ODYSSEY

Borrower: MOU

Lending String: *EXW,EXW,OKT,KSU,IAC

Patron: White, Philip

Journal Title: SIGCSE bulletin inroads.

Volume: 35 **Issue:** 4
Month/Year: 2003 **Pages:** 107-123

Article Author:

Article Title: Carter, Janet et al; How shall we assess this?

ILL Number: 81849737

Call #: QA76.27 .A79a
Location: LL

Mail

Charge
Maxcost: $20.00IFM

Shipping Address:
MISSOURI STATE UNIVERSITY
MEYER LIBRARY ILL
901 S NATIONAL
SPRINGFIELD MO 65897

Fax: 417-836-4538
Email:

How Shall We Assess This?

Janet Carter
Computing Laboratory
University of Kent
Canterbury, CT2 7NF, UK
J.E.Carter@kent.ac.uk

John English
School of Comp and Math Sciences
University of Brighton,
Brighton, BN2 4GJ, UK
J.English@brighton.ac.uk

Kirsti Ala-Mutka
Inst Software Systems
Tampere University of Technology
Finland
Kirsti.Ala-Mutka@cs.tut.fi

Martin Dick
School of CS and SE
Monash University
Victoria 3145, Australia
Martin.Dick@csse.monash.edu.au

William Fone
School of Computing
Staffordshire University
Stafford, ST18 0DG, UK
W.Fone@staffs.ac.uk

Ursula Fuller
Computing Laboratory
University of Kent
Canterbury, CT2 7NF, UK
U.D.Fuller@kent.ac.uk

Judy Sheard
School of CS and SE
Monash University
Victoria 3145, Australia
Judy.Sheard@csse.monash.edu.au

ABSTRACT

Increased class sizes are forcing academics to reconsider approaches to setting and marking assessments for their students. Distributed and distance learning are creating some of the biggest changes. Some educators are embracing new technologies but others are more wary of what they do not know. In order to address this issue it is first necessary to investigate the types of assessment currently in use and the perceptions that are held by academics with and without experience of the new technologies that are becoming available.

In this paper we present the findings of an international survey of Computer Science academics teaching a variety of topics within the discipline. The findings are split into two sections: a snapshot of current assessment practices and an analysis of respondents' perceptions of Computer Aided Assessment (CAA). Academics' opinions about the advantages and disadvantages of CAA are split in line with level of experience of using such techniques. Those with no experience of CAA suggest that it cannot be used to test higher-order learning outcomes and that the quality of the immediate feedback is poor; these negative opinions diminish as experience is gained.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education – *computer science education*

General Terms

Human Factors.

Keywords

Assessment, Computer Aided Assessment, Plagiarism.

1 INTRODUCTION

With increasing class sizes in educational establishments worldwide, the practice of assessment is becoming a problematic issue; increased numbers make it more difficult to assess student attainment. If assessments are graded manually, educators must either set fewer assessment tasks or resign themselves to a greatly increased marking load. In order to cope with increasing student numbers automated assessment is becoming increasingly important in many courses. The number of papers related to the topic that have been presented at ITiCSE conferences in recent years [e.g. 21, 30, 31, 41, 49, 65] reflects this increasing interest. Automated assessment can save time and human resources but its adoption must be pedagogically sound. It is a widely held belief that on-line teaching and learning will be the savior of the educational system. Current research suggests that students initially prefer to be taught by a human, finding a machine too impersonal and a disincentive to learning; but that once the initial stages are completed a machine is an acceptable teacher [54]. There are, however, some important issues to consider: How do you tell that the person taking an on-line examination is the person they should be? How do you tell that they aren't receiving help? There are projects investigating the effective use of such techniques [3, 60, 74], but they are still in their infancy.

1.1 Assessment Principles

All assessments should follow sound educational principles and the most widely adopted epistemology within the CS arena appears to be that of constructivism. Constructivist principles of educational development suggest that:

- Students are active participants in the process of their own learning

- All learning takes place within a context – usually the classroom – where shared meanings and understandings can be created
- Students require time to reflect upon the work that they are doing
- Students require the space to be allowed to make mistakes and to learn from these mistakes

Ben-Ari [5] notes that learning should be active, not passive; students are being called upon to build mental models of abstract conceptions of how computers work, the nature of variables in programming, and so on. Computer Science in particular is a deeply practical subject, and providing as much opportunity for practical work as possible will help to develop students' understandings of the principles behind the subject, as long as this work is undertaken in concert with human assistance to overcome misconceptions and refine mental models. The downside, for educators teaching large numbers of students, is that each piece of practical work needs to be marked.

1.2 What is CAA?

In this paper, Computer Aided Assessment (CAA) is defined as any activity in which computers are involved in the assessment process as more than just an information storage or delivery medium. Computers can be used in various ways to support students' learning and course processes. They can provide numerical marking and feedback in both textual and visual formats. In distance and web-based education CAA is often a natural extension to the course. It can also offer many possibilities for campus-based education, especially when large classes are involved. It also provides an easy mechanism for statistical analysis of assessment results at a later date.

Computer Science as a subject area is well positioned to benefit from automated assessment. Those who teach the subject can often use their expertise to develop systems that help to reduce their workload without compromising student learning.

1.3 Cheating and Plagiarism Issues

The issue of cheating and plagiarism is an increasing problem for academics. A recent UK survey suggests that the incidence is actually much higher than many academics realize [20]. Whilst the interviewing of all students to ensure that they can reproduce the work they submit [38] may be practical when students produce only a few pieces of work, it does not scale to situations where students produce regular weekly solutions.

On-line plagiarism detectors such as JPLAG [50] and MOSS [61] already rely upon electronically submitted work, and it is logical to consider automated marking of such submissions. Another potential benefit of automated assessment techniques is that they can assist in avoiding plagiarism by presenting students with randomly chosen problem sets or individualized exercises. Although students can collaborate on solving the problems they have been set, they are no longer able to submit verbatim copies of solutions obtained by others.

2 A SNAPSHOT OF CURRENT ASSESSMENT PRACTICES

In order to obtain the perceptions and experiences of academics a web-based survey was created (see Appendix). The working group participants then advertised the survey as widely as possible to CS academics.

The responses are not necessarily representative of all CS academics; they are necessarily skewed by the means of advertising and the nationalities of the authors. Within the UK the survey was advertised via the LTSN-ICS (Learning and Teaching Support Network for Information Computer Sciences) mailing list and the UK CAA mailing list. US responses were solicited via the SIGCSE mailing list.

In Finland the questionnaire was advertised via the Virtual University Network, on the IT-PEDA University Network and on the Finnish Society for Computer Science newsgroup. Australian responses were gathered from members of five of the six schools of the Faculty of IT at Monash University along with members of their Computing Education Research group, paper presenters from the ACE (Australasian Computing Education) 2003 conference, and users of the Faculty of IT intranet at Queensland University of Technology in Brisbane. It was also advertised multi-nationally on the KIT e-learning mailing list.

Statistical analysis, including Mann-Whitney U-tests and Kruskal-Wallis tests, of the numerical aspects of the data has been performed, where appropriate, at a 5% level ($p < 0.05$).

2.1 The Respondents

The responses to the questionnaire are not analyzed by gender or geographical area. Table 1 does, however, provide a breakdown of responses by country of origin, gender and topic taught; this merely provides a context for the conclusions that are drawn. Unsurprisingly the overwhelming majority of responses originate from the countries represented by the authors. 35% of replies are from females, and 25% are from those who teach programming.

2.2 Assessment Regimes

A wide variety of assessment techniques are used by CS academics; many using more than one technique per course. Table 2 shows the overall proportion of respondents using different types of assessment; the sum is greater than 100% as it is often the case that assessments of different types are set within the same course. An unusual result emerged: breaking the figures down by topic taught reveals that five respondents assess some aspect of their programming course by means of an essay.

Table 3 shows a breakdown of submission mechanisms and marking techniques for each type of assessment. It is notable that practical work is the only kind of assessment task for which a majority of academics use electronic submission.

