

# Slide Test Maker

## An Educational Software Tool for Test Composition

Salvatore Cuomo<sup>1</sup>, Vittorio Fuccella<sup>2</sup>, and Aniello Murano<sup>1</sup>

<sup>1</sup> Dipartimento di Scienze Fisiche, Università di Napoli Federico II,  
Via Cinthia 26, 80137 Napoli, Italy

<sup>2</sup> Dipartimento di Informatica, Università di Salerno,  
Via Ponte Don Melillo, 84084 Fisciano(SA), Italy

{salcuomo@unina.it}

{vfuccella@unisa.it}

{murano@na.infn.it}

**Abstract.** In every day life education, tests play a key role. They represent a standard vehicle for assessing proficiency, measuring aptitude or determining skill and knowledge in any teaching field. In this paper, we present Slide Test Maker (STM), a software tool based on Java, whose main objective is to help teachers in a quick composition of online tests. Starting from a generic lesson in slide-show format, STM allows the user to easily *tag* the contents (texts, sounds, images, video, etc.) in the presentation, in order to construct the test, in a very easy and intuitive way. The composed test can then be saved and exported in PDF format and as a Learning Object in a standard format.

## 1 Introduction

Nowadays the assessment is a fundamental part of the instructional learning process. The summative assessment programs are used for grading and accountability purposes while the formative assessments are adopted to diagnose and modify the conditions of learning and instruction. The formative use of assessment occurs during the learning process, gives information on the learning state of the class, and allows the teacher to decide the most suitable learning path [15]. Software tools enable the support of many data collections, complex analysis, individualized feedback and scaffolding features needed for the formative use of assessment [10], [18], [14]. Online testing methodologies are used in several international assessment programs. For example, in 2009 the Programme for International Student Assessment (PISA) has included the electronic texts to test reading, and in 2006 PISA experimented an innovative computer-based assessment program in the science disciplines. Companies and non-profit organizations offer service for developing on line tests able to select skills and ability of the students. For example, in 2008 Educational Testing Service (ETS), a U.S. nonprofit educational institution, published a report in which it is observed that

4 million people will take ETS-developed tests on computer. Software tools, that implement *Computerized Adaptive Testing* (CAT), have been repeatedly shown to significantly benefit student achievement.

In the last decade, many software tools for online testing have been fruitfully developed. Among the others, we recall here *Xerte*, *eXelearning*, *Ispring*, *Flypaper*, *Docebo*, *Moodle*, *eLearning Place*, *CADATS*, and *SIETTE* (see [4], [7], [2], [8], [3], [11]).

As reported in a survey [13], online assessment has recently been used in several application areas, such as CAT, automatic question generation, analysis of log data, *m-learning* and *educational gaming*.

Online tests are often packaged in Learning Management System (LMS). A LMS is aimed at managing an e-learning environment, establishing the organization and delivery of content, administrating resources and tracking learning activities and results [9]. Unfortunately, several LMSs are designed in a general purpose way and it is possible to observe that there is an unnatural separation between the teaching and the assessment phases. Usually, teachers have to refer to two or more tools for addressing both aspects. Also, tests are often built from scratch, without taking any advantage from the digital material produced during the previous teaching phase. Most of the research carried out so far has taken into account the fully automatic generation of questions from an input text [19, 12, 17] and the generation of tests through the selection of items from an existing question repository [20, 16]. Nevertheless, the systems for automatic question generation are still far from giving satisfactory results. In fact, after the generation, the human intervention is anyhow necessary for verifying the good sense of the items before using them in a test [13]. Thus, if a user wants to create a test based on his/her syllabus, s/he has to perform the tedious process of copying the content from the teaching material (slide presentations, in most cases) and pasting it to the form-based interfaces used to compose tests.

The above motivations led us to design an educational software tool, named *Slide Test Maker* (STM), aimed at assisting teachers in producing tests in a semi-automatic way, starting from multimedia data stored in Power Point or Impress presentations. The tool allows the user to easily *tag* the material in the presentation, in order to quickly paste it in the form-based interface used to compose the test. The interaction paradigm used to tag the content allows the teacher to avoid the tedious sequence of copy-and-paste interactions needed to fill the test composition form.

Furthermore, STM can produce the tests as Learning Objects (LOs) in a standard format, which can be executed in any standard compliant LMS environment.

