# Computer Based Assessment Systems Evaluation via the ISO9126 Quality Model

*Salvatore Valenti, Alessandro Cucchiarelli, and Maurizio Panti*
*Istituto di Informatica, Università di Ancona, Italy*

valenti@inform.unian.it alex@inform.unian.it panti@inform.unian.it

## Executive Summary

The interest in developing Computer Based Assessment (CBA) systems has increased in recent years, thanks to the potential market of their application.

Many commercial products, as well as freeware and shareware tools, are the result of studies and research in this field made by companies and public institutions. This noteworthy growth in the market raises the problem of identifying a set of criteria that may be useful to an educational team wishing to select the most appropriate tool for their assessment needs. The scientific literature is very poor in respect of this issue. An important help is provided in this direction, by a number of research studies in the field of Software Engineering providing general criteria that may be used to evaluate software systems. Furthermore, a relevant effort has been made in this field by the International Standard Organization that in 1991 defined the ISO9126 standard for "Information Technology – Software Quality Characteristics and Sub-characteristics" (ISO, 1991). It is important to note that a typical CBA system is composed by:

- A Test Management System (TMS) - i.e. a tool providing the instructor with an easy to use interface, the ability to create questions and to assemble them into tests, the possibility of grading the tests and making some statistical evaluations of the results.
- A Test Delivery System (TDS) - i.e. a tool for the delivery of tests to the students. The tool may be used to deliver tests using paper and pencil, a stand-alone computer, on a LAN, or over the web. The TDS may be augmented with a web-enabler used to deliver the tests over the Web. In many cases producers distribute two different versions of the same TDS, one to deliver tests either on single computers or on LAN, and the other to deliver tests over the web.

The TMS and TDS modules may be integrated in a single application or may be delivered as separate applications. Thus, it is of fundamental importance to devise a set of quality factors that can be used to evaluate both the modules belonging to this general structure of a CBA system.

Purpose of this paper is to discuss a set of quality factors that can be used to evaluate a CBA System using the standard ISO9126, which provides a general framework for evaluating a commercial off the shelf software without covering the specificity of the application domain. Thus, our effort has been mainly devoted to the elicitation of a set of domain specific quality factors for the evaluation of a Computer Based Assessment System. The ISO9126 standard is a quality model for product assessment that identifies six quality characteristics: functionality, usability, reliability, efficiency, portability and maintainability. Each of these characteristics is further decomposed in a set of sub characteristics. Thus for instance, Functionality is characterised by Suitability, Accuracy, Interoperability, Compliance and Security. None of the quality fac-



Editor: Karen Nantz

tors discussed above can be measured directly, but must be defined in terms of objective features to be assessed. These features should be identified by taking into account the context of the evaluation, i.e., a description of the target system, and the environment into which it will be deployed. The quality characteristics defined by the ISO 9126 standard may be classified with respect to the domain "specificity" coordinate. Functionality, for instance, is highly dependent on the educational domain to which CBA systems belong. On the other hand, Maintainability is a feature that can be only evaluated either by the developer or by a third party having at his disposal the technical documentation of the project and the source code. A third class is represented by the quality characteristics that, although assessable, are independent from specific domain taken into account, as for instance, portability.

In this paper we will focus the discussion on the domain specific aspects of the ISO9126 standard, i.e. Functionality, Usability and Reliability, leaving untreated the remaining characteristics that are either domain independent or un-assessable by the end users. For each domain specific quality factor, we will discuss the common features of the Test Management and of the Test Delivery sub-systems, and then take into account those applicable to each one of the two functional components in special sub-sections. Thus, the discussion is organised as shown in table A.

| **Characteristic** | **Sub-characteristic** | **Sub-sections** |
|---|---|---|
| Functionality | Suitability | Suitability of TMSs<br>Suitability of TDSs |
| | Security | |
| | Interoperability | |
| Usability | Operability<br>Understandability | Operability & Understandability of TMSs<br>Operability & Understandability of TDS |
| | Learnability | |
| Reliability | | |

Table A – Characteristics that will be discussed in the paper

The term cheating is used to address dishonest practices that students may pursue in order to gain better grades. In the final section of the paper, we will discuss cheating control from the technical point of view, presenting some requirements that should be satisfied either at the component or at the system level of a TDS. We will also discuss how an attempt at controlling cheating may affect the interface, the question management and the test management functional blocks of a TDS. Then we will discuss the effects of cheating control on the security of a TDS.

As a follow-up of this work the list of quality factors identified in the paper will be hosted as a questionnaire on the web site of our department and made available to all researchers wishing to review a CBA system.

The obtained results will be made available to all interested parties.

**Keywords**: Evaluation, Computer Based Assessment Systems, ISO9126, Quality model

# Introduction

Most solutions to the problem of delivering course content, supporting both student learning and assessment, nowadays imply the use of computers, thanks to the continuous advances of Information Technology. According to Bull (1999), using computers to perform assessment is more contentious than using them to deliver content and to support student learning. In many papers, the terms Computer Assisted Assessment (CAA) and Computer Based Assessment (CBA) are often used interchangeably and somewhat inconsistently. The former refers to the use of computers in assessment. The term encompasses the uses of computers to deliver, mark and analyze assignments or examinations. It also includes the collation and analysis of data gathered from optical mark readers. The latter (that will be used in this paper) addresses the use of computers for the entire process, including assessment delivery and feedback provision (Charman and Elmes, 1998).

The interest in developing CBA tools has increased in recent years, thanks to the potential market of their application. Many commercial products, as well as freeware and shareware tools, are the result of studies and research in this field made by companies and public institutions. For an updated survey of course and test delivery/management systems for distance learning see Looms (2001). This site maintains a description of more then one hundred products, and is constantly updated with new items. This noteworthy growth in the market raises the problem of identifying a set of criteria that may be useful to an educational team wishing to select the most appropriate tool for their assessment needs. According to our findings, only two papers have been devoted to such an important topic (Freemont & Jones, 1994; Gibson et al., 1995). The major drawbacks shown by both papers are: a) the unstated underlying axiom that a CBA system is a sort of monolith to be evaluated as a single entity, and b) the lack of an adequate description of how the discussed criteria were arrived at. Since anyone could come up with some kind of list, what needs to be known is what makes them valid.

A typical CBA system is composed by:

- A Test Management System (TMS) - i.e. a tool providing the instructor with an easy to use interface, the ability to create questions and to assemble them into tests, the possibility of grading the tests and making some statistical evaluations of the results.
- A Test Delivery System (TDS) - i.e. a tool for the delivery of tests to the students. The tool may be used to deliver tests using paper and pencil, a stand-alone computer, on a LAN, or over the web. The TDS may be augmented with a web-enabler used to deliver the tests over the Web. In many cases producers distribute two different versions of the same TDS, one to deliver tests either on single computers or on LAN, and the other to deliver tests over the web. This is the policy adopted for instance by Cogent Computing Co. (2000) with CQuest-Test and CQuest-Web.

The TMS and TDS modules may be integrated in a single application, as for instance InQsit (2000) developed by the Ball State University, or may be delivered as separate applications. As an instance of this latter policy, we may cite ExaMaker & Examine developed by HitReturn (2000).

Therefore, it is very important to identify a set of quality factors that can be used to evaluate both the modules belonging to this general structure of a CBA system.