Table 1: Nationality, gender and topics taught

Topic	Country										Totals		
	Australia		Finland		UK		USA		Other				
	M	F	M	F	M	F	M	F	M	F	M	F	Total
Programming	1	3	6		8	2	6	3	2		23	8	31
Mathematics	1				1						2	0	2
Computing theory						1	1	1			1	2	3
Data structures/algorithms		1	3		2		1	2	1		7	3	10
Databases			1	2	1	3	1	1		1	3	7	10
Information systems	1		1		3	1			1		6	1	7
Systems analysis	3	1			2	2					5	3	8
HCI		2	1	1	1	3				1	2	7	9
Networks	1			1	3	3					4	4	8
WWW			2		3						5	0	5
Hardware/computer architecture			1	1	2	1		1			3	3	6
Operating systems	1						4				5	0	5
Real-time systems					1						1	0	1
Distributed systems					1	1					1	1	2
Design			1		1	1					2	1	3
Ethics							1				1	0	1
IT		1	1		1	1				1	2	3	5
Compilers					1						1	0	1
Ubiquitous computing			1								1	0	1
Software Engineering	2		1								3	0	3
Data Mining	1										1	0	1
Totals	11	8	19	5	31	19	14	8	4	3	79	43	122
	19		24		50		22		7				

2.3 Taxonomy of CS Assessment

In order to understand what is going on in assessment, it is necessary to know:

- The subject matter being assessed
- The level of study of the learner (first year, honors year, masters etc)
- The instrument being used (essay, closed test, practical work and so on)
- The cognitive attainment levels that the students are expected to achieve

Our survey, therefore, asked respondents about cognitive attainment, ideally in a way that was simultaneously short, comprehensible and comprehensive. This is not easy.

A number of taxonomies of competence in the cognitive domain have been produced over the past fifty years. By far the best known is that of Bloom *et al* [10]; a committee that examined a large bank of assessment instruments produced it in the USA in the 1950s. More recently, the Structure of the Observed Learning Outcomes (SOLO) taxonomy and the reflective thinking measurement model have been proposed and a revised, two-dimensional version of Bloom has been developed. Chan *et al* [15] report that these taxonomies are closely related and that each could complement the weaknesses of others. The levels in the first three of these are presented in Table 4.

Bloom's taxonomy is by far the most widely used of those described here; in fact some universities and colleges require course designers to check that they are assessing at every one of Bloom's levels. The lowest level, knowledge, is also known as recall or remembering and is concerned with the reproduction of previously learned materials. Comprehension is also referred to as understanding and assesses the ability to explain and summarize materials. Application is concerned with using learned material in new situations. Together, knowledge, comprehension and application are considered to be the lower levels of the cognitive domain and the ones most frequently tested using multiple-choice questions.

Analysis, synthesis and evaluation form the upper levels of the cognitive domain. Bloom placed synthesis and evaluation on a par and there has been much subsequent debate about their ordering. Anderson and Krathwohl's revision of Bloom's taxonomy places synthesis at the highest level [2] and it is generally held that it, uniquely, cannot be assessed through multiple-choice and short answer questions [36].

Table 2: Use of different assessment types

	Essay	Other written exercise	Practical work	Closed-book exam	Open-book exam	In-class test	Presentation	Other
Proportion	25%	55%	74%	57%	16%	31%	34%	17%

Table 3: Marking and submission techniques by assessment type

		Essay	Other written exercise	Practical work	Closed-book exam	Open-book exam	In-class test	Presentation	Other
Submission mechanism	Manual	21	46	46	63	10	28	28	6
	Electronic	13	32	54	10	8	11	22	12
Marking techniques used	Manual	26	47	55	58	11	27	24	5
	Pt manual, pt electronic	4	14	26	10	5	6	6	4
	Electronic	1	5	11	5	4	6	5	6
	Peer assess	3	10	16	1	1	1	12	2
	Interview	2	4	16	0	1	1	10	5

Table 4: Taken from Chan *et al* [15]

Bloom	SOLO	Reflective Thinking
Knowledge	Prestructural	Habitual Action
Comprehension	Unistructural	Thoughtful action
Application	Multistructural	Reflection of content
Analysis	Relational	Reflection of process
Synthesis	Extended Abstract	Critical reflection
Evaluation		

Table 5: Cognitive level by assessment style

(%)	Essay	Other written exercise	Practical work	Closed-book exam	Open-book exam	In-class test	Presentation	Other	Total
Remembering	21	20	24	81	40	56	15	33	38
Understanding	96	80	73	91	85	92	74	73	82
Application	71	64	95	57	60	53	68	67	70
Problem solving	54	67	84	60	55	56	32	47	63
Evaluation	79	43	47	37	45	36	68	53	48
No. assess. types used	24	61	85	68	20	36	34	15	343
Mean levels per type	13.4	4.5	3.8	4.8	14.3	8.1	7.5	18.2	0.9

Respondents indicated the cognitive levels they assess with each type of assessment used. Table 2 shows the incidence of each assessment type across all courses, whilst Table 5 breaks these down by cognitive level. A large number of assessments test 'problem solving', i.e. a combination of analysis and synthesis, so we use this term rather than analysis and synthesis separately.

There are few surprises in Table 5, for example it shows heavy use of closed book examinations to test remembering and concentration on the assessment of evaluation through essays and presentations. In a CS context, the heavy emphasis on problem solving in closed-book examinations and in-class tests may be a plagiarism avoidance technique.

2.4 Plagiarism Issues

Respondents were asked to indicate the level of problem that they experience with plagiarism for each type of assessment task listed in the survey. The problem level was measured on a four-point scale ranging from not a problem, through minor problem and moderate problem to major problem. Respondents could also indicate that they didn't know. Table 6 details the results of the survey. The mode and quartiles are also shown to indicate the problem level of plagiarism for each kind of assessment task. As can be seen, essays, other written exercises and practical work are the assessment types that provoke the highest levels of concern from respondents with regard to levels of plagiarism. This is worthy of further investigation; the survey data has been analyzed to determine which, if any, assessment factors influence the respondents perceptions of plagiarism as a problem (Assessment task 'other' was not specified in the survey, so no further analysis was performed).

- **Exercise Type.** The four types of exercise (fixed set of questions, choice of questions, individually tailored questions and group work) were examined using the Mann-Whitney U test for each kind of assessment task to determine whether or

not there were any statistically significant differences in the level of plagiarism between those respondents who used the exercise type and those respondents who did not use the exercise type. The only statistically significant difference was found for those respondents setting practical work using a fixed set of questions. Examination of the median and mode however revealed no difference between the groups (median and mode were minor problem for both groups). This indicates that the difference is probably not practically significant. Overall, the type of exercise has little impact on the perceived level of plagiarism.

- **Aspects of Learning Covered.** Aspects of learning were examined in two different ways, firstly by comparing each type of assessment task with the respondent's perception of a problem with plagiarism against whether the assessment was attempting to cover one of the aspects of learning (remembering, understanding, application, problem-solving, evaluation). A Mann-Whitney U test found that only one of the forty options was statistically significant when a closed-book examination was assessing understanding. In this case the mode and median for both groups were found to be identical; this indicates that, on a practical level, this aspect of learning/assessment type has no effect upon perceived levels of plagiarism.

  Secondly, the data was recoded to indicate the highest aspect of learning for each kind of assessment. This was then examined with a Kruskal-Wallis test to compare the perceived level of plagiarism for the different levels of the aspect of learning. No statistically significant differences were found for any of the kinds of assessment.

  Overall, the aspects of learning covered have no real effect upon the perceived level of plagiarism.

- **Submission Mechanism.** The impact of the mechanism of submission (manual or electronic) for each kind of assessment was examined using a Mann-Whitney U test. For most types of assessment, there were no statistically significant differences on the level of plagiarism between manual and electronic submission.

  For closed book examinations and in-class tests, the use of electronic submission did result in a small increase in the median for perceived problem level of plagiarism, but did not affect the mode. Table 7 shows the differences in median.

  This indicates that for these two kinds of assessment task, electronic submission can cause a small increase in the perceived level of plagiarism amongst students. This may be due to the increased ease of copying an assessment task that is in an electronic format. Regardless of the reason for the increase, the actual level of increase does not raise large concerns.

  Overall, the submission mechanism has no major effect upon the perceived problem level of plagiarism amongst the respondents.

- **Marking Techniques Used.** Respondents reported the use of a variety of marking techniques: manual, part manual/part electronic, fully electronic, peer assessed, interview. These were then compared with the perceived level of plagiarism problem for each type of assessment in two different ways. Firstly, each type of assessment was compared using a Mann-Whitney U test for users and non-users of the marking technique. This found that only one marking technique/assessment type combination had statistically significant differences; this was the use of part manual/part electronic marking with a closed-book examination. In this case, the median increased from not a problem to minor problem. The modal responses for those who used part manual/part electronic and those who didn't use it were the same however (not a problem). This indicates that while such a combination may cause some slight problems with plagiarism, they are not particularly worrying.

  The second method by which marking technique was examined involved recoding the data for manual, part manual/part electronic and fully electronic for each assessment type into one variable recording the extent of electronic marking. A Kruskal-Wallis test showed no statistically significant differences between the three levels of electronic marking for any of the types of assessment except for closed-book examination, when looking at the perceived problem level with plagiarism. For this type of assessment, there was a small increase in median for part manual/part electronic when compared to wholly manual or electronic.

