provision of feedback guides the learner to some extent, it does not exploit the interactive nature of the computer to the same extent as does adaptive testing. Adaptive testing has been defined as a method of assessment where the selection of later questions depend on the responses to earlier questions [22]. It can arguably be used as an aid to accurate assessment, as in the American Graduate Records Examination (GRE) [9], where correct answers lead to progressively harder questions and incorrect ones to easier questions. Adaptive testing can be applied in the assessment-as-learning setting as a powerful feedback reinforcement mechanism. One way in which this could be employed might be to categorise questions and then track those subjects or question types in which a student is repeatedly failing, and present more of those questions during formative or practice tests. Another way might be to employ GRE-style difficulty adjustment to ensure that every student is stretched, irrespective of their ability.

Some aspects of non-adaptive testing are lost, for example the ability to step back through a test and change answers to previous questions, since they have been used to determine future ones. Another consideration is that the question-setter must try not to set questions with any delivery sequence in mind, since — though sequences are possible with adaptive testing — they reduce the dynamic nature of the question selection.

Adaptive testing is an under-represented concept in both CAA tools and the standards that specify CAA questioning formats. A small minority of tools [8, 7] allow for conditional questioning, though this often takes a rather unsubtle form, where the outcome of simple conditional tests based on ranges of scores attained so far affects the selection of the next section, which typically consists of a large sequence of questions. As is discussed later, the IMS standards do not yet cover adaptive testing and the conditional selection mechanisms required for it.

Generally speaking, the computer has not been significantly exploited as an enabler of new assessments, rather it has been used to implement traditional assessment. Features which exploit the computer's interactive nature, such as adaptive testing and other types of guided learning, peer review systems and so on, are surprisingly rare.

4.3 Summative testing and plagiarism

Summative and diagnostic testing are two of the areas in which computer-assisted assessment brings significant benefits over traditional methods and are therefore desired application areas for lecturers using CAA tools. However, summative testing has several important requirements, stemming from the necessity to obtain an accurate set of results, and therefore to reduce the likelihood of cheating as much as possible. Plagiarism detectors may be employed after results have been collected, but there is a strong argument for choosing software which reduces the likelihood of plagiarism in the first place. The software therefore needs to address the issues of impersonation, obtaining results by computer hacking, and 'over-the-shoulder' copying:

- by allowing students to authenticate themselves adequately;
- by supporting computer and network security;
- by facilitating control of the test environment.

In general, the use of CAA tools for summative testing is problematic, due in part to the difficulty in overcoming security concerns and the ways in which a basic Web browser can be misused to cheat. Most products are in some way suitable for self-testing and formative testing, with the latter role being that in which they are most useful. A general lack of adequate reporting tools means that interpreting the results of diagnostic testing is more confusing than it should be.

4.4 Marking schemes

The use of marking or keeping score is almost universal among CAA tools. A points value, or weight, is associated with each response, and the score awarded is the total of these. Most products allow hints, and those that do usually allow a points value to be subtracted for the hint being given. Generally, the ability to define how marks are awarded over set questions is fairly flexible, though they are usually set on a question-by-question and response-by-response basis. Some products go further, allowing confidence-based assessment [25]. Others allow for the grouping of marks into grades, giving a quicker interpretation of a student's progress. Software which assesses programming ability is, of course, based on different criteria, and Coursemaster allows marking based on typographic features, language features as well as correctness in tests. A handful of systems, of which Coursemaster is one, allow the grouping of marks into grades. Coursemaster also attributes a colour to each grade to make Web-reported statistics more easily understandable.

4.5 Feedback to students

The possibility of including timely feedback to questions allows the computer assisted assessment tool to be used for learning rather than merely assessment [10]. Indeed, it is often said that the characteristics of CAA tools make them best suited to the practice of regular and frequent formative assessment for learning [5, 26]. Repeatable and accessible tests with rich and explanatory feedback can complement and reinforce delivered material effectively with little teacher effort beyond the creation of the test [24].