Although the literature on guidelines to support the selection of CBA systems seems to be very poor, there are many research studies in Software Engineering providing general criteria that may be used to evaluate software systems (Anderson, 1989; Ares Casal et al., 1998; Henderson et al., 1995; Nikoukaran et al, 1999; Vlahavas et al. 1999). A relevant effort has been made in this field by the International Standard Organization which in 1991, defined the ISO9126 standard for “Information Technology – Software Quality Characteristics and Sub-characteristics” (ISO, 1991).

This paper identifies a set of quality factors that can be used to evaluate a CBA System using the standard ISO9126, which provides a general framework for evaluating a commercial off the shelf software without covering the specificity of the application domain. Thus, our effort has been mainly devoted to the elicitation of a set of domain specific quality factors for the evaluation of a Computer Based Assessment System.

## ISO9126 Quality Model

The standard ISO9126 is a quality model for product assessment that identifies six quality characteristics: functionality, usability, reliability, efficiency, portability and maintainability.

Functionality is “a set of attributes that bear on the existence of a set of functions and their specified properties” (ISO, 1991). The functions are those that satisfy stated or implied needs. This characteristic answers the question: Are the required functions available in the software to be assessed?

Usability is “a set of attributes that bear on the effort needed for use, and on the individual assessment of such use by a stated or implied set of users” (ISO, 1991). The degree of usability will depend on who the users are. The problem with usability is that it depends on people’s perceptions of what is easy to use. Therefore, usability is the least objective quality factor and the most difficult to measure.

Reliability is “a set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time” (ISO, 1991). This characteristic answers the question: is the software under evaluation reliable?

Efficiency is “a set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions” (ISO, 1991).

Portability is “a set of attributes that bear on the ability of software to be transferred from one environment to another” (ISO, 1991).

Maintainability is “a set of attributes that bear on the effort needed to make specified modifications” (ISO, 1991). Maintenance requires analyzing the software to find the fault, making a change, ensuring that the change does not have side effects and then testing the new version.

Each of the quality characteristics is decomposed in subcharacteristics, as shown in Table 1.

None of the quality factors discussed above can be measured directly, but must be defined in terms of objective features to be assessed. These features should be identified by taking into account the context of the evaluation, i.e., a description of the target system, and the environment into which it will be deployed. To buy a car, the context is the customer situation, i.e. financial resources, driving patterns, aesthetic tastes, and so on. For an organization, the context includes the organization's mission, its structure, and its existing procedures for the tasks that will be affected by the target system. From the context, the project personnel will adduce various, possibly ill defined, qualities that the target system should exhibit (Hansen, 1999).

The quality characteristics defined in the ISO 9126 standard may be classified with respect to the domain “specificity” coordinate. Functionality, for instance, is highly dependent on the educational domain to which CBA systems belong. On the other hand, maintainability is a feature that can be only evaluated either by the developer or by a third party having at his disposal the technical documentation of the project and the source code. In our opinion it is impossible for the end-user to assess the maintainability of an off-the-shelf package. A third class is represented by the quality characteristics that, although assessable, are independent from specific domain taken into account. Portability, for instance, belongs to this category. Pilj (1996) suggests adopting the checklist of table 2 to evaluate Installability, a sub-item of Portability.

The checklist of Table 2 is general enough to be used to evaluate any kind of software.

| Characteristic | Sub-Characteristics |
|---|---|
| Functionality | • Suitability covers fitness for purpose.<br>• Accuracy checks the degree of precision of calculated values defined as "attributes of software that bear on the provision of right or agreed results or effects".<br>• Interoperability deals with the "attributes of software that bear on its ability to interact with specified systems."<br>• Compliance covers the adherence "to application-related standards or conventions or regulations in laws and similar prescriptions".<br>• Security is intended as "attributes of software that bear on its ability to prevent unauthorized assess, whether accidental or deliberate, to programs and data". |
| Usability | • Understandability evaluates the attributes of software that bear on the users' effort for recognizing the logical concept and its applicability.<br>• Learnability evaluates the attributes of software that bear on the users' effort for learning its application.<br>• Operability evaluates the attributes of software that bear on the users' effort for operation and operation control. |
| Reliability | • Maturity means "the frequency of failure by faults in the software".<br>• Fault tolerance is directed towards the evaluation of software robustness and is defined as "attributes of software that bear on its ability to maintain a specified level of performance in cases of software faults or infringement of its specified interface".<br>• Recoverability addresses the "capability to re-establish its level of performance and recover the data directly affected in case of failure and on the time and effort needed for it". |
| Efficiency | • Time behavior means evaluating the software with respect to "response time and processing time and on throughput rates in performance of its function".<br>• Resource behavior is "the amount of resources used and the duration of such use' |
| Portability | • Adaptability defines the attributes of software that bear on the opportunity for its adaptation to different specified environments without applying other actions or means than those provided for this purpose for the software considered.<br>• Installability deals with the attributes of software that bear on the effort needed to install the software in a specified environment.<br>• Conformance defines the attributes allowing the software adhere to standards or conventions relating to portability.<br>• Replaceability is related to the attributes of software that bear on the opportunity and effort of using it in the place of specified other software in the environment of that software. |
| Maintainability | • Analyzability defines the characteristics of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.<br>• Changeability defines the attributes of software that bear on the effort needed for modification, fault removal or for environmental change.<br>• Stability deals with the attributes of software that bear on the risk of unexpected effect of modifications.<br>• Testability is related with those attributes that bear on the effort needed for validating the modified software. |

**Table 1 – The ISO9126 Characteristics and Subcharacteristics**

## Quality Factors for the Evaluation of a CBA System

In this section we will focus our discussion on the domain specific aspects of the ISO9126 standard Functionality, Usability and Reliability, leaving untreated the remaining characteristics that are either domain independent or un-assessable by the end users. The interested reader can find a discussion and a

| |
|---|
| • The Installation procedure is menu driven and interactive. |
| • The Installation procedure makes no changes written to system files without user consent. |
| • The Installation procedure provides default settings that user can override (drive, directory, etc.). |
| • Upon completion, the installation procedure returns the user to the place where it was executed. |
| • The Installation procedure is capable of working with networks. |
| • The Installation procedure allows archiving of the old version. |
| • Aborting installation is possible. |
| • Deinstallation is possible. |
| • Updates change only the affected files. |
| • There is a version statement on startup. |

**Table 2 - A checklist for the evaluation of Installability (adapted from Pilj, 1996)**

number of checklists that support the evaluation phase of the remaining quality factors in the IEEE Recommended Practice for Software Acquisition Standard (IEEE, 1993): a standard describing "a set of useful quality practices that can be selected and applied during one or more steps in a software acquisition process".

For each domain specific quality factor, we will discuss the common features of the Test Management and of the Test Delivery sub-systems, and then take into account those applicable to each one of the two functional components in special sub-sections.

Table 3 provides a synoptical view of the quality factors and of their sub-characteristics that will be discussed in this section.

## *Functionality*

The subcharacteristics for functionality include suitability, security and interoperability.

### Suitability

In our view suitability represents the most important quality factor to be taken into account when evaluating a CBA system.