  The median for part manual/part electronic was mid way between not a problem and minor problem as compared to not a problem for the other two groups. The modal response for all three groups was 'not a problem', which combined with the small size of the increase in median indicates that the extent of electronic marking has little effect on plagiarism in closed-book examinations. Overall, it can be concluded that the marking techniques used for an assessment task will have little if any effect on the perceived level of plagiarism.

Table 6: Plagiarism problems for kind of assessment task

	Plagiarism level for ...						
	Essay	Other written exercise	Practical work	Closed-book exam	Open-book exam	In-class test	Presentation
N	37	66	88	70	25	41	37
Mode	Minor	Minor	Minor	Not	Not	Not	Not
25th percentile	Not	Minor	Not	Not	Not	Not	Not
50th percentile	Minor	Minor	Minor	Not	Not	Not	Not
75th percentile	Moderate	Moderate	Moderate	Not	Minor	Minor	Minor

Table 7: Impact of Electronic Submission on Plagiarism

	Closed-Book Exam		In-Class Test	
	Doesn't Use	Uses	Doesn't Use	Uses
Median	Not a problem	Minor problem	Not a problem	Midway between not and minor
Mode	Not a problem	Not a problem	Not a problem	Not a problem

- **Teaching Experience and Course Level Taught.** Neither of these factors had any statistically significant differences using a Kruskal-Wallis test for any type of assessment task when examining the perceived level of problem with plagiarism.

2.5 Perceptions of CAA

The respondents' perceptions of CAA are shown in Table 8. This shows that there is strong agreement with the proposition that CAA reduces marking time and improves the immediacy of feedback to students, and some agreement that CAA offers greater objectivity in marking and that it allows students to work at their own pace and more flexibly. There is also some disagreement with the propositions that CAA has fewer security risks than manual assessment, that it is more time-consuming than manual assessment, that it improves the quality of feedback to students, and that it disadvantages special-needs students. Most respondents were broadly neutral on the issues of the possibility of testing higher-order learning using CAA and that the use of CAA makes students more anxious.

The questionnaire also included space for comments on the perceived advantages and disadvantages of CAA. Analysis of these comments shows that the primary perceived advantages of CAA are:

- The time saved in marking, particularly with large groups
- The immediacy of feedback, although not necessarily the quality of feedback (however, one respondent noted that 'poor feedback from CAA is preferable to detailed manual feedback which arrives weeks after the work was completed')
- The objectivity and consistency of marking
- That CAA frees students to work at their own pace

These comments broadly agree with the numerical findings presented in Table 8. Some respondents also noted that the speed of marking allows more frequent assessments, both formative and summative, to be conducted; again, this is particularly true for large groups of students. Several respondents commented that having student submissions available in electronic form allows post-hoc analysis of submissions to be performed, although this is of course true of any form of electronic submission system whether the submissions are marked electronically or not.

A number of other drawbacks were widely reported. Many respondents were concerned about reliance upon technology, both in terms of downtime and of system failures that might lose or corrupt data. Issues of security and plagiarism also caused widespread concern, particularly in distance learning environments where it is difficult, or impossible, to verify the identity of the person undertaking the assessment. This is to some extent offset by the recognition that some CAA systems allow questions to be individually tailored, either by presenting different students with different sets of questions or by parameterizing questions with different values. (One of the authors has used this approach for in-class tests, where each student is given the same questions but with different values in each question. On one occasion a student submitted of set of completely incorrect answers, but further investigation showed that the answers would have been correct if he had been given the same set of questions as the student next to him, and because of the demonstrable plagiarism he failed not only the test but the entire course.)

Some interesting differences emerge when comparing the attitudes of those who have not used CAA with those who have, either a little or a lot. Table 9 shows the proportions of respondents having experience of using Learning Environments and CAA.

The respondents as a whole were broadly neutral on the question of using CAA to test higher-order learning, but when the data is correlated against the respondents' experience of CAA the picture is quite different.

Respondents with more experience of CAA tend to believe more strongly that it could be used to test higher-order learning, but those without experience are more skeptical. This would appear to reflect the widespread myth that CAA can only be used for multiple-choice tests; many of those who have little experience with CAA have used commercial learning environments such as Blackboard [9] and WebCT [77] which only support multiple-choice tests, and this reinforces the acceptance of the myth. Figure 10 shows a more detailed breakdown of the responses to this question.

A Kruskal-Wallis test shows significant statistical differences in the views of respondents on objectivity, flexibility and the immediacy and quality of feedback, with those having more experience with CAA consistently believing more strongly that CAA had significant benefits in each of these areas. Table 11 shows the median and mode for each category of respondent in each of these areas.

- **Usage of CAA.** The level of problem with plagiarism respondents had found with the different kinds of assessment task was compared with the usage of CAA to determine if there was any relationship between plagiarism problems and CAA. Usage of CAA was measured on two scales, firstly whether they had used CAA at all or not. A Mann-Whitney U test found that there were no statistically significant differences in the two groups for any of the assessment tasks. The second scale measured usage of CAA on three levels (not at all, used a little, used a lot). A Kruskal-Wallis test found no statistically significant differences in the three groups. This indicates that the usage of CAA has no noticeable effect on the perceptions of respondents about the level of plagiarism for assessment tasks.

- **Security of CAA.** Kruskal-Wallis tests found that there was no statistically significant difference in perceptions of the security of CAA amongst respondents for any of the following factors:
  - Extent of electronic marking
  - Course level taught
  - Level of teaching experience of the respondent
  - Usage of CAA at all
  - Usage of CAA at varying levels (Not at all, Used a little, Used a lot)

Table 8: Opinions regarding aspects of CAA

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Total
CAA has fewer security risks than manual assessment	5%	32%	40%	15%	7%	114
CAA is more time-consuming than manual assessment	10%	35%	28%	21%	5%	113
CAA reduces marking time	2%	4%	12%	50%	32%	114
It is possible to test higher-order learning using CAA	4%	27%	33%	33%	3%	114
CAA offers greater objectivity in marking	3%	19%	25%	45%	9%	113
CAA allows students to work at their own pace and more flexibly	1%	10%	26%	45%	18%	114
The use of CAA makes students more anxious	2%	25%	54%	18%	2%	112
CAA improves the immediacy of feedback to students	0%	4%	5%	64%	26%	114
CAA improves the quality of feedback to students	5%	29%	42%	19%	5%	111
CAA disadvantages special-needs students	4%	28%	52%	13%	2%	113

Table 9: Actual CAA experience

	No	Yes		Total
		A little	A lot	
Do you have any experience (past or present) of using online learning environments?	36%	64%		116
Have you ever used computer-aided assessment?	38%	41%	21%	117
Is computer-aided assessment used within your department?	26%	59%	15%	117

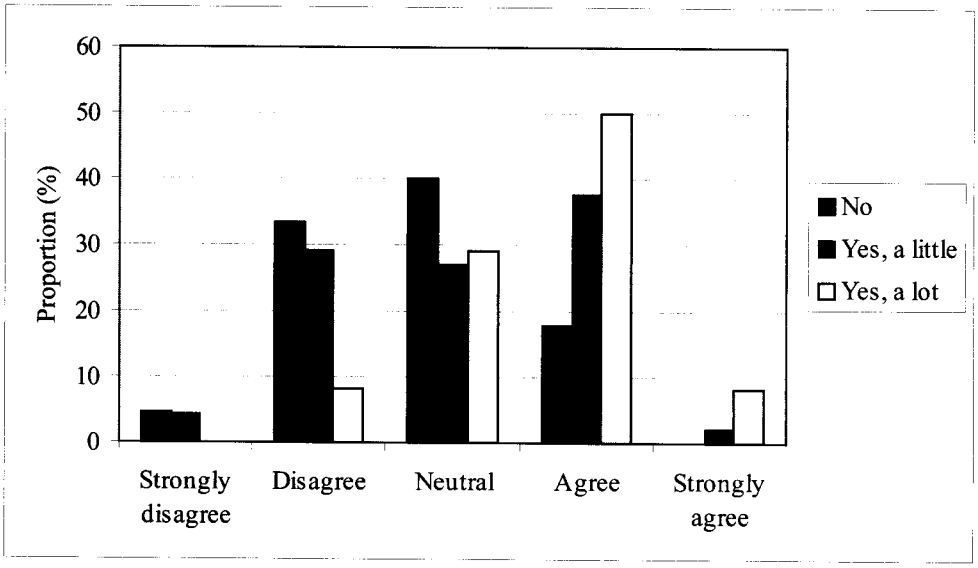

Figure 10: Higher order learning

Table 11: Medians and modes for perceptions with statistically significant group differences

Used CAA?	N	Can test higher-order learning	CAA offers greater objectivity	More flexible for students	Immediacy of feedback	Quality of feedback
No	43	disagree/disagree	disagree/neutral	disagree/disagree	neutral/neutral	strong dis-dis /strongly dis
Yes,a little	48	disagree/neutral	neutral/neutral	neutral/neutral	neutral/neutral	disagree/disagree
Yes, a lot	23	neutral/neutral	neutral/neutral	agree/agree	agree/agree	neutral/neutral

3 COMPUTER AIDED ASSESSMENT

Many academics claim to be interested in the issues surrounding the adoption of CAA and the experiences related here encapsulate many aspects of their expressed concerns and experiences. In this section we present two stories that exemplify some issues of concern to many.