It is worth noting that STM, being a tool that produces online tests, retains all positive aspects of similar tools developed in this field. Moreover, STM takes into account an important aspect for the students: the exercise and the self-assessment. In *formative assessment* [14], at the end of each lesson, the student practices on a test based on the lesson itself. During the test, the feedback given on each question has an important role. STM integrates in teacher's feedback

precise references to the slides from which the question is based. This way, a student can investigate any matters over which it has gaps without having to spend too much time looking for them in the teaching material.

The tool has been evaluated with pilot users. Interviews with them allowed us to enhance its design and to demonstrate that their level of satisfaction with the tool's usability and efficiency is high.

The rest of the paper is organized as follows: the next section describes the STM tool in details. In Section 3, we report the results of an informal evaluation of the software tool with pilot users. Lastly, Section 4 offers our conclusions and outlines the future work.

## 2 The STM Software Tool

Slide Test Maker (STM) is a software tool based on Java, whose aim is twofold: on one hand, it helps teachers to quickly compose online tests; on the other hand, it enables the packaging of tests and the educational material on which the test is based in a single standard LO.

Starting from a lesson in a slide-show format, STM allows the teacher to tag all contents (texts, sounds, images, video, etc.) slide by slide. The tagged content can be pasted to the test through a very easy and intuitive interaction paradigm.

The tool provides a graphical interface that assists the teacher to produce the online tests under the form of PDF (pencil and paper) or LOs. Therefore, the teachers can design new quizzes according to students study schedule and integrate the course material in it.

Through the main menu of the application, the user can execute the following main operations:

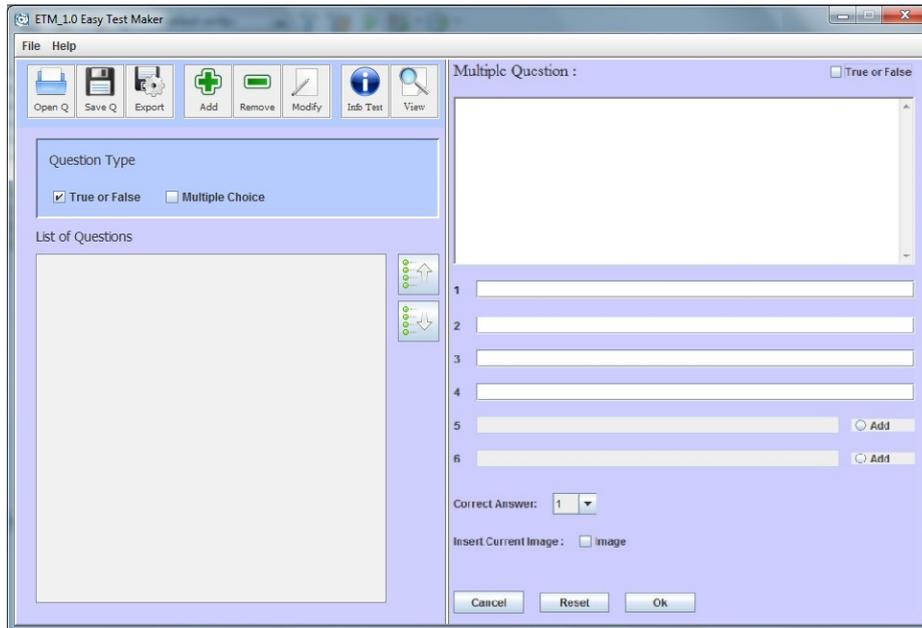
1. Creating/editing a test;
2. Creating questions from a slide show;
3. Exporting the test in a standard format.

### 2.1 Creating/Editing a Test

The interface to create and edit a test is simple and intuitive. A new test can be created (or an existing one opened) using the menu or the toolbar. The application interface is vertically divided into two views, each taking approximately half the application window. The *Test View* is on the left hand side, while the *Question View* is on the right hand side, as shown in Figure 1.

The *Test View* represents the current state of the test composition, reporting a summary of the main test information, including the number and the type of the questions currently inserted in the test. From the toolbar located above the view it is possible to save and export the test, and edit the test by adding a new question or deleting an existing one.

The *Question View* contains a simple form through which a new question can be created or an existing one edited. Two types of questions are supported



**Fig. 1.** The *STM* interface.

at present: *multiple choice* and *true/false*. A *multiple choice* item is composed of a *stem* and a list of *options*. The *stem* is the text that states the question. The only correct option is called the *key*. *True/false* items are a variation of multiple choice items where the stem contains a statement and only two options (*true* and *false*) are provided.

## 2.2 Creating Questions from a Slide Show

At any point during the editing of a test, the teacher can import a slide-show lesson, by means of the command *Open Slide Show*. As soon as this operation is performed, a pop-up window is opened. This view, called *Slide Show View*, allows the teacher to browse the slides of the presentation (see Figure 2): by using the toolbar buttons on the top of the new window, it is possible to move through the slides one by one or five by five.