Many tools available provide such feedback, though often it is very much based on the delivery of stored text for each possible response. This in conjunction with a score so far or other simple statistics presented at the end of a test is often all that CAA tools present. Given this rigid framework, the feedback given always needs to be well phrased and needs to take the tool's feedback structure into account. Clearly, the feedback provided cannot be as flexible as that from proper dialogue with a person, though the ability to define feedback on anything other than the response-level is lacking. There is generally no provision for the delivery of feedback which spans several questions, for example to give a general indication of topics in which the student is (consistently) doing poorly, though some tools allow students to see their recent performance in a course. Feedback to students is a weak area, a clear missed opportunity for CAA, though it may not become feasible until schemes for question meta-data are devised.

4.6 Feedback to staff

Another important kind of feedback in the educational process is that which the lecturer receives in the form of summaries of the results of the assessment, for example statistics such as average mark and the spread of marks over the class. This is an area in which much of the software is lacking, and again an area in which computers could significantly improve the educational process by visualising the progress of their class.

Feedback to staff is most commonly embodied in the form of reporting tools, giving statistics for a group of students taking the course, and suggesting bad (e.g. poor discrimination) questions to be removed. This is a area which is lacking in the currently available CAA tools. Meaningful reporting depends partly on students being able to identify themselves to the system. Most systems don't allow this, and even some systems which do lack any reporting capabilities. Many that have reporting capabilities give only the more basic information, such as the average score or a simple table showing students' identifiers and their scores, delegating the analysis to a spreadsheet program. Visualisation of results, for example as a bar chart, is unusual. More common are tables of student name by assessment. However, features which give a more easily-interpretable display of the class progress are few and far between. Coursemaster's 'traffic light' approach is a step in the right direction.

There is considerable scope for the extension of CAA tools to provide deeper and more readily interpretable feedback to lecturers [24]. This could allow lecturers to weed out problem questions or identify areas of a course which are not being understood, rather than just identifying students who are underperforming. This type of feedback is especially valuable since what is being measured changes over time [22], and a timely reaction could improve the understanding of current and later topics significantly. The incorporation of breakdowns by subject or the use of scoring grids might be one way in which this could be achieved [10], though this depends on the lecturers ability to set appropriate questions as well as the software's capabilities. Good feedback is a combination of the products capabilities and the user's assessment-setting ability — good assessment and question design and meaningful feedback are often related.

4.7 Question types

A key aim of most computer-assisted assessment software is to support learning activities [10]. Generally the question types offered are rather simple, though a number of tools have more flexible provisions for extending question types. Multiple choice, multiple-response and fill-in-the-blanks questioning are typically provided, but often only a few question types beyond these exist. The dialogue between the student and the tool can often be rather shallow — the tools do not naturally lend themselves to the types of questioning which would invite more sophisticated thinking and learning in students [22]. The use of computers has, however, encouraged newer question types, such as confidence assessment, to be formulated, as well as the variation of questions by randomising numbers or choosing different distractors. However, both of these are under-represented in the tools, even though randomisation is an advantage for a class-based computer assessment.

Some products are not oriented toward multiple choice questioning, and offer different questioning as a result. Computer programming [14, 16] is one main focus. Diagramming is one interesting area of extension. One product [6] has simple chemical diagramming built in, and Coursemaster's diagramming extensions [29] are a very flexible and capable solution to diagrammatic testing.

Computer programming assessment tools are in general more capable, perhaps having been motivated by more specific requirements and learning goals. Coursemaster's [16] approach of allowing the user to dynamically link modules to mark responses provides the Java-capable lecturer with a very flexible solution, enabling the marking system to be extended significantly, and limiting the possibilities of assessment to that which can be done generally with a computer program, while still staying within the framework of the Coursemaster tool.

Active learning, styles of assessment which ask for text and diagrams, and other methods of engaging the student in the response are recognised to be effective forms of assessment-based learning. These too are under-represented in the tools provided, possibly due to the more complex development of these question types.