Andrew Solomon from the University of Technology in Sydney [70] is typical of many academics in having over 250 first year students to teach each year. One of the things he is expected to teach them is basic competence with Unix. In previous years the students were provided with a book and sample questions in preparation for the manually marked examination at the end of the course. Marking these examinations consumed a disproportionate portion of the staff time devoted to the subject.

This year Andrew wrote a program to semi-randomly generate Unix oriented tasks for the students and then determine whether or not the students had accomplished them. This means that the students receive their mark immediately on completion of the examination. It also means that staff time, which was previously devoted to marking, is now spent on teaching.

The problems Andrew experienced with this include:

- The expense of generating questions
- The restricted type of questions that can be electronically marked
- The rather unforgiving marking of which a computer is capable

This was a new scheme and Andrew did not want his students to be disadvantaged by the new system, so he decided to evaluate this new assessment regime. Andrew asked two experts to interview 20 students and attempt to assess their general Unix competence, without too close reference to the curriculum. He compared the experts' opinions of the students' abilities with the marks they attained in the examination. The experts felt that the students all had approximately the same level of competence, but the tests appeared to discriminate between the students; the distinction between students that was detected by the tests is likely to be no more than an indicator of the amount of time students devoted to studying for the examination rather than a difference in general understanding; this is common in non CAA examinations also. The experiment was time consuming and stressful, but ultimately successful.

John English from the University of Brighton provides another example. He regularly uses online practical programming examinations for first-year students [31]. They are conducted under normal closed-book examination conditions; however, many of the questions involve writing program fragments that are marked electronically after the end of the examination. The marking is performed by embedding the answers into a test program, which is then compiled and run against several sets of test data. All code is initially marked electronically, and any answers which fail to compile or which fail to process any of the sets of test data successfully are flagged for manual inspection. In many cases, manual inspection reveals that the answers submitted are 'nearly right', with simple errors such as missing semicolons causing the compilation to fail. Marks are awarded manually for such answers. This CAA tool does not differentiate between the students that did not obtain completely correct solutions due to its simplicity; it cannot distinguish between 'nearly correct' and 'wrong'.

3.1 Pros and Cons of CAA

Using CAA can provide an increased amount, quality and variety of feedback for students as well as providing new assessment tools for educators. The benefits include speed, consistency, and availability 24 hours per day. Such tools can support a variety of different learning habits and needs amongst the students. Well-written CAA tools can allow teachers more time for advanced support, assessment and education design tasks rather than basic support. Digitally formatted assessments can also offer possibilities for reuse and resource sharing between teachers.

There are, however, disadvantages with the use of CAA. The set-up phase for a new system or new assessments requires more time and resources than the development of traditional assessments. Despite careful design work there can be errors or technical problems that prevent the assessment system from working as expected. This, especially on courses that use computer-based systems heavily, can demotivate students or even hinder their learning.

3.2 Types of CAA in CS Education

This section presents a summary of existing CAA approaches used within CS education, describing their connections to Bloom's taxonomy [10]. Respondents report that they mostly use electronic marking to test for higher cognitive levels of learning, i.e. problem solving and evaluation. The survey did not contain specific questions about the types of CAA respondents used, but the large numbers of answers from programming teachers setting practical work suggests that a large number of respondents use at least partially automatic marking when assessing programming tasks.

- **Multiple-choice questions**. These can be used to assess many cognitive levels. Whilst poorly constructed multiple-choice questions assess nothing but logic in answering, well-designed questions are a good tool for assessing knowledge and comprehension [56], or even higher cognitive levels [28]. Multiple-choice questions are often offered by learning environment software, such as WebCT and Blackboard [77, 9]. The questions can take several forms; visual, multiple

choice, combination, gap filling [22, 35, 51, 63]. Generating different permutations of question sets or different values for question variables means that tests can be individualized to prevent cheating and memorizing, and also to increase complexity [32, 71]. In addition, some tools have been developed to assess the comprehension of a specific topic, e.g. pointers in C++ [53].

- **Textual answers**. Textual assessment often requires students to write short answers or essays. Again, dependent upon the skill and experience of the question designer, these can be used to test both lower and higher order learning skills. There are several approaches to supporting automatic assessment of free text answers. Some learning environments and question systems base the assessment on either direct text comparison or regular expressions, and these support short answer questions. There are, however, many more sophisticated approaches to the assessment of textual content [12, 16, 59].

- **Programming assignments**. Practical programming is the most common way of evaluating students application and problem-solving skills. When aiming to assess lower level application skills, CAA can be helpful in assessing a student's ability to produce correctly functioning classes or functions, without the need to be able to produce a whole program [3, 11, 65]. CAA can also be used to test the analysis level. For example, by providing students with a program containing bugs and requiring them to correct it into a program that compiles and functions correctly [31]. There are also several integrated systems and tools for assessing complete programming assignments. The most common metrics are program functionality, complexity, style and efficiency [41, 74]. Some systems also provide additional metrics, such as test data coverage [57] and programming skill [35]. In addition to integrated tools there also exist independent, aspect specific, tools, for example: dynamic memory management or coding conventions in C++ [1]; spreadsheet and database testing [67].

- **Visual answers**. Visualizations can be used for assessing both understanding and application, e.g. with data structures and algorithms [52]. Translating programs into flowcharts can be used to assess a student's understanding of program flow [11]. It is also possible to assess the problem-solving level by using CAA for design diagrams [43].

- **Peer assessment**. With large numbers of students, it would be very difficult to arrange organized peer evaluation without the aid of computers. With electronic submission it is possible to automatically randomize peer assessors, to anonymize answers, to give and store the marks and to compare their consistency; peer assessment strategies can be greatly developed. Some approaches to computer assisted peer assessment have already been introduced [34, 76]. Peer assessment can also be utilized for assessing students' evaluation skills, by comparing their assessment to their teacher's assessment or to each other's [23].

It can be deduced from the survey that it is more common to combine CAA use with manual marking than to rely on it totally for fully electronic marking; for example either flagging all cases that are not clearly correct for human inspection [31] or complementing the automatic assessment with human assessment [1, 46].

According to survey respondents, people use CAA more for summative assessment than formative assessment, although it is used for both. The literature suggests that CAA can be efficiently incorporated into many courses to aid formative assessment and student support throughout a course [19, 2, 71]. CAA makes it possible for students to learn from feedback and resubmit their work, which supports their learning and self-assessment [57], and it facilitates the generation of individualized questions according to students' attainment levels or previous success [35, 75]. This fits with the educational principles of constructivism: allowing students to learn from their mistakes and being contingent upon them [80].

3.3 What we can learn about CAA from other disciplines

Our survey shows that most respondents are teaching and assessing programming, so it is probably inevitable that their main use of electronic methods of coursework submission and marking focus on practical work and written exercises. Nevertheless, the survey covers teaching of a wide range of topics such as mathematics, information systems and ethics. It is therefore appropriate to see whether we can learn anything from the use of CAA in other disciplines.

Extensive work on the automated assessment of mathematics led to the formation in 2001 of the Scottish Centre for Research into On-Line Learning and Assessment (SCROLLA). This group has been working on assessment at the school-university interface and its' work is described by Beevers and Paterson [8]. They provide examples of automated assessment using simple questions, such as

Find the equation of the normal to the curve with equation

$f(x) = x^3 - 2x$ at the point where $x = 1$

And more complex ones, such as

A function is defined by $f(x) = (x^2+6x+12)/(x+2)$, $x \neq -2$

(a) Express $f(x)$ in the form $ax+bx+b/(x+2)$ stating the values of a and b

(b) Write down an equation for each of the two asymptotes

(c) Show that $f(x)$ has two stationary points

(d) Sketch the graph of f

(e) State the range of values of k such that the equation $f(x) = k$ has no solution

Their approach is to use software that can process user-entered mathematical expressions and to split each problem into a series of intermediate steps. This provides support to the weaker students, who can obtain partial marks even if they cannot answer the whole question immediately. Making this optional has benefits for stronger students in accordance with Wood's notions of contingency [80]. They also provide a range of modes of delivery ranging from an examination mode to a help mode that is useful for self-assessment. Much of this work could be adopted directly by computer scientists who are teaching their students mathematics. The approach of using optional, intermediate steps

has wider applicability for assessment of core computer science topics.