#### Suitability of TMS

The Test Management System should provide the instructor with an easy to use interface, the ability to create questions and to assemble them into tests, and the possibility of grading the tests and making some statistical evaluations of the results. Therefore, as an indirect measure of suitability we decided to adopt the question management and test management capabilities. Question management is related to all aspects of the authoring questions; test management concerns the assembling of questions into exams and the evaluation of the results.

**Question Management.** "Types of questions" and the "Question structure" can be used to assess the question management issue.

**Types of Questions.** The most common types of questions are multiple choice, multiple response, true/false, selection/association, short answer, visual identification/hot spot and essay (Cucchiarelli, 2000). Each of the question categories may be used to evaluate different types of knowledge. Therefore, the selection of a TMS may be driven by the ability to be assessed, according to the class of questions made available. On the other hand, many universities are adopting the same tool for all courses in order to reduce costs and to allow students to interact in the same way throughout their evaluation process. This obviously imposes the requirement of selecting a TMS that provides the widest range of question types available since different learning outcomes may be assessed within different courses.

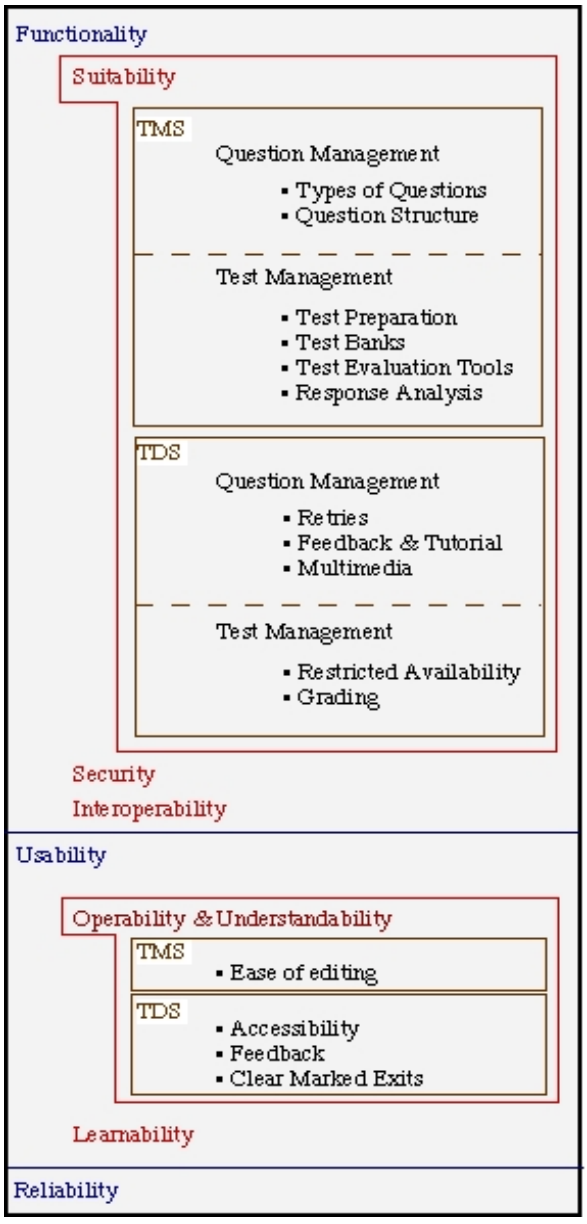



**Table 3 – Domain specific quality factors and subcharacteristics for CBA Systems**

While almost all commercial TMS provide the ability to build multiple choice questions (MCQ), very few of them implement Hot Spot or Selection/Association type questions. An even smaller subset of TMS claim to implement Essay type questions (CQuest, 2000; InQsit, 2000). Although there are some research efforts on the automatic scoring of essay type questions, mainly in the area of natural language understanding (Burstein & Chodorow, 1999; Foltz et al.,1999), the assessment of this class of questions relies on the manual intervention of the teacher for the commercial products on the market.

**Question Structure.** We can distinguish among information specific to the question type, information tied to the educational task to be assessed through the question and information related to the scoring policy adopted (thus dependent on the question type). Not all this type of information is made available by existing TMS: therefore this is an interesting aspect for identifying the system that best matches the educational needs to be assessed.

As an example of information available on question type, we will discuss multiple choice questions. This class of questions is organized into three parts: a stem, a key and some distractors. The problem to which the student should give an answer is known as stem. The list of suggested solutions may include words, numbers, symbols or phrases and are called alternatives, choices or options. The student is asked to read the stem and to select the alternative that is believed to be correct. The correct alternative which must be one, and only one, is simply called the key, whilst the remaining choices are called distractors, since their intended function is to distract students from the correct one.

To evaluate the question structure of a TMS, the number of different choices allowed and the appearance they have (radio vs. push buttons) must be taken in account. It is useful to note that the spread among the maximum number of allowed distractors for different TMS is very large, ranging from 3 to “no reasonable limit” for Perception (Perception, 2001). This could be used as a metric for the evaluation of a TMS, although many authors suggest that four choice items are good enough to reduce the chance of guessing the result while maintaining the effort of devising a real equivalent number of distractors (usually the fourth distractor in five choice questions tend to be difficult to devise and is weaker than the others). (Gronlund, 1985, p. 183)

The educational task to be assessed represents another important issue. In fact, if the test is used to evaluate the instructional process, additional fields to store a) the source of each question, b) the paper to which it is related and c) the topic covered and d) the author of the question itself ought to be provided. Furthermore, a teacher may wish to assess a topic at different cognitive levels, such as those defined in

the well known Bloom's taxonomy (Bloom, 1956). Thus, an additional field for storing such information should be defined.

Many commercial TMS allow user-defined fields. Therefore, a good selecting criterion would be the ability to access these fields to perform test evaluation procedures.

Each class of available questions may support different scoring schemes. The simplest way to assign a score to a MCQ is to mark 1 for the correct answer and 0 for the other options. This strategy allows students who make blind guesses or give random responses to all questions to obtain a score that may be evaluated as the number of questions divided by the number of distractors used: this means that a lucky student who is submitted to a test with 100 MCQ with 4 distractors may obtain a score of up to 25. Another approach called negative marking, assigns 1 for the correct response, 0 for no response and -1/(n-1) for an incorrect response. With this approach, a student who knows nothing, and therefore makes completely blind guesses may be marked with the plausible score of "about" zero. Obviously, a TMS should allow both of these marking schemes. For short answer questions the scoring scheme could either take into account or ignore letter case. Furthermore, it could prove useful to find a phrase inside an answer rather than considering the whole answer. The TMS should allow both features. For hotspot questions it should be possible to associate different scores to different areas of the image containing the information to be identified.

A question should provide feedback containing the mark to the given response along with comments reflecting the user's performance. The feedback could be presented either after any single question (this solution being preferable for self-evaluation tests) or at the end of the test and may be based on the overall performance.

Last, but not least, the inclusion of multimedia, such as sound and video clips or animated images, may improve the level of comprehension of a question.

**Test Management.** Among the issues that qualify a TMS with respect to Test Management, we suggest taking into account:

- the way in which exams can be prepared (test preparation);
- the availability of features for importing test banks that may be used to simplify the task of test preparation (test banks);
- the tools available to the teacher to evaluate the test (test evaluation tools);
- the tools available to the teacher to analyze the responses produced by the students (response analysis).