4.8 Relevance to Computer Science

Many of the products reviewed have no special bias toward the assessment of Computer Science, being general-purpose question-and-answer tools. Some notable exceptions, however, are the Coursemaster [16] and BOSS [14] systems, among whose capabilities are the testing of programming assignments.

4.9 Deeper learning and Bloom's taxonomy

Bloom's taxonomy [3] characterises questions by their level of abstraction, representing the competencies and activities involved in answering them. Though question types often encourage questions belonging to a particular categorisation, a given question type will typically be more appropriate for a given range of different levels of abstraction. Though many of the tools are able to test the lower levels of the taxonomy (knowledge, comprehension, application), and though some question types better encourage higher levels of activity than others, the lecturer often needs to be skilled in question setting and design in order to evaluate the higher levels (analysis).

Deeper learning is associated with the participation in higher-level tasks, particularly with constructed response question types, since they are often far more useful in terms of conveying the principles behind the tasks [22] and can promote higher-order learning through reflective execution of the tasks. However, the inability of computers to assess tasks exercising more general skills, and therefore based on a more complex scoring rubric [22], apparently limits CAA software to processing this type of answer by simple techniques such as pattern-matching of input. Though technology is unlikely to be able to deliver such assessment in the near future, it remains possible that the application of CAA will be able to replace some assessment strategies in particular subject areas. The provision for group work is lacking, though some interesting systems have been developed based on peer assessment, for example OASYS [2].

4.10 Student autonomy

The rapid increase in student numbers over recent years has in part led to a change in the role of student and lecturer in the higher education environment. There has been a shift toward the student taking responsibility for their own learning and the lecturer providing the learning environment necessary and facilitating learning [5]. Indeed, self-evaluation is now regarded as an important life skill which should be encouraged as part of the educational process [24]. Self-evaluation is facilitated by a number of features in software. Provision for a student to view their progress in the context of the module — and ultimately the course — acts as both an indicator of weak areas and as a motivator. Some of the software reviewed, particularly those in the 'virtual learning environment' category, give this kind of overview, though this is as far as the promotion of self-evaluation goes.

Related to this is the idea of student-centred learning and the change in emphasis leading to students having more control over their learning than would be done traditionally [24, 5]. Student-centred learning might include the ability to undertake practice tests before undergoing an assessed one (or repeating tests until a given grade is attained). Even this relatively simple measure is rarely supported in CAA software. Student-centred learning also is facilitated by the openness of the assessment, for example minimal secrecy of the rubric and criteria for assessment [22], as well as ease of access. In these respects, much of the reviewed software makes good provision. By making the assessment more open and less hidden and subjective, the processes of learning are more visible; the student knows better the aims of the course and how best to undertake their own learning, and can determine whether concepts have been learned and which are yet to be learned [24, 4]. Computer-based assessment is often said to make the process of learning more visible to the student [22], and therefore encourages greater autonomy, though explicit efforts to make clear the learning process in the software are often neglected.

5 Developer and Institution

This section discusses the tools from the viewpoints of the software developer and the user institution, and the relationship between them.

5.1 CAA tool producers

With CAA tool producers either being companies or academic institutions, there is an association between the quality of a product and its cost. Academic products are generally free, or cheap, but are weak in support, where the level of input and continuing updates is concerned. Companies all charge for their products, but these tend to be more polished, are better maintained, and provide more consistent support. A number of products come from spin-off companies from academic departments, often giving some pedagogical relevance to the tool.

5.2 Quality of interface

The quality of the human-computer interface tended to be limited by some of the technologies used, with many suffering as a result of the use of (often non-standard) HTML code for presentation on the Web. Often products seemed to have an overdesigned or 'busy' interface, particularly those developed by commercial providers. Other products use HTML which gives the interface a somewhat dated look and feel. Some of the products using non-Web interfaces are better thought out.

5.3 Support

The quality of support provided is dependent on the source of the product. Commercial products offer a wide range of help from installation through to advice on question creation, but at a price. Academic institutions are not often in a position to provide more than informal support, but typically are friendly and willing to help via telephone and e-mail.