The view shows a preview of the current slide. The slide elements, including text and multimedia content, can be interactively selected and used for the currently edited question of the test by directly *tagging* them. In particular, we adopted the following interaction design:

- *Element Selection*. Any element can be selected in the classic way: a mouse left-button click allows to select an element while the text can be selected by clicking and dragging the mouse pointer.

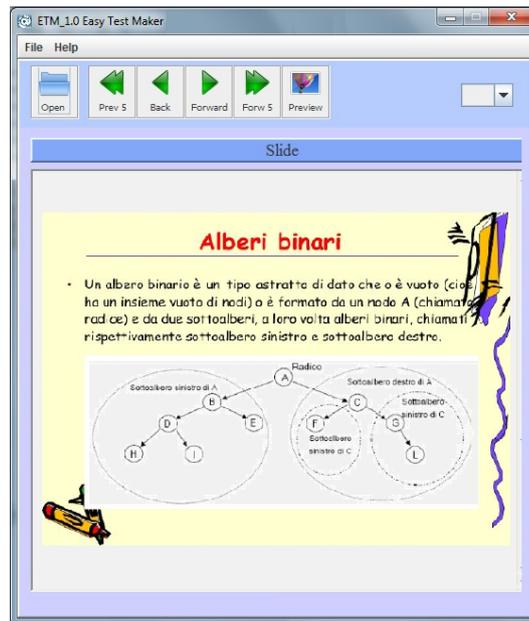


Fig. 2. The *Slide Show View*.

- *Composition of the question stem.* A mouse left-button double-click pastes the selected element to the form field representing the *stem* of the currently edited question in the *Question View*. A single multimedia element can be used in a stem.
- *Composition of the question options.* The creation of an option can be done through the mouse or through a keyboard shortcut: a mouse right-button click opens a pop-up toolbar allowing the user to choose the index of the option; pressing a number key on the keyboard, while a selection is active, is a shortcut to create an option: the number associated to the pressed key corresponds to the index of the option.

It is worth noting that the multimedia can be easily added to a stem as it is done with the text. This is different and much easier than classic test making software tools, where a multimedia is usually loaded by browsing the file system. This feature saves considerable time and effort for making a test.

The *Slide Show View* is shown in superposition to the *Test View*. In this way, the user can observe both the slides and the currently edited question in the *Question View*, at the same time. An example of execution of the tool where a text segment is pasted from the presentation to the text field of an option is shown in Figure 3.

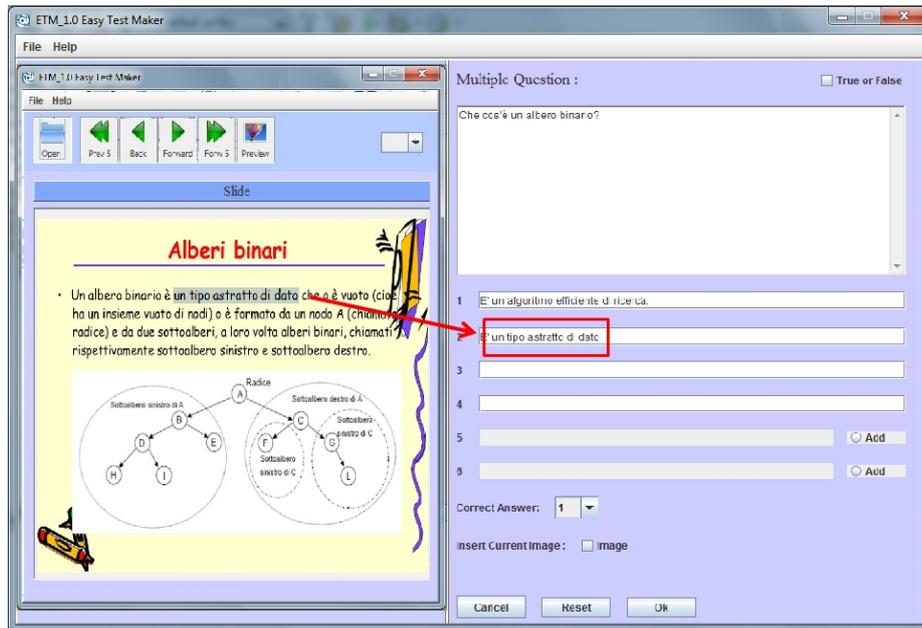


Fig. 3. Screenshot of the tool showing a text segment pasted from the *Slide Show View*.