The use of automated assessment in business and management is also growing as teachers struggle to cope with rapidly growing class sizes, usually in conjunction with a Virtual Learning Environment (VLE). The area of learning tested is often quantitative but it differs from the assessment of mathematics described above in that the emphasis is on interpretation. Smailes [69] describes the benefits of using multiple choice, graphic and 'fill in the gaps' tests in Blackboard [9] for a first year Data Analysis course. Fuller [36] also discusses the use of CAA in assessing quantitative methods at both first year and postgraduate level. He presents sample questions based on every level of Bloom's taxonomy except synthesis, and argues that it is perfectly possible to assess analysis and evaluation through multiple choice questions or ones requiring a short phrase that can be automatically parsed.

Modern language teachers have developed a range of tools for supporting and assessing the acquisition of language skills, generically referred to as CALL (Computer Assisted Language Learning). Techniques used to assess learning of grammar and vocabulary include sentence reordering and gap filling; these are considered to have a high degree of validity in measuring attainment [66].

Work to apply automated assessment to extended pieces of writing is now producing results that are claimed to be as accurate as human markers for technical essays. Content and style are both measured and graded. The disadvantage of this approach for long essays is that a large dictionary of reference material is required, which makes the approach infeasible for most people. However for short answer questions this approach can be used to mark against a single sample response supplied by the setter. A good review of this work is provided by Whittington and Hunt [79].

The work described here has obvious parallels in computer science teaching and mainly measures recall, comprehension and application to straightforward problems. There is, however, another very different group of applications of computer aided assessment. These involve its use in measuring participation in discussion and creative writing. Hatzimoysis [40] gives a very clear picture of the use of virtual seminars in teaching philosophy topics, including ethics. He points out that the ongoing, asynchronous format of the virtual seminars gives formative feedback and that electronic discussion captures the contributions of individual students to the synthesis of a group report. In a computer science context, this can be used to measure students' ability to work in a group, as described by Fuller *et al* [37]. Virtual seminars for a final year course in Computer Support for Cooperative Working taken by psychology and computing students are described by the ASTER project [42], where the evaluation showed that assessment of seminar participation increased students' motivation to contribute and reflect on the subject matter. Sloman [68] also reports the benefits of virtual seminars in promoting student reflection; he used them to teach a course on economic principles. He concludes that 'Virtual seminars are best suited to what I call 'metaeconomics'. These involve students questioning assumptions behind policy objectives, relating economic concepts to current economic issues...'

4 USING CAA

It is apparent that CAA is not a panacea for the problems of assessment. The most widely available test type in current CAA systems is the multiple-choice test, or variations on this theme (e.g. [33]). These are most suitable for assessing lower order skills, and this is perhaps reflected by the survey responses where approximately 63% of first-year courses used some form of CAA, declining to approximately 44% for intermediate level courses and 31% for final year and postgraduate courses. This may, however, also be due to other factors such as the larger class sizes in first-year courses compared to later years, and the 'little and often' assessment philosophy used in many institutions for first-year courses.

A serious limitation is the 'black and white' marking schemes employed in CAA, which are only really suitable for assessments in courses like programming where an answer can be clearly categorized as right or wrong. Many systems for the automatic assessment of programming exercises have been developed (e.g. [3, 21, 47, 57, 65, 74]) and it is notable that our survey shows approximately 70% of first-year programming courses using CAA to some extent. Shades of gray in marking schemes have to be aggregated from a number of 'black and white' scores for different aspects of the assessment.

One of the major perceived drawbacks to the use of CAA is the cost in time and effort required to set up a CAA-based assessment. This depends partly upon the quality of the authoring framework used to create the assessments and their marking schemes, but it also depends on the time and effort spent in staff training to be able to use the authoring system and the imagination of staff in finding ways to use CAA effectively for particular courses. There is also a cost involved in administering the system, which may include making adjustments to cater for special-needs students, granting extensions to deadlines and collating results. It may be possible to reuse questions once they have been developed, but the rate at which the contents of courses can change, particularly in a fast-moving field like computing, means that a process of continuous maintenance is needed to prevent assessments from becoming stale.

Test design for automatic assessment is also more complicated than it is for manual assessment. If a marking scheme for manual assessment is badly designed, human markers can use their judgment to overcome its deficiencies. With an automatic marker, no such judgment is possible, and it therefore takes more effort to not only design questions that can be assessed successfully, but to implement and test the corresponding marking schemes.

In order to maximize the possibility of reuse, question banks can provide a solution. A locally-developed question bank can provide the necessary variety in assessment exercises; externally developed question banks may also help, but here there is a question of quality control and how well externally developed questions fit in with the needs of the courses at a particular institution.

The use of externally developed question banks requires adherence to standards to ensure interoperability. The overwhelming majority of survey respondents considered interoperability to be an important issue, regardless of whether they themselves use CAA at present. The IMS Question & Test Interoperability Specification [45] is designed specifically to

address this particular issue, although it is an open-ended specification that provides for vendor-specific extensions. The obvious danger with an extensible standard is that, like programming languages, different vendors will provide incompatible extensions that will actually hinder interoperability.

4.1 Alleviating Disadvantage

It has been suggested that the majority of university level courses offer a similar experience to each student taking them [24]; increasing numbers of students often implies a reduced amount of time to be spent upon individuals. This is complicated by the increasingly widely differentiated past experiences that students bring to university. An assessment system should not disadvantage any identifiable portion of the cohort. For example it is well documented that male and female students pursue electronic debates in subtly different fashions [48], so designing an assessment that rewards typical male behavior and punishes (by lack of marks) typical female behavior [18] is to be avoided or, if this is not possible, compensated by an assessment that is biased in the other direction.

It is obviously important that assessment is impartial and effective. The role of assessment has expanded over the last 50 years to provide evaluation of programs in addition to the traditional placement, selection and certification of individual learners. We must balance the need to provide robust evaluation data with the duty to serve the individual requirements of the learner [25].

Assessment has an important pedagogical role and must match the desired learning paradigms and taxonomic levels; assessment and learning are inextricably linked and the planned assessment style will shape the teaching and learning strategies. It therefore follows that assessment plays a pivotal role in driving the motivation of individual learners [39]. Personal motivation is driven by two sets of forces:

- Those provided externally by society that may include peer group pressures, goals set by progression requirements and career aspirations
- Emotional and internal expectations such as confidence, prior experiences, personal esteem, etc

So a learner can perceive a task as:

- Necessary and enjoyable
- Unnecessary but enjoyable
- Necessary but not enjoyable
- Unnecessary and not enjoyable [7]

If the style of the assessment matches the learning style of the learner the process of assessment will become more enjoyable. Cognitive style is one of the dimensions that affect learning style [29]. It has been shown to be of significance among students that fail to persist beyond the first year of university study [64] where students with a sequential style are more likely to be persistent than those with a random style. It is important to ensure the constraints of technology do not bias CAA towards a particular learning style.

As technology increases in capacity and functionality the constraints it imposes reduces. However there are still difficulties in adapting technology to evaluate higher level learning skills such as synthesis [36]. It is easy to automate multiple selection tests and considerably more difficult to analyze free form text answers. This generates the perception that automated testing is more appropriate for lower order learning skills. This perception is likely to change as lexical analysis improves or as questioning styles evolve. Carneson *et al* [13] have used multiple-choice questions to assess higher order learning skills; they demonstrate how to assess all but synthesis.

Automated assessment requires careful design and administration and the development effort can be considerable. The developments of question banks and support networks are demonstrating benefits and reducing the overheads of implementing CAA. Some interoperability problems exist when CAA products are designed to be platform specific or require particular security protocols [78], but there is widening acceptance of CAA. Interoperability standards are emerging [45] that will help to promote widespread use of question banks and the popularity of CAA.

4.2 Plagiarism and Security Issues

One important issue that needs to be considered when adopting CAA is security. Security in this discussion refers to activities by the students that could affect the validity of the assessment outcomes, for example, plagiarism, impersonation of the candidate, electronic eavesdropping and system hacking. Although students may cheat in any form of assessment, cheating is generally only perceived as a problem with summative assessment tasks. In formative assessment the students are seen as only cheating themselves and this of lesser concern.

Of particular concern is the security of assessment that is conducted online and in isolation from the educator, and the opportunities this offers for cheating. Recent research by Carpenter *et al* [14] found that 62% of students did not consider that working in groups on take home examinations or Web-based quizzes was cheating. This concurs with Barnett [6] who reported that the incidence of cheating increased when the supervisor left the room during an examination. These studies would tend to suggest that students sitting an online examination would be more inclined to cheat if they were unsupervised. In an attempt to address this problem, there have been various developments that facilitate automatic invigilation of online assessment. Technologies such as fingerprint identification have been developed to verify the online user [17]. Others describe processes for the management of unsupervised examinations to increase security. Thomas *et al* [74] conduct synchronized online examinations that use password access and a set time limit for completion. Mulligan [62] suggests using software to synchronize a student's computer clock with the network to verify the time that a test was taken, but there appears to be no clear and reliable way to verify the identity of the person who sits an assessment task online.