In many cases support consists of supporting the product only; associated learning materials are neither provided nor seem available and it is up to the course organiser to create them. A few products have an emphasis on material as well as software and both sell additional learning material modules as well as providing an infrastructure for the distribution of materials from textbook authors and other content providers.

5.4 Time and money costs

Computer-assisted assessment is often introduced out of a desire to counter the increasing time demands on staff while still delivering an acceptable educational experience. It should be considered, however, that this application carries costs of its own, both in terms of time and money.

The most time-intensive stage of using CAA is when it is being introduced — long-term gains in efficiency are often at a cost of short-term effort [10]. A major problem lies in the fact that materials need to be prepared for delivery by the computer. The lack of adoption of standards and the consequent lack of modularity leads to major difficulties in reusing materials, which means that materials need to be constructed from nothing, rather than being generated. There are also possible ‘hidden delays’ in installation, upgrade and upkeep procedures, deadline extension and mark adjustment on the computer, general system failures and collation of necessary statistics and their transfer to a central database in the absence of adequate reporting and database connectivity facilities.

As with most software solutions, there are also financial costs involved in applying CAA software. In addition to initial costs, some of the software requires third-part software. Many of the suppliers charge upgrade and support costs. There are also costs associated with the use of third-party assessment materials.

6 Ideas for further work

This section discusses possible future directions for the CAA project, firstly for the review exercise itself, and then topics for larger-scale CAA research which have been prompted by the LTSN work.

6.1 Review exercise

- Further investigation of the pedagogy of computer-assisted assessment, for example its role in learning.
- Survey work to gain a better idea of the requirements of Computer Science lecturers in higher education, to act as a set of new criteria against which to evaluate software.
- A study of how CAA is used in higher education, particularly computer science, to get a better idea of patterns of use, best practice and areas which software needs to address. Particularly, in which parts of the deployment of CAA software is the most time spent, and what measures might be taken to make the process less time-consuming?
- A survey and summary of anti-plagiarism software and techniques.
- Further investigation into the potential value of conditional questioning and adaptive testing and the degree to which it is supported in CAA tools.
- A study into the usages of CAA for summative and diagnostic testing and whether CAA is intrinsically primarily of use as a formative tool.
- An analysis of the extent to which the reporting capabilities of the CAA products fulfil the needs of the lecturers using them. This again would involve some requirements capture work.

- A requirements analysis of the security aspects of CAA software, and the establishment of an acceptable minimum level of security provision.

6.2 Topics in CAA research

- A rethink of the form of feedback given to users. At present feedback is limited in its form. How could feedback be extended to be of more use to the disparate roles of student and lecturer, from a pedagogical perspective?
- Investigation of the structure and reuse of CAA materials, and whether the IMS standard markup is appropriate for 'real world' use. This is strongly related to question banks and their organisation and use.
- A study into how computers and CAA might be better applied to group work, and the degree of support they already give. For example, do VLEs provide the pedagogically-sound groupwork support they promise?
- Extending work done in specific assessment situations, for example computer programming or other subject domains, or widening the range of question types, such as with the diagramming work.
- Making provision for deeper learning. The dialogue between the student and computer is often shallow in nature, and the tasks which the students are expected to do require only low-level thinking on Bloom's taxonomy. There is no provision for deeper learning in software — for example, there is a lack of constructed response questioning. To what extent could this be achieved?
- How could the principles of student-centred learning and student autonomy be better encouraged in CAA software? We identify one possibility in this report, namely by enhancing the visibility of the learning process, though it is possible that many other measures like this exist.

7 Conclusions

Developed assessment materials are made less reusable and less distributable by a general lack of provision of standardised or interchangeable formats. Furthermore, standards are slow to be developed and often lag behind the requirements and capabilities of the products. Developed materials are therefore neither portable nor do they facilitate sharing between lecturers and products (the question bank concept), reducing the motivation for, and value of, such materials.