**Test preparation.** Once questions have been defined, they should be selected and organized into tests. Test preparation is a non-trivial task, since it requires the ability both to manually choose the questions from the database and to construct the exam through a random selection approach. This implies that questions could be selected with respect to different objectives as for educational goals, keywords, contents, statistical value and so on.

Furthermore, to build adaptive tests represents an important "add-on" for the selection of the TMS. Adaptive testing is used to allow the student to move forward or backwards in a test depending on his or her performance. This is a very powerful feature, since it allows the creation of material reacting "intelligently" to what the student does. Very few commercial TMS provide adaptive testing features (FastTest Pro, 2001; Perception, 2001) and usually the construction of adaptive tests is not very simple from the instructor's point of view since it requires the use of a sort of programming language to cope with the possible actions to be enacted according to the student's responses.

Moreover, it should be possible to create multiple forms by rearranging questions, either by some instructor intervention or automatically, in order to discourage cheating. Tools that provide the ability to randomize the order of answers for a question may further discourage cheating.

Finally, some countermeasures should be provided to prevent testing from turning into a guessing activity. This result could be obtained by introducing penalties for guessing, and or by adopting "restriction" functions to specify that no other testing attempt could be made within a given time span. This way the student may be allowed to reflect on his/hers mistakes.

**Test banks.** Questions can be assembled in a bank that is further referenced by the test. Test banks are very useful in a number of ways, since organizing questions related to the same topic in a bank may simplify both the random selection of questions and the evaluation of the understanding of the topic through statistical measures. Moreover, the same bank can be shared by different tests. This last point suggests that it is possible to reuse the same material, saving time and effort. Obviously, different instructors may share the same questions thus obtaining synergies and homogenizing the way in which the same topic is assessed in different courses. Building well-formed questions is an arduous task. The possibility of accessing question banks provided by well-known scientists or by professional organizations is significant for the educational community. As an example, we can cite the effort made by a number of Student Chapters of the Association for Computing Machinery that are collecting test banks related to computer science (ACM-SC, 2001).

Therefore, a TMS should provide the possibility to create multiple banks with an unlimited number of items in each bank, and the ability to import questions and corresponding data from existing banks.

**Test evaluation tools.** Tests should be evaluated both before and after administration (Gronlund, 1985).

Evaluating a test before administration means analyzing it in terms of adequacy of test plan, text items, and text format and directions. From the point of view of test plan, analyzing a test means finding an answer to the following questions:

- does the test plan adequately describe the instructional objectives, and the contents to be measured?
- does the test plan clearly indicate the relative emphasis to be given to each objective and each content area?

Each test item needs to be evaluated in terms of appropriateness, relevance, conciseness, ideal difficulty, correctness, technical soundness, cultural fairness, independence and sample adequacy.

Finally, for the test format and directions, analyzing a test means, for example, finding an answer to the following questions:

- Are the test items of the same type grouped together in the test or within sections of the test?
- Are the correct answers distributed in such a way that there is no detectable pattern?
- Is the test material well-spaced, legible, and free of typographical errors?

Evaluating a test after administration helps a) to verify whether it worked as intended in order to adequately discriminate between low and high achievers and b) to discover whether the test items were of appropriate difficulty and free of irrelevant clues or other defects (for instance, all distractors behaved effectively in MCQ).

**Response analysis.** Once questions have been devised and the test delivered, it is of fundamental importance to obtain an assessment of the students individually and with respect to the class. We have already discussed the importance of providing the instructor with some tools for the assessment of the evaluation

process. To attain such results, the TMS should provide the instructor with at least the following information:

- test performance report for each individual examinee, with the percentage of correct answers and relative ranks;
- individual response summary by item, with an error report that lists wrong vs. correct responses;
- class test performance with distribution, means and deviations;
- item statistics and analysis with indicators that may be useful to evaluate the questions in terms of reliability, discrimination and difficulty.

Although the system may provide some numerical results to measure the test, the responsibility of evaluating them and to identify strategies and policies to modify the educational process in order to improve the understanding of mis-concepted topics is left to the instructor.

Most TMS provide built-in facilities for the analysis of responses. More sophisticate analyses can be carried out via optional external modules. As an example, Assessment System Co. delivers a large set of different programs both for item and test analysis. “These programs are based on classical test theory, on Rasch model analysis using the 1- 2- and 3-parameter logistic Item Response Theory (IRT) model, on nonparametric IRT analysis, and on IRT analysis for attitude and preference data” (Assessment System Co, 2001).

**Suitability of Test Delivery System**

A TDS is a tool for the delivery of tests to the students. We decided to adopt the Question and test management capabilities to evaluate the Suitability. Question management relates to all aspects concerning questions handling while test management relates to test delivery.

To evaluate the question management unit of a TDS we selected:

- the ability to provide multiple attempts at solving a question (Retries);
- the existence of feedback and tutorials on the topic covered by the questions (Feedback & Tutorials);
- the capability of including multimedia (Multimedia).

**Retries.** Retries are the ability to allow multiple attempts to answer a question. Obviously, this ability is of great importance for self-assessment, since it may be useful to improve the knowledge of the student while reducing the need to provide feedback and/or tutoring.

On the other hand, the inability to change the answer to a question during an examination is often perceived as unfair. According to a study conducted by King et al. (1998) on the evaluation of a CAA protocol, about 34% of the students providing adverse comments needed the ability to repeat/retry responses. However, multiple attempts at question answering may affect the use of adaptive systems whenever item presentation depends on previous responses. On the other hand, retries may represent a vehicle for cheating as will be discussed in section 4 of this paper.

**Feedback & Tutorials.** Feedback and tutorials are related to the ability to provide information to the student once the answer to a question has been given. The feedback may be provided after each question (this solution being preferable for self-assessment), after a set of questions covering a given topic, or at the end of the test, and can be based on the overall performance. Furthermore, the feedback may be used to indicate the correctness of the answer, to correct mis-conceptions or to deliver additional material for deepening and/or broadening the coverage of the topic assessed by the question. Tutorials represent an extended approach to providing additional information to the students. The existence of some facility for

easy inclusion of tutorials in the TDS represents an important feedback aid. As an example, Perception provides explanation-type questions that may be used for "information screens, title pages, or to display large bodies of text" (Perception, 2001).

**Multimedia.** The use of questions incorporating multimedia, such as sound and video clips or images, may improve the level of knowledge evaluation. This aspect may be of great importance, for example, in language assessment, where the comprehension of a talk or a movie can be assessed through multimedia only. The use of multimedia can raise issues related to portability and interoperability since it may require special hardware and software, both for the server delivering the questions and for the client used by the students. These issues may not represent a problem whenever a Web-enabled TDS is selected, since the nature of the WWW is inherently multimedial. In this case, the choice of standard plug-ins for the most common browsers may reduce risks of portability and of interoperability.

**Test Management.** Among the issues used to evaluate the test management unit of a TDS are the ability to make tests available at a given time (Restricted Availability) and the grading capabilities (Grading).

**Restricted Availability.** Tests can be made available at a specified date and time. They can equally be made unavailable at a different date and time. This allows test designers to specify exactly when people can access a test. It should be possible to leave out either or both the restrictions to provide maximum flexibility. This lends itself nicely to the computer lab setting where students are required to complete an on-line test during a specified time frame on a specified day.