### 2.3 Exporting the Test

Once all questions have been composed, the teacher can shuffle them in the desired order.

During the composition of a test, by means of the command *Save Questions*, the teacher can save the test produced so far. Finally, by means of the commands *Create PDF* and *Export to LO* it is possible to export a created test in PDF or in a standard LO format.

In particular, an LO is produced in compliance to SCORM 2004 [1]. The presentation files used to compose the test can optionally be packaged in the LO together with the test. The test is exported in the *IMS Question and Test Interoperability (QTI)* [6] standard format. During the production of the test, STM records the association of every single item to the slides of the presentation on which the question is based. Thus, using the *feedback* feature embedded in the QTI format, the slides can be referenced during the execution of the test by a student. This way, the student can deepen those topics where s/he has weaknesses without having to spend too much time looking for them in the teaching material.

STM also provides an environment in which the produced quizzes can be quickly tested. A screenshot of a test executed in such an environment is shown in Figure 4. At the end of the execution, a score is provided by STM. It is worth noting that if a question contains multimedia objects, only a thumbnail of it is

attached to the question. The object can be viewed in full size or launched in an external player through a simple mouse click.

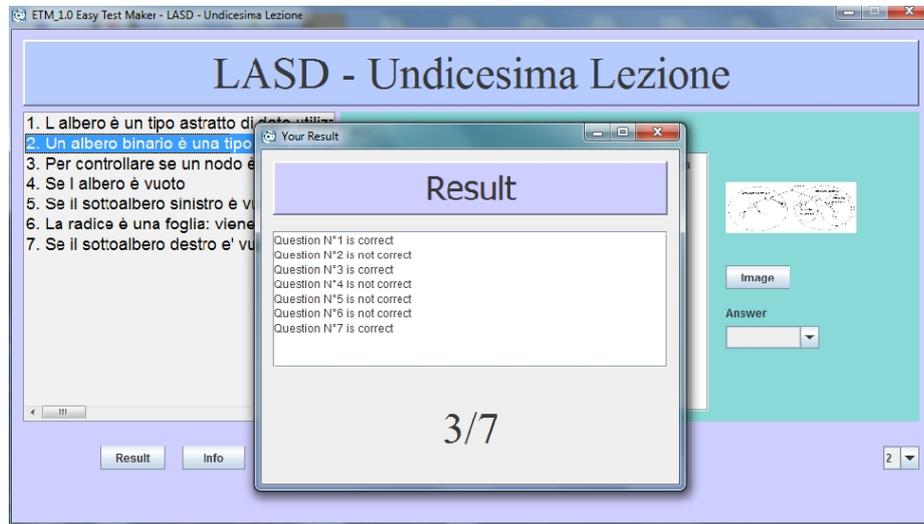


Fig. 4. The *STM* results

Further features supported (not reported in detail here) in *STM* are a detailed help tutorial and a digital and "pencil and paper" test output format.

### 3 Evaluation

*STM* has been evaluated with 5 (4 male, 1 female) pilot users, all of them being university professors (three computer scientists). The participants were asked to use the tool and then their judgment was gathered in an interview with the authors. As a main result, we were interested in assessing the interaction paradigm presented in Section 2.2 to test if it gives a significant advantage to the teacher in quickly producing a test from a presentation.

The evaluation was carried out in two rounds. The first round was aimed at obtaining feedback from users and modify the tool accordingly. The second round, instead, was aimed at validating the changes performed after the first round.

In the first round the users were invited to freely test the tool, with the only required task of creating a test from a presentation of their choice from the learning material of their course syllabus. According to the opinions expressed in interviews, we obtained a good level of satisfaction, with all the users being favorable to use *STM* in their daily teaching activities. The critical issues raised by

users were mainly related to the arrangement of the interface widgets. Furthermore, the users requested the development of an online help and the production of a short video demonstration showing the salient features of the tool.

In the second round, the users were provided with a short presentation and were instructed to create a small test composed of five questions. The users were invited to study the presentation for a while and write down on a paper the questions they felt most relevant to the lesson. After that, they were invited to compose the test with both STM and their favorite software tool. It is interesting to note that three users out of five used the same test making tool (*QuizFaber* v.2 [5]). The users confirmed the positive impression they had after the first round and noticed the greater efficiency of STM, compared to the other classic tools.

As a further interesting feedback obtained through the test with users, we report that a user appreciated the *Test View* interface, saying that it gave a better summary of the test information than analogous interfaces of other tools. Another user reported that the simultaneous display of both the *Test View* and the *Slide Show View* helped her in settling questions more related to the content of the lesson.