Security often is an issue, particularly for those wishing to undertake assessment in a summative or diagnostic context.

For such a crucial part of the education-by-assessment process, feedback is not flexible enough in the tools. Feedback to students is often too concentrated on the responses themselves rather than at a test level.

The reporting capabilities of most of the tools is lacking. The great potential of using the computer as a analyser of class statistics, perhaps by finding problem students, areas of misunderstanding, or simply poor questions, is generally wasted.

The majority of CAA tools cannot be applied to general situations. They are typically intended for a particular setting in which CAA commonly takes place. This might be a simple quiz incorporated into Web-delivered materials, or a more capable summative assessment system, or a part of a VLE. In their limited application areas, most of the tools are capable, and in many cases have been proven to some extent in educational institutions.

There are some interesting CAA tools available, at various stages of development, with varying functionality and of varying quality. None of the products is a 'clear winner' and most lack features specific to the needs of education in Computer Science. Some products are strong in a few areas but none can be considered to meet every educational need; there are specific and fundamental shortcomings of the currently available tools and the approaches they take.

8 Disclaimer

Every effort has been made to ensure the accuracy of the information supplied in this report and the associated Website, and to the best of our knowledge and belief the report and the associated Website fairly describes the tools and packages which are referred to. Any error or omission can be reported to the authors and we will correct them as soon as possible

Neither the University of Warwick nor the LTSN Subject Centre for Information and Computer Sciences assumes any liability for errors or omissions, or for damages resulting from the use of the information contained herein and on the associated Website, and are not responsible for the content of external internet sites or any other other sources. The information contained herein and one the associated Website is provided on the basis that the reader will not rely on it as the sole basis for any action or decision. Readers are advised to contact the stated primary source or the product's originating institution before acting on any of the information provided.

This report includes some words and other designations which are, or are asserted to be, proprietary names or trade marks. Where the authors are aware that a word or other designation is used as a proprietary name or trade mark, this is indicated in the list below, but the absence of a word or designation from this list does not imply any judgement of its legal status:

Apache, Authorware, Ceilidh, Coursemaster, DES, EQL, GRE, IMS, Interactive Assessor, Java, JavaScript, Macromedia, Perception, Perl, Question Mark, TestPilot, Unix, Web@ssessor, Webassessor, WebCT, Windows, WebMCQ, XML.

9 Appendix: Summaries of the reviews

This appendix describes the tools reviewed, and briefly summarises the reviews, as presented on the Website in January 2002.

BOSS

BOSS is an open-source solution developed by the University of Warwick to handle online submissions and course management. It allows students to submit their work securely, and for staff to manage and mark assignmens. It has been developed with sound pedagogy in mind. It uses a role-based interface to keep track of the marking process of marking, moderation and publishing. The software stores much of its data in an SQL database, promoting scalability and integration with other systems.

The software is easy to use despite its comparative complexity, and has good security. A client/server approach is taken, and its lightweight clients which download the latest versions of the student and staff client ease large-scale deployment along with its platform independent Java implementation. Perhaps its main use is for formative testing; many of its features are suited to facilitating regular programming assignments. It is also useful for diagnostic and even controlled summative testing, though does not lend itself to ad-hoc self-testing due to the marking procedures it implements. Automatic testing works well with a few limitations, though active development will mean this area will become more flexible as development progresses. Documentation is a strength.

CASTLE

The CASTLE toolkit is a set of tools designed for tutors to create multiple choice based online tests. Emphasis is placed on ease of use throughout — no prior knowledge of any languages (such as HTML or Javascript) is required. CASTLE produces HTML form-based quizzes capable of handling multiple choice, multiple response and text-matching. It is comparatively basic in nature, its main strength being its ease of use and deployment.

ClydeVU/Miranda

CVU is a comparatively early attempt at a virtual university developed by Strathclyde University for use in a number of Higher Education institutions in Western Scotland. Its general aim to host Web-based teaching packages and discussions as well as assessments, though the assessment engine is more capable than many other virtual university solutions. In this evaluation we considered only the assessment engine, which is referred to as Miranda.