Restricted availability may raise some concerns with respect to the policies for handling borderline situations that will be discussed in the next section of this paper.

**Grading.** Obviously, any software for assessment should be able to compute student grades. Furthermore, grades must be delivered as feedback to the course coordinator, to the instructor and to the students. Each of these categories of users needs to obtain a different kind of feedback on the grades associated with a test. For instance, a student needs to know where he/she stands with respect to other students and to the class average besides his/her own individual and cumulative grades. This need raises obvious concerns about privacy that may be handled through the security facilities provided with the assessment tool.

## Security

Security is the quality factor dealing with those attributes of software that "bear on its ability to prevent unauthorized access, whether accidental or deliberate to program or data". The Test Delivery unit is by far the most vulnerable component, since it represents the interface to the "external world" of a CBA system. The security of a TDS directly impacts the process of assessing the competence of the students. This is the reason why we postpone the discussion on this issue to the section on Cheating.

## Interoperability

The last area of functionality is interoperability. Communication with other software is useful both for exporting answers and for calling external applications. Exporting answers is usually performed through test files and data conversion utilities to customize the reports generated by the application or whenever an analysis more detailed than the one provided by the assessment tool is needed to evaluate the results obtained.

Many available tools enable the calling of a program as a block within a question. The called program returns a score in points that may be added to the test score. This tool may be useful for assessing abilities that cannot be evaluated through the basic question-answer paradigm of most assessment tools.

Some tools allow external applications to be called at the very end of the test phase for printing certificates for all users who pass the test; for the electronic submission of the answer file to a central location

for analysis and evaluation, or for the storage of the results in a file to be accessed by a user program (Perception, 2001).

Finally, communication with other software is required in order to allow the integration of TDS and TMS distributed by different commercial producers.

## *Usability*

Usability addresses the relationship between a software tool and its users. It represents an important aspect for the evaluation of CBA systems since they are designed to be used by teachers and students without specific background knowledge in computer science. Thus, usability can make the difference between performing assessment accurately and completely or not, and enjoying the process or being frustrated.

Although there is a lot of work in the literature on the criteria to be adopted for the evaluation of the User Interface (UI) from the point of view of usability (see for instance Nielsen & Molich, 1990 and Gilham et al., 1995), this issue appears to be systematically overlooked in the evaluation of educational software. We strongly believe that the evaluation of the interface is a qualifying aspect for the evaluation of both subsystems of a CBA tool. This is true if we take into account the fact that neither the teacher nor the students may have advanced computer skills.

As Nielsen & Molich (1990) simply proposed, the interface must be easy to learn, efficient to use, easy to remember, error free and subjectively pleasing. Furthermore, the UI must speak the users' language. The European Union (EU) comprises eleven official languages plus a large number of national-specific versions and of regional languages. Additional language requirements are issued by the European Free Trade Association involving four more countries and by Eastern Europe. It is obvious that the assessment process of users with different languages should be made according to a chosen language and in a familiar cultural environment (taking into consideration the cultural bias or acceptability of icons, key words, etc.). The availability of features allowing users to switch among different languages, yet maintaining the same assessment capabilities would be very valuable. This aspect may be very interesting for educational institutions providing cross-countries learning material (CEN/ISSS-WS/LT, 2000).

## Operability and Understandability

### Operability and Understandability of Test Management Systems

**Ease of Editing.** A TMS should be designed so that questions and tests can be written in a simple and easy way. The ease if editing can be enhanced through the existence of a GUI that provides standard features such as a "wyswyg" editor, a clipboard and cut-and-paste and undo operations. At the same time, the possibility to include text, graphic images for diagrams and properly displayed mathematical, chemical or other symbols is of great importance for the instructor.

Moreover, the existence of spelling and grammar checking may greatly improve the ease of editing of a TMS by helping the instructor to build well-formed questions. The existence of ad-hoc dictionaries tailored to the domain may represent a plus to improve the ease of editing.

Finally, another criterion that is useful to evaluate the ease of editing is related to the "programming" abilities required to the instructor. The usability of the system may be dramatically reduced whenever the tutor is required to possess HTML, XML, Perl/CGI, Java or JavaScript knowledge.

### Operability and Understandability of Test Delivery Systems

**Accessibility.** Accessibility is used in this context as the usability of information systems by persons who cannot use the standard text and image based computer interaction.

The United Nations estimates that approximately ten percent of the population of a country has some sort of disability (impairment). These data vary considerably from country to country, rising to 25% of the population if moderate forms of sight and hearing losses are taken into account. Thus, the EU is funding cross-programme themes in the fifth framework programme for research, aimed at improving the accessibility of ICT systems.

Accessibility plays a crucial role for TDS since it affects the possibility of granting equal opportunities to students (CEN/ISSS-WS/LT, 2000).

**Feedback.** This item is related to the ability to provide information to the student once the answer to a given question has been entered. Feedback has been discussed in more detail in the section 3.1.1.2.

**Clear Marked Exits.** King et al. (1998) report that about 6% of students providing adverse comments (7 out of 112) to an evaluation of a CAA protocol, addressed the problem of obtaining an end-screen to be sure of having answered all questions. Thus, the operability of a TMS may be improved by clearly identifying the end of the assessment procedure.

### Learnability

An online help system providing some sort of tutorial on how to build questions and to prepare exams may greatly improve the ease of editing of the TMS. FastTest Pro(2001) by Assessment System Co. represents a good example of a system showing such features.

Finally, the existence of a training package aiding the instructor in creating good objective tests represents a very important add-on for the selection of a TMS. An instance of a test building support utility is represented by “Better Testing” developed by Question Mark Computing Ltd. (2000) and sold separately with respect to the CBA System.

### Reliability of the Software

The ability of a system to perform under adverse conditions may be of great importance for a Test Delivery System. In particular it is important that no termination procedures should result in any loss of data. To ensure this, both student and system files should be updated after each transaction, so that no data is lost if the test is terminated because of machine or power failure (Ring, 1994). With respect to this issue, a TDS should collect at minimum the following data for each test: student identifier, question identifier and student’s response. This is the minimum amount of data needed to reconstruct the performance of the student. The possibility of providing examination printouts may further enforce the survivability of the system.

Finally, after a crash, the system should be able to restart from the point of termination with all aspects of the original status unchanged, including the answers already given and the clock still displaying the time remaining.

## Cheating

The term “cheating” is used to address dishonest practices that students may pursue in order to gain better grades. Copying from books and assignments set in previous years, collusion among students in preparing assignments, getting help from relatives, using illegal notes in tests, sending colleagues to take one’s place in assessment and copying during classroom tests are just some examples of assessment dishonesty.

According to the literature, cheating is practiced by students at all levels of schooling, ranging from “approximately 40% in the upper primary year to nearly 80% in the latter years of secondary school falling to approximately 40% again in tertiary institutions” (Godfrey and Waugh, 1998). This old problem has new life with the widespread use of computer and web based assessment. Many researchers suggest that

this phenomenon can be discouraged, although not entirely prevented, by using certain simple practices such as informing students of the penalties for cheating and enforcing those penalties; ensuring that seating arrangements in examination and testing centres are adequate so as to prevent cheating; and being aware that cheating seems more likely to occur in larger classes than in smaller classes. Teachers can also assist in discouraging cheating by being aware of the high frequency of the phenomena and acknowledging the pressures under which many of these students are working. They must be patient and caring in their approach and make certain that students know that they can come to them for help or assistance and that some students may require more attention at times than others. Parents, of course, can assist in discouraging cheating by ensuring that their children are not overly pressured in their academic endeavors. (Godfrey & Waugh, 1998)

In this section we will discuss cheating control from the technical point of view, presenting some requirements that should be satisfied either at the component or at the system level of a TDS. We will also discuss how an attempt at controlling cheating may affect the interface, the question management and the test management functional blocks of a TDS. Then we will discuss the effects of cheating control on the security of a TDS.