CAA, conducted under supervision, can be used to reduce the opportunities for cheating. Heron [55] successfully used computer based assessment to reduce cheating on paper-based assignments. She reports that requiring students to produce assignment work under supervision has virtually eliminated cheating on these assessment tasks. However, such practices do greatly restrict the types of assignment work that can be set. When we consider that assignment work and class tests have been found to have the highest rates of cheating and that students consider these to be among the most acceptable practices, CAA becomes an important

tool for the educator [14, 26]. Of further concern here is that a 10-year follow up study by Diekhoff [27] found that these forms of cheating are on the increase.

Using computers for assessment has provided new ways in which students may cheat. For example, Ala-Mutka [1] found that students generate meaningless comments in program code to satisfy an automatic style checker requirement for a certain percentage of comments. However, using computer-aided assessment has also provided educators with mechanisms to curb cheating. For example, providing tests on computer rather than on paper has also facilitated the generation of instant and individual sets of questions reducing the opportunity to cheat.

Hunt *et al* [44] stress the importance of providing students with assessment tasks that are relevant and appropriate. In support of this Ashworth [5] found that students will be more inclined to cheat if they do not see a learning purpose to the assessment. Furthermore, anxiety produced by an assessment task that is unfamiliar can induce students to cheat. Providing situationally authentic assessment tasks could therefore reduce the likelihood of cheating. For example, online programming examinations or open book examinations may provide a more realistic assessment of programming skills and are closer in nature to the actual work the students have already done and will potentially need in their future employment. In an evaluation of online programming tests by Mason and Doit [58], student responses indicated that the online tests provided a fairer assessment of those who deserved to pass or fail and that the online tests 'motivated them not to copy and to obtain the practical skills expected of them through course work' (p.144).

It seems that the issues here are complex. The use of CAA in certain situations is seen as increasing the risk of cheating and providing students with different opportunities to cheat. However, assessing the learner in an electronic environment has allowed for new ways to address the cheating problem.

4.3 Guidelines for adopting CAA

Adopting CAA can have a number of benefits for both staff and students. Staff benefit by saving time on marking, particularly when dealing with large groups, and by being able to assess 'little and often'. Students benefit by having the opportunity to work at their own pace, and to receive immediate feedback on the work that they submit. Other benefits may include reducing plagiarism and using data-mining techniques to discover trends in student submissions. However, making a decision to adopt CAA techniques requires careful preparation. Some of the factors to be considered are:

- **What types of assessment are possible?** CAA is still in its infancy, and most existing systems provide little more than multiple-choice questions, which are really only suitable for assessing the lowest-order skills. Systems for automatic assessment of programming assignments are also fairly common, although many other assessment types are possible. As CAA matures, these will undoubtedly become more widely accepted.
- **Which courses can benefit?** Mass courses with large enrolments will yield the greatest savings in terms of marking time, and lower-order skills are easiest to assess. Since first-year courses generally have the highest numbers and also concentrate more on lower-levels skills such as remembering, understanding and applying knowledge, these are obviously a good place to start. Courses on topics such as programming, which benefit from 'little and often' assessment and where submissions can be clearly categorized as right or wrong are also good prospects for CAA. The use of individualized questions can also help to reduce plagiarism.
- **Which system to adopt?** The choice is between adopting a third-party system (either a commercial system such as Blackboard [9] or WebCT [77], or a non-commercial systems such as BOSS [49] or CourseMaster [41]), and developing an in-house system. There may be other factors, such as an institutional decision to adopt a particular system as a vehicle for course management, but if such a system does not provide the desired CAA facilities it will be necessary to develop additional plug-in CAA modules, if this is possible. Developing an in-house system is the most expensive option, although it can provide a solution that is more closely aligned to the institution's requirements. Quality of support is an important issue when adopting a third-party system. Before taking a decision to buy such a system, it will be necessary to investigate what assessment facilities it provides and how easy it is to develop new plug-in modules for specialist institutional requirements.
- **How much investment will be needed?** There are costs involved in the initial deployment of a system, but there are also costs associated with staff training, maintenance, development of additional specialist tools, and development and testing of questions and marking schemes. The cost of developing questions can however be reduced by investing in the creation of a question bank to enable reuse.
- **What effects will a system failure have?** CAA requires reliable systems, where servers are available at all times. Disruptions due to server downtime, network outages or disk crashes must be guarded against, and a fallback position should be designed to mitigate the effects of disruptions according to the perceived level of risk. Using pilot schemes can help to discover what effects CAA systems have on system loading before attempting wide-scale deployment.
- **Are standards important?** The simple answer is 'yes'. Following IMS interoperability standards will make it easier to share material with systems at other institutions, and will also help with migration issues if upgrading to a different system becomes necessary. However, it is necessary to be aware of the open-ended nature of such standards and the possible use of incompatible vendor-specific extensions.

We should not forget that the primary beneficiaries of CAA systems should be the students. The system as a whole must be based on sound pedagogical principles and must avoid discrimination, or exclusion, based upon gender, disabilities or other factors. Careful design of any CAA system is necessary to ensure that this is so.

5 ACKNOWLEDGMENTS

The members of the working group would like to thank Petco Tsvetinov, from the Queesnland University of Technology, who helped in the data collection and preparation, but could not attend the conference in Thessaloniki where this report was prepared.

We would also like to thank Andrew Solomon for sharing his CAA experiences with us.

6 REFERENCES

[1] Ala-Mutka K, *Computer-assisted Software Engineering Courses*. Proceedings of IASTED International Conference Computers and Advanced Technology in Education, Cancun, 2002

[2] Anderson W, Krathwohl D, *A Taxonomy for Learning, Teaching and assessing,* http://www.cours.polymtl.ca/plu6035/PDF/anderson.pdf, 2001

[3] Arnow D, Barshay O, *Online Programming Examinations using WebToTeach*, Proceedings of ITiCSE'99, Krakow, 1999

[4] Arnow D, Barshay O, *WebToTeach: an interactive focused programming exercise system*, 29th Annual IEEE Frontiers in Education Conference, 1999

[5] Ashworth P, Bannister P, Thorne P, *Guilty in whose eyes? University students' perceptions of cheating and plagiarism in academic work and assessment*, Studies in Higher Education, 22, 1997

[6] Barnett DC, Dalton JC, *Why college students cheat?* Journal of College Student Personnel, 22, 1981

[7] Bedny GZ, Seglin MH, Meister D, *Activity theory: history, research and application*, Theoretical Issues in Ergonomics Science, 1(2), 2000

[8] Beevers E, Paterson JS, *Automatic assessment of problem-solving skills in mathematics*, Active Learning in Higher Education, 4(2), 2003

[9] Blackboard, http://www.blackboard.com/

[10] Bloom B, *Taxonomy of Educational Objectives: The Classification of Educational Goals: Handbook I*, Longman, New York, 1956

[11] Buck D, Stucki DJ, *Design Early considered Harmful: Graduated Exposure to Complexity and Structure Based on Levels of Cognitive Development*, SIGCSE 2000, Austin, 2000

[12] Burstein J, Leacock C, Swartz R, *Automated Evaluation of Essays and Short Answers*, Proceedings of 5th Annual CAA Conference, Loughborough, 2001

[13] Carneson J, Delpierre G, Masters K, *Designing and Managing Multiple Choice Questions*, http://www.le.ac.uk/cc/ltg/castle/resources/mcqman/mcqman01.html, 1999

[14] Carpenter DD, Harding TS, Montgomery SM, Steneck N, *P.A.C.E.S – A study on academic integrity among engineering undergraduates (preliminary conclusions),* ASEE Annual Conference & Exposition, Montreal, 2002

[15] Chan CC, Tsui MS, Chan MYC, *Applying the Structure of the Observed Learning Outcomes (SOLO) Taxonomy on Student's Learning Outcomes: an empirical study*, Assessment & Evaluation in Higher Education, 27(6), 2002

[16] Christie JR, *Automated Essay Marking for both Style and Content*, Proceedings of 3rd Annual CAA Conference, Loughborough, 1999

[17] Clariana R, Wallace P, *Paper-based versus computer-based assessment: key factors associated with the test mode effect*, British Journal of Educational Technology, 33, 2002

[18] Cook J, Leatherwood C, Oriogun P, *Online Conferencing with Multimedia Students: Monitoring Gender Participation and Promoting Critical Debate,* Proceedings of 2nd Annual LTSN-ICS Conference, London, 2001

[19] Cox K, Clark D, *The use of formative quizzes for deep learning*, Computers and Education, 30, 1998