## 4 Conclusions and Future Work

In this paper, we have described the main features of the STM tool, designed to help teachers to compose online tests, based on textual and multimedia elements extracted from a slide-show lessons. As positive aspects of this tool, we mention the following: first, the teacher does not need to write the test from scratch; second, the tool allows to produce LOs that can be run in a standard compliant LMS.

Through a test with users, we got a positive feedback on the usability and efficiency of the tool. In particular, we had an encouraging positive judgment on the interaction paradigm we designed for quickly producing a test from a presentation.

As a future work, the support of more question types is planned. Furthermore, we are going to develop the tool as a production software and to distribute it under an open source license. Lastly, we are designing a client-server run time environment where the tests can be deployed and taken by the students.

**Acknowledgments.** We thank Christian Alfano for his help in the implementation of the STM software.

This research paper is partially supported by F.A.R.O. Project, founded by University of Naples Federico II and Compagnia di San Paolo Foundation.

## References

1. Adl scorm. <http://www.imsglobal.org/>.

2. Docebo. <http://www.docebo.com/>.
3. elearning place. <http://education.qld.gov.au/learningplace/>.
4. exelearning. <http://www.exelearning.it/>.
5. <http://www.lucagalli.net/>. QUIZFaber.
6. Ims question and test interoperability. <http://www.imsglobal.org/>.
7. Ispring. <http://ispring.com/>.
8. Moodle. <http://moodle.org/>.
9. Sandy Britain and Oleg Liber. A framework for the pedagogical evaluation of elearning environments. Technical report, February 2004.
10. J. Brown, S. Hinze, and J.W. Pellegrino. *T. Good (Eds.) - 21st Century Education. Vol 2. Technology*, chapter Technology and formative assessment, pages 245–255. Sage, Thousand Oaks, CA, 2008.
11. Ricardo Conejo, Eduardo Guzmán, Eva Millán, Mónica Trella, José Luis Pérez-De-La-Cruz, and Antonia Ríos. Siette: A web-based tool for adaptive testing. *Int. J. Artif. Intell. Ed.*, 14(1):29–61, January 2004.
12. Gennaro Costagliola, Filomena Ferrucci, Vittorio Fuccella, and Rocco Oliveto. eWorkbook: a computer aided assessment system. *International Journal of Distance Education Technologies*, 5 (3):24–41, 2007. IGI Global Inc., Hershey, PA, USA.
13. Gennaro Costagliola and Vittorio Fuccella. Online testing, current issues and future trends. *Journal of e-Learning and Knowledge Society (Je-LKS)*, 5 (3):83–93, 2009. Roma, Italia.
14. Salvatore Cuomo, Vittorio Fuccella, and Aniello Murano. Full formative assessment based on educational video games. In *Proceedings of DMS'10 (DET Workshop) - The 16th International Conference on Distributed Multimedia Systems*, pages 228–231, Oak Brook, Illinois, USA, Oct. 14-16 2010.
15. Paolo Frignani and Vincenzo Bonazza. *Le prove oggettive di profitto. Strumenti docimologici per l'insegnante*. Carocci, 2003.
16. Ayako Hoshino and Hiroshi Nakagawa. A cloze test authoring system and its automation. In *Proceedings of the 6th international conference on Advances in web based learning, ICWL'07*, pages 252–263, Berlin, Heidelberg, 2008. Springer-Verlag.
17. Gwo-Jen Hwang, Hui-Chun Chu, Peng-Yeng Yin, and Ji-Yu Lin. An innovative parallel test sheet composition approach to meet multiple assessment criteria for national tests. *Computers and Education*, 51(3):1058 – 1072, 2008.
18. Petros Lalos, Fotis Lazarinis, and Dimitris Kanellopoulos. E-snakes and ladders; a hypermedia educational environment for portable devices. *Int. J. Mob. Learn. Organ.*, 3(2):107–127, 2009.
19. Chin-Ling Lee, Chih-Hui Huang, and Cheng-Jian Lin. Test-sheet composition using immune algorithm for e-learning application. In Hiroshi Okuno and Moonis Ali, editors, *New Trends in Applied Artificial Intelligence*, volume 4570 of *Lecture Notes in Computer Science*, pages 823–833. Springer Berlin / Heidelberg, 2007.
20. Ruslan Mitkov and Le An Ha. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing - Volume 2, HLT-NAACL-EDUC '03*, pages 17–22, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.