Any system should attempt to ensure that any given student takes the right test at the right time and that the right student takes the test. The latter task may be solved only through organizational countermeasures and will be discussed at the end of this section. The former task is not difficult and is usually handled by asking students for their name and/or an identification number. The previous remark implies that the interface of a TDS should be designed so that access control could be enforced. This implication becomes less trivial than how it may appear at a first glance, if we take into account the fact that access control should be enforced by the teacher too, in order to avoid unauthorized access to tests before they are administered. Most systems actually on the market allow three classes of users to access the system: Student, Teachers and Administrators, each with different privileges and allowed functions.

There is a wide range of security issues from the point of view of Question Management. Among these issues are security of the availability of the test material and of the HTML code that implements testing. For HTML code, commercial programs usually implement encrypting approaches, a lot of issues should be taken into account for freeware. In fact, most freeware applications rely either on Perl/CGI or on JavaScript. The use of CGI-based application may raise an important issue since a CGI program is executable; it is basically the equivalent of letting the world run a program on the server side, which is not the safest thing to do. Therefore, there are some security precautions that need to be implemented when it comes to using CGI based applications. The one that will probably affect the typical Web user is the fact that CGI programs need to reside in a special directory, so that the server knows to execute the program rather than just display it to the browser. This directory is usually under direct control of the webmaster, prohibiting the average user from creating CGI programs.

On the other hand, since the JavaScript code runs on the client's side of the application, the obvious drawback of this approach is that the assessment program cannot be completely hidden, and a "smart" student can access the source discovering the right answer associated to each question. Some advanced coding techniques can be used to partially overcome the problem, which can be reduced to a minimum: for instance, Hazari (1999) suggests overcoming this problem by using "cookies" to hide answers from users.

Some TDS provide the ability to scramble the answers, so that the same question is never submitted in the same examination with the answers in the same position. In order to obtain well formed questions, answers like "None of the above" or "All of the above" should be avoided in multiple choice questions as suggested in the literature (Gronlund, 1985). Obviously the previous considerations hold for multiple choice and for multiple answer questions only, while they do not make sense for short answers, essays or hot-spot questions.

Another aspect that may affect cheating from the point of view of Question Management is the possibility of attempting multiple responses to the same question that we addressed as the "retries issue" in the previous section of this paper. In fact, students may try to access all the hints provided to questions, and then backtrack through the pages only to proceed again as if they have never seen them (and thus not losing any marks for seeing them). In order to avoid this drawback, the test designers of WebTest (1996) are provided with the ability to disable backtracking. This solution raised a number of problems (for instance the need of appropriate warning messages to be issued to inform the user not to click Back or Reload), including the fact that clicking the Reload button has the same effect as moving backwards and forwards thus corrupting the test again.

Most TDS provide the ability to scramble the position of questions within a test. This raises the concern that questions related to the same topic may be spanned around, thus implicitly increasing the level of difficulty of the test, and therefore representing a sort of unfairness to students. Furthermore the fact that question scrambling may interfere with adaptive testing where the set of items that constitute the exam is not predefined and depends on the students' performance level must be taken into account.

As we discussed earlier, restricted availability of the tests may prove useful to ensure that a given student takes the right test at the right time. Obviously, constraining the time limits for the execution of a test imposes both functional and non-functional requirements on the architecture of the TDS. To note time limits, both the possibility of displaying a clock with the residual time available and the existence of appropriate warning messages as the time limit approaches are important. There needs to be policies for handling "border-line" situations e.g.: what should happen to the student who does not complete the test on time? Should a student's test terminate and be handed in automatically? Or should the student be allowed to finish the test and hand it in himself under the assumption that the test-administrator will eventually make him/her leave?

The existence of features for locking out access to the operating system may be useful for preventing cheating if the Test Delivery System is running locally or over a LAN. In fact, another issue to be taken into account is the possibility of copying tests from the workstations. Printing and saving browser information on a disk is done through the caching feature. By disabling the cache system, it is possible to prevent students from making unauthorized copies of tests they are taking. Implementing the «kiosk» mode available for most major browsers prevents copying the text from the browser, using email or accessing any other applications.

Some TDS are designed to hand the test in for marking via e-mail. This raises

*"the concern that students may catch on to the format of the results email and attempt to create a fake one (naturally with very good overall results). It is possible to detect such email messages by paying close attention to things such as the user-id, when, and where it was emailed from, etc., however, that requires a lot of awareness from those administering the test. To prevent this situation, the test designer can specify a verification code, or secret code, to be used with each test. The code is only included in the email message that is sent to the administrator. It is impossible for students to find out what this code is as long as the problem files are not accessible to the general public" (WebTest, 1996).*

It is vital to remember that IP packets may be intercepted and read with relatively common technical knowledge and tools. Tests transmitted by the TDS could thus be stolen. A possible solution to avoid this problem may require adding data encryption-decryption features to the TDS.

Ensuring that the right student takes the test cannot be handled in a cost-effective way without human intervention. Therefore, the following discussion is independent from the software adopted but is related to the organizational aspects of Computer Based Assessment. For students doing the test on site and under supervision, the procedures are the same as for a conventional test. If students are taking the tests at remote locations some form of human supervision is normally required. Most educational organizations

address this issue by asking students to arrange for their tests to be proctored by an approved education agency and thus paying any proctoring fees. Approved agencies include a college testing center or the office of a public or private school administrator. Working with small classes is referenced in the literature as a good starting point to reduce cheating (Davis et al, 1992).