Though in some respects it is beginning to show its age, for most applications save summative testing in an examination environment, CVU is a useful and comparatively well-thought-out tool.

Coursemaster

Coursemaster is a client-server based system for delivering courses based around programming or diagramming exercises. It provides a solution for delivering coursework specifications, allowing students to develop programs and submit them, and the automated marking and monitoring of students' responses. Unlike much of the other software reviewed on this site, Coursemaster is tailored to the needs of teaching programming, and has been developed and used over several years.

Coursemaster is an updated version of the tried-and-tested (10 years) server/client-based Ceilidh system. It has a number of unique capabilities, for example the assessment of diagram-based work. It is robust and easy-to-use, particularly for the student. The lecturer is given a wide variety of statistics, though through a slightly dated interface. Marking is very flexible, though the built in tools do not achieve anything particularly innovative, but are still useful. Installation may be a problem area, depending on the platforms and configurations used at your institution. All aspects of implementation seem to have been thought through, and security is unusually good.

EQL Interactive Assessor

EQL Interactive Assessor is an application-based assessment system for Windows. Its emphasis is on tests taken against the clock in the classroom or at home with automatic analysis of results. The teacher uses a special application for designing tests and the student uses a similar one for taking it. The manufacturers emphasise security (separating lecturers' question banks and the resulting tests) and flexibility (freedom in question type and layout).

EQL Interactive Assessor is unusual in that its method of delivery is through a Windows-based application. However, the software is old and a new version is due soon. The modular, flexible tools makes the creation of tests easy, though the design of questions themselves is a little more involved and require a little more experience than with other tools. Ease of use from the student's point of view is generally very good, with a number of very minor drawbacks. The whole suite could not be tested due to the limitations on the demonstration suite. The software is perhaps best applied to self and formative testing, though under the right conditions could also be applied to summative testing.

Note that EQL Interactive Assessor has since been withdrawn from EQL's product range.

Netquest

The Netquest project was set up to investigate how 'question banks' for CAA might be set up to facilitate testing. The project is based around the TML — the tutorial modelling language – a superset of HTML intended to describe question data while keeping formatting information separate from its semantics. Software was also developed to process TML into a form suitable for delivery to a Web browser.

Netquest is a demonstrator of TML, a proposed language for assessment; it is not intended totally as a product for delivering tests. However, for this application it is certainly useful, though it takes more setting up than it possibly could. Delivery of tests is fast and clear, and basic analysis of individual students is possible. Though the software is not immediately applicable to summative testing, it works well as a regular formative or diagnostic tool.

Question Mark Perception

Perception is a package of five applications for designing and conducting surveys. The software is seen as something of an industry standard, and is certaintly one of the more developed products available. Its capabilities are wide-ranging and well thought out, its particular strengths being in its management of students, reporting of results, and the development and organisation of question data. Its question types are flexible, including many variations on the basic question types and others. The security is good and the database-centred approach gives the product robustness and scalability. Generally, a proven and fairly comprehensive CAA solution.

TACO

TACO is a Web-based application which allows lecturers to create ‘courseworks’ for students. TACO allows the lecturers to select question sets, made up of a variety of question types, and TACO marks and gives feedback.

TACO is a well-structured HTML assessment engine whose development was motivated by a strong requirements phase. It has a simple user interface and a good degree of flexibility in the the form of assessments, though its analytical capabilities are lacking. It is perhaps most suited to formative and perhaps diagnostic assessment, the lack of tight security being the main issue for summative assessment.

Test Pilot

Test Pilot is a moderately capable Web-based assessment engine. Lecturers author questions and build assessments from them, which can be automatically graded, with good feedback and analysis possibilities. Its main strengths lie in its relatively intuitive interface, which offers lots of help and is relatively easy to work with. Equal attention has been paid to all stages of the assessment process, from creation to assessment to statistical analysis. Being a commercial product, it is relatively expensive, especially the support fees. Test Pilot would be suitable for a formative or diagnostic level assessment application.

