# Monitoring On Line Tests Through Data Visualization

Gennaro Costagliola, Vittorio Fuccella, Massimiliano Giordano, Giuseppe Polese

***Abstract*—We present an approach and a system to let tutors monitor several important aspects related to on line tests, such as learner behavior and test quality. The approach includes the logging of important data related to learner interaction with the system during the execution of on line tests, and exploits data visualization to highlight information useful to let tutors review and improve the whole assessment process. We have focused on the discovery of behavioral patterns of learners, and conceptual relationships among test items. Furtermore, we have led several experiments in our faculty in order to assess the whole approach. In particular, by analyzing the data visualization charts we have detected several previously unknown test strategies used by the learners. Lastly, we have detected several correlations among questions, which gave us useful feedbacks on the test quality.**

***Index Terms*—Distance learning, Data visualization, Interactive data exploration and knowledge discovery.**

## I. Introduction

E-TESTING systems are being widely adopted in academic environments, also in combination with other assessment means, providing tutors with powerful tools to submit different types of tests in order to assess learners' knowledge. Among these, *Multiple Choice* tests are extremely popular, since they can be automatically corrected. However, many learners do not welcome this type of test, because often it does not let them properly express their capacity, due to the characteristics of *multiple choices* questions of being "closed-ended". Even many examiners doubt about the real effectiveness of structured tests in assessing learners' knowledge, and they wonder whether learners are more conditioned by the question type than by its actual difficulty.

In order to teach learners how to improve their performances on *structured* tests, in the past several experiments have been carried out to track learners' behavior during tests by using the *think out loud* method: learners were informed of the experiment and had to speak during the test to explain what they were thinking, while an operator was storing their words using a tape recorder. This technique might be quite invasive, since it requires learners to modify their behavior in order to record the information to analyze [1]–[5], which might vanish the experiment goals, since it adds considerable noise in the tracked data. Nevertheless, having the possibility of collecting data about learners' behavior during tests would be an extremely valuable achievement, since it would let tutors exploit many currently available data exploration and knowledge discovery strategies to elicit important insights of the testing activities that can be used to teach learners how to improve their performances. However, it would be desirable devising non invasive data collection strategies that do not influence learners behavior during tests, so as to convey more faithful feedbacks on the testing activities.

In this paper we present a solution enabling the recording of learners' habits during on-line tests without informing them of the underlying experiment, and consequently, without asking them to modify their behavior, which potentially yields more realistic results.

Regarding the exploration of the collected data, several knowledge discovery techniques could be used. Classical data mining algorithms aim to automatically recognize patterns in the data in order to convey knowledge [6], [7]. However, classical data mining algorithms become inappropriate in several situations such as multidimensional data and data not uniformly distributed. One way to overcome these problems is to use proper visual representations of data in order to stimulate user involvement in the mining process. In particular, information visualization can potentially enhance human capability to detect structures, patterns and relationships between data elements while exploring data. Information visualization is defined as the use of interactive visual representation of abstract data to amplify cognition [8]. In the past, information visualization has been successfully used in an e-learning application to measure the participation of the learners to on-line activities [9].

In this paper we propose a data exploration approach exploiting information visualization in order to involve tutors in a visual data mining process aiming to detect structures, patterns and relations between data, which can potentially reveal previously unknown knowledge inherent tests, such as the test strategies used by the learners, correlations among different questions, and many other aspects, including their impact on the final score.

Finally, we also present a system implementing the proposed approach. The system logs all the interactions of learners with the *e-testing* system interface. In particular, it captures the occurrence of question browsing and answering events by the learners, and uses these data to visualize charts containing a chronological review of tests. Other than identifying the most frequently employed strategies, the tutor can determine their effectiveness by correlating their use with the final test scores. Moreover, the proposed system helps tutors to detect correlations among questions: if a question contains a suggestion to solve other questions, this is made easily visible in the charts. Lastly, we give a first glance at the way our system can be used for detecting unethical behaviors from the learners, such as cheating by watching on others' screens

G. Costagliola, V. Fuccella, M. Giordano and G. Polese are with Università degli Studi di Salerno, Via Ponte D. Melillo, 84084, Fisciano (SA), Italia.

and *gaming the system* [10] attempts. The system is web based on the *AJAX* [11] technology in order to capture all of the learners interactions with the *e-testing* system interface (running in the Web browser). The system is composed of a logging framework that can be instantiated in any *e-testing* systems, and a stand-alone application that analyzes the logs in order to extract information from them and to graphically represent it.

In order to demonstrate the effectiveness of the approach and of the proposed system we have used them in the context of a university course offered in our faculty: the framework has been instantiated in an existing *e-testing* system, *eWorkbook* [12], which has been used to administer on-line tests to learners. The grade obtained on the tests has concurred to determine the final grade of the course exam. The rest of the paper is organized as follows: Sect. II provides background concepts of information visualization on which our approach is based. The whole approach is presented in Sect. III. The logging framework and its integration in *eWorkbook* is presented in Sect. IV, whereas in Sect. V we discuss experimental results. Several final remarks and a brief discussion on future work conclude the paper.