[20] Culwin F, MacLeod A, Lancaster T, *Source Code Plagiarism in UK HE Computing Schools, Issues, Attitudes and Tools*, Southbank University, London, Commissioned by JISC, 2001

[21] Daly C, *RoboProf and an Introductory Programming Course*, Proceedings of ITiCSE'99, Krakow, 1999

[22] Dalziel J, Gazzard S, *Next generation computer assisted assessment software: the design and implementation of WebMCQ*, Proceedings of 3rd Annual CAA Conference, Loughborough 1999

[23] Davies P, *Computer Aided Assessment MUST be more than multiple-choice tests for it to be academically credible?* Proceedings of 5th Annual CAA Conference, Loughborough 2001

[24] Davis H, Carr L, Cooke E, White S, *Managing Diversity: Experiences Teaching Programming Principles* in Proceedings of 2nd Annual LTSN-ICS Conference, London, 2001

[25] Delandshire G, *Implicit theories, unexamined assumptions and the status quo of educational assessment*, Assessment in Education, 8(2), 2001

[26] Dick M, Sheard J, Markham S, *Is it OK to cheat?* Proceedings of ITiCSE'01, Canterbury, 2001

[27] Diekhoff GM, LaBeff EE, Clark RE, Williams LE, Francis B, and Haines VJ, *College cheating: ten years later*, Research in Higher Education, 37, 1996

[28] Duke-Williams E, King T, *Using Computer-Aided Assessment to Test Higher Level Learning Outcomes,* Proceedings of 5th Annual CAA Conference, Loughborough, 2001

[29] Dunn R, Beaudy J, Kalvan A, *Survey of research on learning styles*, Education Leadership, 46, 1989

[30] English J, Siviter P, *Experience with an Automatically Assessed Course*, Proceedings of ITiCSE'00, Helsinki, 2000

[31] English J, *Experience with a Computer-Assisted Formal Programming Examination*, in Proceedings of ITiCSE'02, Aarhus, 2002

[32] Farthing DW, Jones DM, McPhee D, *Permutational Multiple-Choice Questions: An Objective and Efficient Alternative to Essay-Type Examination Questions*, Proceedings of ITiCSE'98, Dublin, 1998

[33] Farthing DW, McPhee D, *Multiple choice for honours-level students? A statistical evaluation,* Proceedings of 3rd Annual CAA Conference, Loughborough, 1999

[34] Freeman M, McKenzie, *SPARK, a confidential web-based template for self and peer assessment of student teamwork: benefits of evaluating across different subjects*, British Journal of Educational Technology, 33, 2002

[35] Frosini G, Lazzerini B, Marcelloni F, *Performing automatic exams*, Computers & Education, 31, 1998

[36] Fuller M, *Assessment for real in Virtual Learning Environments – how far can we go?* Proceedings of Interface - Virtual Learning and Higher Education, Mansfield College, Oxford 2002

[37] Fuller U, Slater J, Tardivel G, *Virtual Seminars, Real Networked Results?* Proceedings of ITiCSE'98, Dublin, 1998

[38] Hagan D, Sheard J, *Monitoring and Evaluating a Redesigned First Year Programming Course*, Proceedings of ITiCSE'97, Uppsala, 1997

[39] Hargreaves DJ, *Student learning and assessment are inextricably linked,* European Journal of Engineering Education 22(4), 1997

[40] Hatzimoysis A, *A Philosophy Primer in Virtual Seminars* http://www.prs-ltsn.leeds.ac.uk/philosophy/events/hatzimoysis1.html, 2002

[41] Higgins C, Symeonidis P, Tsintifas A, *The Marking System for CourseMaster*, Proceedings of ITiCSE'02, Aarhus, 2002

[42] Hogarth S, *Virtual Seminars in Psychology and Computing* http://cti-psy.york.ac.uk/aster/resources/case_studies/reports/yk_06/yk_06.html, 2002

[43] Hoggarth G, Lockyer M, *An Automated Student Diagram Assessment System*, Proceedings of ITiCSE'98, Dublin, 1998

[44] Hunt N, Hughes J, Rowe G, *Formative automated computer testing (FACT),* British Journal of Educational Technology, 33, 2002

[45] IMS http://www.imsglobal.org

[46] Jackson D, *A Semi-Automated Approach to Online Assessment*, Proceedings of ITiCSE'00, Helsinki, 2000

[47] Jackson D, *A Software System for Grading Student Computer Programs*, Computers & Education, 27(3/4), 1996

[48] Janson S, *Gender and the Information Society: A Socially Structured Silence*, in Siefert M, Gerbner G, Fisher J (Eds.), The Information Gap: How Computers and Other Communication Technologies Affect the Social Distribution of Power, Oxford: Oxford University Press, 1989

[49] Joy M, Luck M, *Effective Electronic Marking for Online Assessment*, Proceedings of ITiCSE'98, Dublin, 1998

[50] JPLAG, http://www.jplag.de/

[51] Kashy E, Tsai Y, Thoenssen M, Morrissey D, CAPA, *An Integrated Computer Assisted Personal Assignment System*, American Journal of Physics, 61(12), 1993

[52] Korhonen A, Malmi L, *Algorithm simulation with automatic assessment*, Proceedings of ITiCSE'00, Helsinki, 2000

[53] Kumar A, *Learning the Interaction Between Pointers and Scope in C++,* ITiCSE '01, Canterbury, 2001

[54] Lee S, *Development of instructional strategy of computer application software for group instruction* in Computers and Education, 37(1), 2001

[55] Le Heron J, *Plagiarism, learning dishonest or just plain cheating: The context and countermeasures in information systems teaching*, Australian Journal of Educational Technology, 17, 2001

[56] Lister R, *Objectives and Objective Assessment in CS1*, Proceedings of SIGCSE '01, Charlotte, 2001

[57] Malmi L, Korhonen A, Saikkonen R, *Experiences in Automatic Assessment on Mass Courses,* Proceedings of ITiCSE'02, Aarhus, 2002

[58] Mason DV, and Woit DM, *Integrating technology into computer science education.* Proceedings of 29th SIGCSE Technical Symposium on Computer Science Education, Atlanta, 1998

[59] Mason O, Grove-Stephenson I, *Automated free text marking with Paperless School*, Proceedings of 6th Annual CAA Conference, Loughborough 2002

[60] Medley MD, *Online Finals for CS1 and CS2*, Proceedings of ITiCSE'98, Dublin, 1998

[61] MOSS, http://www.cs.berkley.edu/~aiken/moss.html

[62] Mulligan B, *Pilot study on the impact of frequent computerized assessment on student work rates*, Proceedings of 3rd Annual CAA Conference, Loughborough, 1999

[63] Question Mark, http://www.questionmark.com/

[64] Ross JL, Drysdale MTB, & Schulz RA, *Cognitive learning styles and academic performance in two postsecondary computer courses.* Journal of Research on Computing in Education, 33(4), 2001

[65] Saikkonen R, Malmi L, Korhonen A, *Fully Automatic Assessment of Programming Exercises*, Proceedings of ITiCSE'01, Canterbury, 2001

[66] Shimatani H, Kitao K, *Computer Aided Instruction: A bibliography,* http://ilc2.doshisha.ac.jp/users/kkitao/library/biblio/cai-bib.htm

[67] Simon, Summons P, *Automated testing of Databases and Spreadsheets – the Long and the Short of it*, Proceedings of ACE2000, Melbourne, 2000

[68] Sloman J, *Use of virtual seminars in Economic Principles*, http://www.economics.ltsn.ac.uk/showcase/sloman_virtual.htm, 2002

[69] Smailes J, *Experiences of using Computer Aided Assessment within a Virtual Learning Environment*, http://www.business.ltsn.ac.uk/events/BEST2002/Papers/smailes.PDF, 2002

[70] Solomon A, http://www-staff.it.uts.edu.au/~andrews/ personal communication, 2003

[71] Thelwall M, *Computer-based assessment: a versatile educational tool*, Computers & Education 34(1), 2000

[72] Thomas P, Price B, Petre M, Carswell L, Richards M, *Experiments with Electronic Examinations over the Internet*, Proceedings of 5th Annual CAA Conference, Loughborough 2001

[73] Thomas P, Price B, Paine C, Richards M, *Remote electronic examinations: student experiences*, British Journal of Educational Technology, 33, 2002

[74] Thorburn G, Rowe G, *PASS: An Automated System for Program Assessment*, Computers & Education, 29(4), 1997

[75] Trentin G, *Computerized adaptive tests and formative assessment*, Journal of Educational Multimedia and Hypermedia, 6, 1997

[76] Ward A, *Using peer assessment assisted by ICT for programming assignments*, Interactions, 5(1), 2001