TRIADS

The name of the TRIADS project stands for ‘Tripartite Assessment Delivery System’, and is a collaboration between the University of Liverpool, the University of Derby and the Open University. The project’s focus is on learning outcomes and their influence on curriculum design. By understanding this it aims to produce a CAA system which allows a theoretically justified approach to formative and summative testing of knowledge, understanding and skills. The TRIADS system is a series of programs which run under Macromedia’s Authorware. The Authorware editor is therefore required to make use of the software.

The software presents an unrivalled number of question types and flexibility in question-setting, though the authoring for questions requires programming skills. Despite the TRIADS project’s noticable efforts, the Authorware environment is complex to use and introduces a steep learning curve. The authors’ plans for a better staff user interface will go a long way to fixing this problem. Taking a test if straightforward and good feedback can be specified, though there are some user interface issues which need to be considered here as well. Overall, a promising and flexible academic product in need of an easier user interface.

Webassessor

Webassessor (also written Web@ssessor) is an assessment, authoring and delivery system aimed primarily at industry, but which also claims to be as relevant for use in academia. Emphasis is placed on ease of use both in assessment creation and administration, while still allowing flexibility and ‘multimedia’ capabilities. The main application and area of collaboration with industry is seen as distance learning rather than incorporation in mainstream education.

Webassessor is a capable solution which caters for most applications. Its reporting capabilities mean that it has features somewhat more suitable for diagnostic testing. It has a reasonably good user interface and is easy to use, with good feedback capabilities. The basic question types are provided. Security has been considered and it has been proven in some limited commercial contexts. However, both the server software licence and the hosting service represents a significant considerable financial outlay.

Webassign

WebAssign is described as a ‘Web-based homework delivery, collection, grading and recording service’, aimed specifically at university lecturers. It has been, and is being, developed at the Department of Physics at North Carolina State University in the USA. As well as allowing lecturers to write their own questions, there is a emphasis on the development of question banks, in particular sets of questions from recognised textbooks.

WebAssign is significantly more capable in many respects than its competitors. It is particularly strong on the question-bank philosophy, feedback, randomisation, reporting and communication with students. Attention has been paid to allowing a flexible, modular approach to assignment building. After some experience with the mostly intuitive interface, the software can be used to create a good variety of assignments in a structured manner, though the limited range of question types may be a problem. Generally, a flexible and capable package, well-suited to the requirements of a university lecturer needing to build assignments.

WebCT

WebCT is sold as a course management system, claimed by the Website to be the most popular. Emphasis is on increasing communication between lecturers and students, though online assessment is one of its capabilities. Another emphasis is the delivery of online content, and the easy acquisition of delivered material from publishing companies rather than developing it 'in-house' by the lecturer.

The assessment component is concentrated on in the review, though broader considerations take in the more general application of the software. The assessment capabilities are limited to the point that WebCT may not be appropriate as an assessment tool only. Furthermore, WebCT is aiming to replace its assessment component with Question Mark in the future, using inter-program communication to achieve the integration.

WebCT is a comprehensive course management system, with relatively good CAA capabilities. Simply as a CAA tool WebCT would be expensive, cumbersome to manage, and over complex for assessment creation, reflecting its aim to fulfil a much wider role.

WebMCQ

WebMCQ provides a Web based service allowing quizzes focused on multiple choice questions to be created, edited and delivered though a Web browser. All administration and delivery of quizzes can be carried out from the WebMCQ site itself. Emphasis is on the provision of service, as opposed to software, and on ease of use.

WebMCQ provides a relatively easy to use system, with strong security and a good support service. It is slightly let down with some confusing interface features and a restrictive range of question types.

References

[1] Apache Software Foundation. Apache website. `http://www.apache.org/`.

[2] Abhir Bhalerao and Ashley Ward. Towards electronically assisted peer assessment: a case study. *Alt-J — Association for Learning Technology Journal*, 9(1):26–37, 2001.