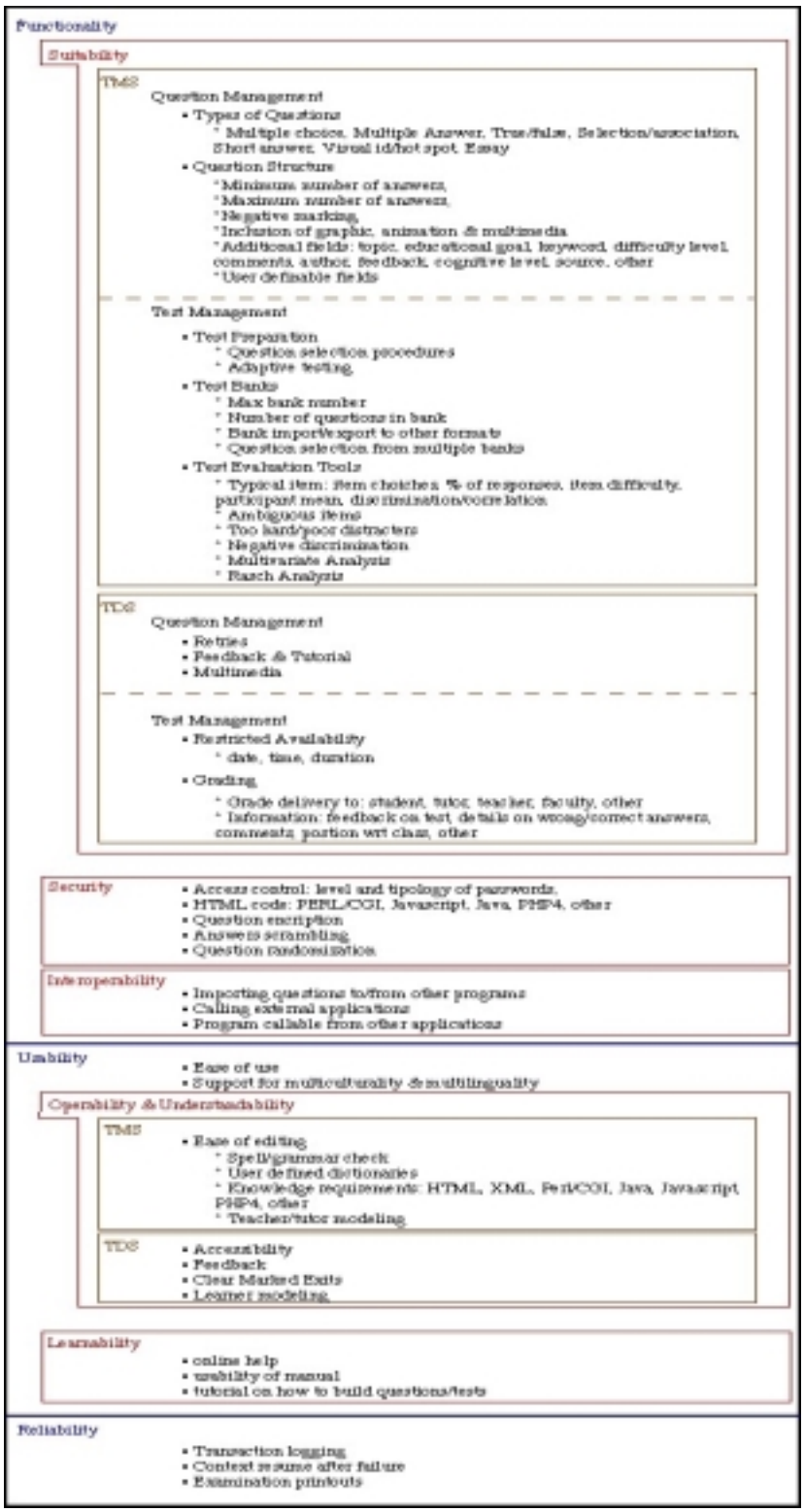



**Table 4 – Domain specific quality factors for the evaluation of CBA systems**

Using alternative assessment methods that do not rely on multiple choice questions can further discourage cheating. For example short answers or filling the blanks question types seem to be less subject to cheating. Furthermore, assigning each assessment worth only a few points can be a good countermeasure for controlling the pressure to cheat.

Godfrey and Waughn (1998) discuss a list of other issues that should be taken in account to reduce/prevent cheating.

## Final Remarks

The interest in developing Computer Based Assessment systems has increased in recent years, thanks to the potential market of their applications. Many commercial products, as well as freeware and shareware tools, are the result of studies and research in this field made by companies and public institutions. Such a large number of available assessment systems obviously raise the problem of identifying a set of criteria that may be useful to an educational team wishing to select the most appropriate tool for their assessment needs.

In this paper we have discussed some quality factors for the evaluation of a CBA system defined according to the ISO 9126 standard for the evaluation of software quality characteristics and subcharacteristics, a model for product assessment that identifies six quality characteristics: functionality, usability, reliability, efficiency,

maintainability and portability.

The discussion has been focused on those criteria that are domain specific (namely functionality, reliability and usability) and therefore highly dependent on the educational domain to which CBA systems belong. Table 4 summarizes the list of quality factors identified in the paper and classified according to the framework depicted in table 3.

The remaining quality factors (efficiency, portability and maintainability) that are independent from the educational domain can be evaluated through the checklists supplied in the annex A of the IEEE Recommended Practice for Software Acquisition Standard.

As a follow-up of this work the list of quality factors identified in the paper will be hosted as a form on the web site of our department and made available to all researchers wishing to review a CBA system. All the items of the list involving an evaluation by the reviewer will allow stating both agreement and disagreement via a 5 + 2 Likert Scale as suggested by Pilj (1996). This scale, "which has the strengths of a 5 point scale, yet has two additional selections having nothing to do with the discriminate scale but address the validity of the question, and maintains the integrity of the answers. The points will be 1. Strongly Agree - 2. Agree - 3. Neutral - 4. Disagree - 5. Strongly Disagree - 6. I don't know - 7. I don't understand the question". The obtained results will be made available to all interested parties.

## Acknowledgements

This work is derived from a critical revision and rethinking of two papers (Valenti et. al, 2000 and Valenti et al 2002).

We would like to thank Karen Nantz for her invaluable suggestions and help in obtaining the final version of this paper.

## References

ACM-SC. (2001). Retrieved Dec 9, 2001 from the World Wide Web http://www.cs.uidaho.edu/~acm/test_bank.html , http://www.utdallas.edu/orgs/acm/testbank.html

Anderson E.E. (1989). A heuristic for Software Evaluation and Selection. Software Practice and Experience, 19(8), 707-717.

Ares Casal J.M., Dieste Tubio, O., Garcia Vasquez R., Lopez Fernandez, M., Rodriguez Yanez S. (1998). Formalizing the software evaluation process, Proc. SCCM'98 – 18th Int'l. Conf. of the Chilean Society of Computer Science, IEEE, 15-24.

Assessment System Co. (2001). Retrieved Dec 9, 2001 from the World Wide Web http://www.assess.com/

Bloom B. (1956). Taxonomy of Educational Objectives, Handbook I, Cognitive Domain, New York, David McKay Co. Inc.

Bull J. (1999). Computer-Assisted Assessment: Impact on Higher Education Institutions, Educational Technology & Society, 2(3).

Burstein J. & Chodorow M. (1999). Automated essay scoring for nonnative English speakers. Retrieved Dec 9, 2001 from the World Wide Web http://www.ets.org/research/acl99rev.pdf

CEN/ISSS-WS/LT (2000) - Learning Technologies Workshop A Standardization Work Programme for "Learning and Training Technologies & Educational Multimedia Software", CWA. Retrieved Dec 9, 2002 from the World Wide Web http://www.cenorm.be/isss/Workshop/LT/draft-final-report/cwa4-5.pdf

Charman D. and Elmes A. (1998). Computer Based Assessment (Volume I): A guide to good practice. SEED Publications, University of Plymouth.

Cogent Computing Co. (2001), Retrieved Dec 9, 2001 from the World Wide Web http://cqtest.com/

CQuest (2000). Cogent Computing Co., Retrieved Dec 10, 2001 from the World Wide Web http://www.cogentcorp.com/

Cucchiarelli A., Panti M., Valenti S. (2000). Web-based assessment of Student Learning, in Web-Based Learning and Teaching Technologies: Opportunities and Challenges, A.K. Aaggarwal ed., 175-197, Idea Group Publishing.