[77] WebCT, http://www.webct.com/

[78] White S, Davis H, *Creating large-scale test banks: a briefing for participative discussion of issues and agendas*, Proceedings of 4th Annual CAA Conference, Loughborough 2000

[79] Whittington D, Hunt H, *Approaches to the computerized assessment of free text responses*, Proceedings of 3rd Annual CAA Conference, Loughborough 1999

[80] Wood D, *Aspects of Teaching and Learning,* in Light P, Sheldon S and Woodhead M (eds.) Learning to Think, Routledge, London, 1991

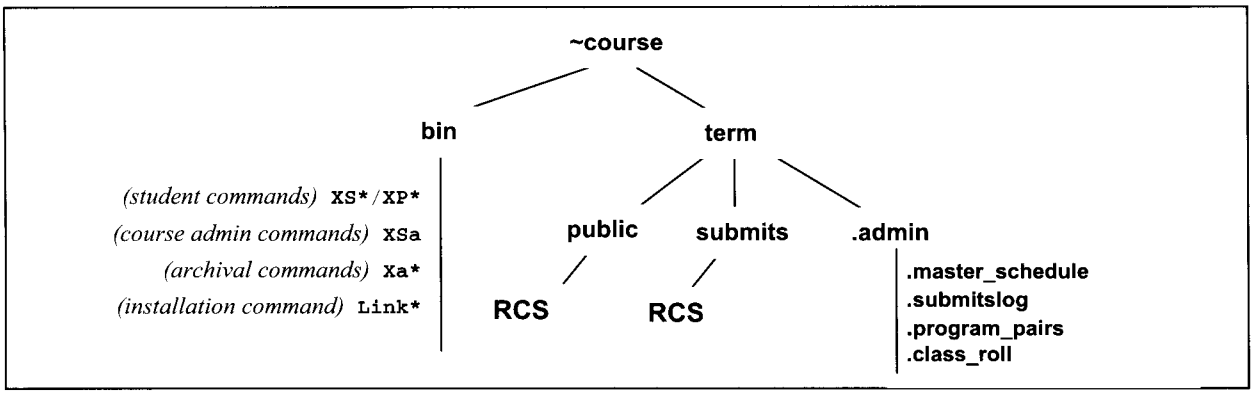

Figure 2. Allocation of Repository Files

extensively in the literature, there are few examples of successful technology transfer. The authors posit that technology transfer is the more difficult problem, even within a small department. In this section we present an account of our experience at technology transfer.

Closing the sale. The commitment by colleagues to use the repository required assurance that the repository worked as promised, that they would have technical support, and that, in the end, they would actually save time. Fortunately, one of the adopters had used an earlier version of the repository.

That first step. Special administrative commands were provided to the instructor to automate the set up of the repository in the course account. The adopters were provided, as promised, direct technical support and explanation (a draft of this paper served as the user guide). One adopter expressed annoyance at benign error messages that belied the successful completion of repository commands. Students received the two-command sequence to "install" the repository commands in their Unix accounts. As can be expected, some students failed to execute the two commands and were not able to submit the first assignment. Initial feedback from the adopters and students indicated that user documentation and orientation were lacking. Task-oriented documentation has since been developed for use in the summer term.. The need for a user-friendly product can not be understated.

Immediate benefit. The trauma of the first step notwithstanding, success was guaranteed once the adopters saw the preparation time for grading decrease from 6 hours (unpacking emails) to several minutes. At the request of the adopters, a customized script was written to extract assignment files from the repository into student directories in which plagiarism checking and program grading were performed. The adopters realized that the efficiency introduced by the repository make it possible to give more programming assignments.

A few false steps. Although the repository simplifies managing submitted programs, the instructor is responsible for maintaining the master schedule. For one assignment, all student submissions were rejected because the adopter forgot to add a new assignment to the master schedule. (Rather than returning to the tedious step of unpacking e-mails, the adopter assigned all students the maximum score.) In the next release, an administrative repository commands will be included to remind the instructor to update the master schedule, and to guide the instructor through the process.

Why not do this? Success breeds demands for new features. Each realization of a new benefit of the repository was followed by a request to add new features to automate manual tasks. Several of these "why-not-do-this?" sessions made it clear that submission via the web and e-mail must be supported.

Serendipity. The adoption of the repository for the Java course coincided with a graduate thesis project to automate the grading of student Java programs. The automated grading integrated easily into the repository. The adopters received the unexpected benefit of free grading for several assignments. The prospect of efficient file management, coupled with automated grading, solidified the adopters' commitment to use the repository in the future.

A step forward. Although we resisted providing a non-Unix interface to the repository system, the technology transfer experience made it clear that having multiple modes of access best serve the needs of faculty and students. A web-based interface is planned for the summer term. We are currently developing a list-server style e-mail interface to the repository to enable students to submit programs via e-mail in the manner similar to Arnow [4]. Authentication requires that the student register the source email address while logged into their departmental Unix system. E-mail submissions follow the simple protocol: (1) the subject line must contain the repository command to be executed; and (2) only one file may be submitted per email; and (3) the body of the e-mail message must contain the file.

6. Conclusions and Future Work

In our department there is growing realization that the repository system is a valuable course management tool that can enforce student accountability (submission deadlines, plagiarism detection) while reducing the "bookkeeping" overhead of courses. Ultimately, the repository makes it possible to support large programming classes by reducing the "handling costs" of student programs. The benefits are magnified when students are provided multiple interfaces to the repository – Unix commands, e-mail and the web. The repository environment is programmable to support instructor preferences and submission-time actions such as compilation, execution with test data, and plagiarism checking. Faculty members involved in the technology transfer experience are convinced that the repository makes it possible for them to give students a better learning experience by being able to assign more programs.

Future work includes capitalizing on existing momentum towards adoption of repositories for all programming courses, taking into account the lessons learned from the technology transfer experience.

- Support multiple modes of student interaction – email and web browser.
- Apply research results in automated software testing to program grading.
- Provide training and support documentation and services for students and adopting faculty.
- Use the programming environment to gain insight into student programming behavior, for diagnostic and training purposes.

7. ACKNOWLEDGMENTS

The authors thank the many students who endured the growing pains of the repository, and are grateful to colleagues who saw the value of adopting the repository for use in their courses. This work was partially supported by National Science Foundation grant EIA-9906590.

8. REFERENCES

[1] Jones, E.L., "Grading Student Programs - A Software Testing Approach," *Journal of Computing in Small Colleges 16,* 2, January 2001, 185-192.

[2] Canup, M.J. and Shackelford, R.L., "Using Software to Solve Problems in Large Computing Courses," *Proceedings SIGCSE '98*, Atlanta, GA, USA, 135-139.

[3] Emory, D. and Tamassia, R., "JERPA: A Distance Learning Environment for Introductory Java Programming Courses, "*Proceedings SIGCSE'02*, February 27-March 3, 2002, Covington, Kentucky, USA, 307-311.

[4] Arnow, D.M., ":-) When You Grade That: Using E-Mail and the Network in Programming Courses," *Proceedings of the 1995 ACM Symposium on Applied Computing,*, February 1995, 10-13.

[5] Churcher, N.I., Cockburn, A.J.G., McMaster, B.N., Horlor, J., "CUTE: The Design and Evolution of a First Year Programming Environment," *Proceedings of the 1998 International Conference. on Software Engineering: Education & Practice*, Dunedin, New Zealand, January 26-29, 1998.

[6] Woid, D. and Mason, D., "Effectiveness of Online Assessment," *Proceedings SIGCSE'03*, February 19-23, 2003, Reno, Nevada, USA, 137-141.

[7] Jackson, D. and Usher, M., "Grading Student Programs Using ASSYST," *Proceedings SIGCSE '97,* 1997, *335-339.*

[8] Reek, K.A., "The TRY System – or – How to Avoid Testing Student Programs," *Proceedings SIGCSE Bulletin vol. 21*, no. 1, February 1989, 112-116.

[9] Walter F. Tichy, "RCS -- A System for Version Control," *Software--Practice & Experience 15*, 7 (July 1985), 637-654.

Appendix A. Administrative Commands for Course Management

Submission Management	
XSastatus	List names of files in submission repository.
XSamcopy	Place a copy of the specified submission repository file(s) into current Unix directory.
XSamview	Display on the contents of specified submission repository file(s).
XSapost	Store file into a student's submission repository.
XSaclean	Remove file from a student's submission repository.
XSasubmitslog	Display submission history for a student and/or assignment.
XSlate	Check whether submission is attempted past the due date.
Xa_rename	Rename source files to expose extension, so files can be compiled or edited.
Xa_jrename	Rename Java source files to expose extension, and store files in per-student directories.
Repository Archival	
Xa_initialize	Create Unix directories for repository system for a new term.
Xa_change_terms	Create repository for new term after archiving repository from previous term. Archived repository commands renamed to indicate term.