[3] Benjamin S. Bloom and David R. Krathwohl. *Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. Handbook I: Cognitive Domain.* Longman, 1956.

[4] David Boud. *Assessment and Learning: Contradictory or Complimentary?*, pages 35–48. Staff and Educational Development Series. Kogan Page, 1995.

[5] Don Charman. *Issues and Impacts of using Computer-based Assessments (CAAs) for Formative Assessment*, pages 85–94. Staff and Educational Development Series. Kogan Page, 1999.

[6] Clyde Virtual University. About CVU. `http://cvu.strath.ac.uk/admin/cvudocs/`.

[7] James R. Dalziel and Scott Gazzard. Assisting student learning using web-based assessment: An overview of the webmcq system. `http://www.webmcq.com/public/pdfs/ovrview.pdf`.

[8] Drake Kryterion. Webassessor. `http://www.webassessor.com/webassessor.html`.

[9] Educational Testing Service. Graduate records examination website. `http://www.gre.org/`.

[10] Jen Harvey and Nora Mogey. *Pragmatic Issues When Integrating Technology into the Assessment of Students*, pages 7–20. Staff and Educational Development Series. Kogan Page, 1999.

[11] IMS Global Learning Consortium. *IMS Question & Test Interoperability: ASI Selection and Ordering Specification*, October 2001.

[12] IMS Global Learning Consortium. *IMS Question & Test Interoperability Specification*, October 2001.

[13] Institute for Learning and Research Technology. Netquest. `http://www.ilrt.bris.ac.uk/netquest/`.

[14] M. S. Joy and M. Luck. The BOSS system for on-line submission and assessment. *Monitor: Journal of the CTI Centre for Computing*, pages 27–29, 1998.

[15] Mike Joy, Simon Rawles, and Michael Evans. Computer assisted assessment tools review exercise. `http://www.dcs.warwick.ac.uk/ltsn-ics/resources/caa/reviews/`, 2001.

[16] Learning Technology and Research Group. Coursemaster. `http://www.cs.nott.ac.uk/CourseMaster/`, 2001.

[17] EQL International Ltd. EQL. `http://www.eql.co.uk/`.

[18] LTSN-ICS. LTSN-ICS Website. `http://www.ics.ltsn.ac.uk/`.

[19] Don Mackenzie. Triads (tripartite interactive assessment delivery system) manual. `http://www.derby.ac.uk/assess/manual/tmanual.html`.

[20] Macromedia. Authorware. `http://www.macromedia.com/software/authorware/`, 2001.

[21] Misc. Perl website. `http://www.perl.com/`, 2002.

[22] Malcolm Perkin. *Validating Formative and Summative Assessment*, pages 55–62. Staff and Educational Development Series. Kogan Page, 1999.

[23] Question Mark. Question Mark. `http://www.questionmark.com/`.

[24] Kay Sambell, Alistair Sambell, and Graham Sexton. *Student Perceptions of the Learning Benefits of Computer-assisted Assessment: A Case Study in Electronic Engineering*, pages 179–192. Staff and Educational Development Series. Kogan Page, 1999.

[25] M. A. Sasse, C. Harris, I. Ismail, and P. Monthienvichienchai. *Support for Authoring and Managing Web-based Coursework: The TACO Project*. Springer Verlag, 1998.

[26] Leith Sly and Léonie J. Rennie. *Computer Managed Learning as an Aid to Formative Assessment in Higher Education*, pages 113–120. Staff and Educational Development Series. Kogan Page, 1999.

[27] Sun Microsystems. Java website. `http://java.sun.com/`.

[28] Christian Süß and Burkhard Freitag. Learning material markup language: LMML. `http://www.ifis.uni-passau.de/publications/reports/ifis200103.pdf`.

[29] Athanasios Tsintsifas. Diadalos. `http://www.cs.nott.ac.uk/~azt/daidalos/`.

[30] Dave Whittington. *Technical and Security Issues*, pages 21–28. Staff and Educational Development Series. Kogan Page, 1999.