Davis S.F., Grover C.A., Becker A.H. and McGregor L.N. (1992). Academic Dishonesty: Prevalence, determinants, techniques and punishments, Teaching of Psychology, 19(1), 16-20.

FastTest Pro (2001), Assessment System Corporation. Retrieved Dec 11, 2001 from the World Wide Web http://www.assess.com/FastTEST.html

Foltz P.W., Laham D. & Landauer T.K. (1999). Automated Essay Scoring: Applications to Educational Technology, in Proc. of EdMedia'99. Retrieved Dec 12, 2001 from the World Wide Web http://www-psych.nmsu.edu/~pfoltz/reprints/Edmedia99.html

Freemont D.J., Jones B. (1994). Testing Software: a review, New Currents 1.1, Retrieved Dec 7, 2001 from the World Wide Web http://www.ucalgary.ca/pubs/Newsletters/Currents/Vol1.1/TestingSoftware.html

Gibson E. J., Brewer P.W., Dholakia A., Vouk M.A., Bitzer D.L. (1995). A comparative analysis of Web-based testing and evaluation systems, Proceedings of the 4th WWW conference, Boston.

Gillham, M., Kemp, B. & Buckner, K. (1995). Evaluating Interactive Multimedia Products for the Home, The New Review of Hypermedia and Multimedia Vol.1, p. 199-212.

Godfrey J.R. and Waugh R. F. (1998). The perception of students from religious schools about academic dishonesty, Issues in Educational Research, 8 (2), 95-116.

Grondlund N.E. (1985). Measurement and Evaluation in Teaching, Macmillan Pub. Co., NY.

Hansen, W.J. (1999). A Generic Process and Terminology for Evaluating COTS Software, SEI, Retrieved Dec 8, 2001 from the World Wide Web http://www.sei.cmu.edu/staff/wjh/Qesta.html

Hazari, S. I. (1998). Online Testing Methods for Web Courses. Proceedings of the 14th Annual Conference on Distance Teaching and Learning, 155-157.

Henderson, R.D., Smith M.C., Podd J., Varena-Alvarez H. (1995). A comparison of the four prominent user-based methods for evaluating the usability of computer software. Ergonomics, 38(10), 2030-2044.

HitReturn (2000), Retrieved Dec 10, 2001 from the World Wide Web http://www.hitreturn.com/index.htm

InQsit (2000). Ball State University. Retrieved Dec 10, 2001 from the World Wide Web http://www.bsu.edu/inqsit/

ISO (1991). Information Technology – Software quality characteristics and sub-characteristics. ISO/IEC 9126-1.

IEEE (1993). IEEE Recommended Practice for Software acquisition. IEEE Std 1062-1993.

King T., Billinge D., Callear D., Wilson S., Wilson A. and Briggs J. (1998). Developing and evaluating a CAA protocol for University Students, Proc. of the 2nd Annual Computer Assisted Assessment Conference, Loughborough, UK.

Looms T. (2001). Survey of Course and Test Delivery / Management Systems for Distance Learning, Retrieved Dec 8, 2001 from the World Wide Web http://www.student.seas.gwu.edu/~tlooms/assess.html

Nielsen, J. & Molich, R. (1990). Heuristic evaluation of user interfaces, Proceedings of CHI'90, ACM, 249-256.

Nikoukaran J., Hlupic V., Paul R.J. (1999). A hierarchical framework for evaluating simulation software, Simulation Practice and Theory, 7(3), Elsevier, 219-231.

Perception (2001), Question Mark Computing Ltd. Retrieved Dec 12, 2002 from the World Wide Web http://www.questionmark.com/us/perception/index.htm

Question Mark Computing Ltd. (2001), Retrieved Dec 9, 2001 from the World Wide Web http://www.questionmark.com/us/home.htm

Pilj, Gerald H. Lessons from the Software Engineering literature (1996). Retrieved Dec 8, 2001 from the World Wide Web http://www.phys.ksu.edu/~pilj/article.html

Ring G. (1994). Computer administered testing in an IMM environment: Research and development. Proceedings of the Second International Interactive Multimedia Symposium, McBeath C. and Atkinson R. eds., 478-484. Perth, Western Australia, 23-28 January, Promaco Conventions. Retrieved Dec 10, 2001 from the World Wide Web http://cleo.murdoch.edu.au/gen/aset/confs/iims/94/qz/ring1.html

Valenti S., Cucchiarelli A., Panti M. (2000). "Some Guidelines to Support Tool Selection for Computer Assisted Assessment", in "Challenges of Information Technology Management in the 21st Century", Proceedings of the 2000 Information Resources Management Association International Conference, M. Khosrow-Pour ed., 609-613, Idea Group Publishing.

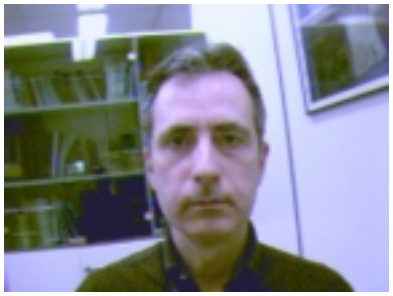
Valenti S., Cucchiarelli A., Panti M. (2002). “Relevant Aspects for Test Delivery Systems Evaluation”, in "Web-Based Instructional Learning", M. Khosrow-Pour ed. , 203-216, IRM Press, Hershey, PA, USA.

Vlahavas I., Stamelos I. , Refanidis I, Tsoukias A. (1999). ESSE: an expert system for software evaluation, Knowledge Based Systems, Elsevier, 4 (12), 183-197.

WebTest (1996), Retrieved Dec 12, 2002 from the World Wide Web http://fpg.uwaterloo.ca/WEBTEST/WEBTEST_intro.html

## Biographies

**Salvatore (Sal) Valenti** is senior researcher at the University of Ancona. He has been a member of several research projects funded by the Ministry of Instruction, University and Research (MIUR), by the National Research Council (CNR) and by the European Community. His research activities are in the fields of Computer Based Assessment and on Distance Learning. He is Board member of the JITE. He is serving as reviewer for Educational Technology & Society and for Current Issues in Education. He has been chair of the track on “Virtual Universities” at the 2002 International Conference of the International Resources Management Association. He is author of more than 60 papers published on books, journals and proceedings of international conferences.



**Alessandro Cucchiarelli** is senior researcher at the University of Ancona. His main research interests are focused on Automatic Evaluation of Software and on NLP techniques applied to Information Extraction. He has been also involved in research activities on Models and Tools for Cooperative Distributed Information Systems, Requirement Engineering and Robotics.He has been a member of groups working in several research projects funded by the Ministry of Research, the National Research Council and the European Union (Cost13, ECRAN). He is author of more than 70 papers published on books, journals and proceedings of international conferences.

**Maurizio Panti** is Associated Professor of Computer Science at University of Ancona. Previously he has been researcher at University of Urbino and University of Salerno (1971/74) and Assistant Professor at University of Ancona (1974/1984). Actually, he teaches Data Bases and Fundamentals of Computer at Universities of Ancona. He managed the Computer Centre of University of Ancona and now is member of the Academic Senate of the same University. His research interests concern, databases, information systems and agent technologies for IS integration. He is serving as member of program committee in international conferences as CAISE, WMC02, SEBD, CoopIS and served as referee in international journals.