## II. Information Visualization for Knowledge Discovery

In the last decade the database community has devoted particular efforts to the extraction of knowledge from data. One of the main approaches for knowledge extraction is Data Mining, which applies automatic algorithms to recognize patterns in huge data collections [6], [7], [13]. Alternatively, visual data mining presents data in a visual form to stimulate user interaction in the pattern detection process. A combination of visual and automatic data mining draws together human cognitive skills and computer efficiency, which permits faster and more efficient knowledge discovery.

Since in this paper we are interested in a knowledge extraction process in which the tutor plays a central role, in what follows we will briefly review the basic concepts underlying data visualization and visual data mining.

### A. Data visualization

As opposed to textual or verbal communication of information, data visualization provides a graphical representation of data, documents, and structures, which turns out to be useful for various purposes. Data visualization provides an overview of complex and large datasets, shows a summary of data, and helps humans in the identification of possible patterns and structures in the data. Thus, the goal of data visualization is to simplify the representation of a given dataset, minimizing the loss of information [13], [14].

Visualization methods can be either geometric or symbolic. In a geometric visualization data are represented by using lines, surfaces, or volumes, and are usually obtained from a physical model, or as a result of a simulation or a generic computation. Symbolic visualization represents non numeric data using pixels, icons, arrays, or graphs.

A general classification of visualization methods is presented in [15], [16]. It constructs a three-dimensional visualization space by classifying the visualization methods according to three orthogonal criteria: the data type, the type of the visualization technique and the interaction methods, as shown in Fig. 1.

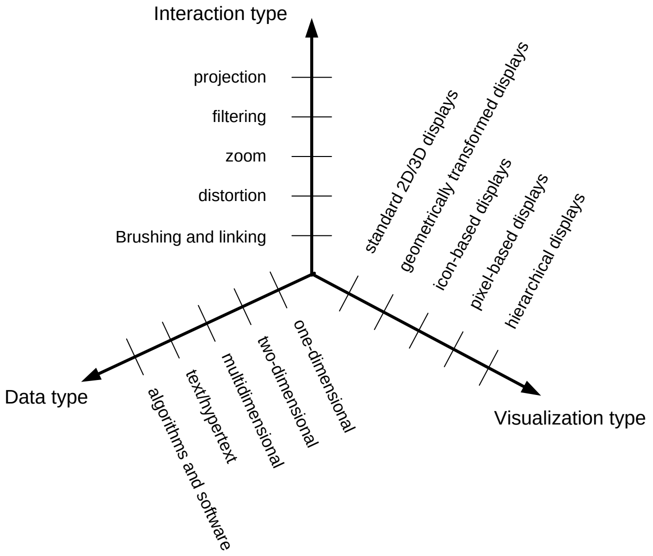


Fig. 1. Tree-Dimensional Visualization Space.

Examples of 2D/3D displays are line graphs and isosurfaces, histograms, kernel plots, box-and-whiskers plots, scatterplots, contour plots, and pie charts [6], [13], [17]. The scatterplot matrix, the permutation matrix, and its closely related survey plot are all examples of geometrically transformed visualization methods [18]. One of the most commonly used icon-based visualization method is star icons [13], [17]. However, another well known method falling into this category is Chernoff faces [18], [19]. An example of pixel based display method is recursive pattern visualization, which is based on a recursive arrangement of pixels [20]. Finally, examples of hierarchical visualization methods include dendrograms, structure-based brushes, Magic Eye View, treemaps, sunburst, and H-BLOB [21].

### B. Visual data mining

The process of visual data mining can be seen as a hypothesis generating process: the user generates a hypothesis about relationships and patterns in the data [13], [20].

Visual data mining has several advantages over the automatic data mining methods. It leads to a faster result with a higher degree of human confidence in the findings, because it is intuitive and it requires less understanding of complex mathematical and computational background than automatic data mining. It is effective when little is known about the data and the exploration goals are vague, since these can be adjusted during the exploration process. Visual mining can provide a qualitative overview of the data and allow unexpectedly detected phenomena to be pointed out and explored using further quantitative analysis [22].

The visual data mining process starts by forming the criteria about the visualizations to choose and the attributes to display. These criteria are formulated according to the exploration task. The user recognizes patterns in open visualizations and selects a subset of items s/he is interested in. The result of this selection is a restriction of the search space, which may show new patterns to the user, some of which s/he might not have been aware of before. The whole process can then be repeated on the selected subset of data items. Alternatively, new visualizations can be added. The process continues until the user is satisfied with the result, which represents a solution to her/his initial problem. The user has full control over the exploration by interacting with the visualizations [13].

Visual data mining has been used in a number of scientific disciplines. Some recent examples include detecting telephone call frauds by a combination of directed graph drawings and barplots [23], a classifier based on a parallel coordinates plot [24], and a visual mining approach by applying 3D parallel histograms to temporal medical data [25].

### C. Combining automatic and visual data mining

The efficient extraction of hidden information requires skilled application of complex algorithms and visualization tools, which must be applied in an intelligent and thoughtful manner based on intermediate results and background knowledge. The whole knowledge discovery process is therefore difficult to automate, as it requires high-level intelligence. By merging automatic and visual mining the flexibility, creativity, and knowledge of a person are combined with the storage capacity and computational power of the computer. A combination of both automatic and visual mining in one system permits a faster and more effective knowledge discovery process [13], [20], [26], [27], [27]–[32].

## III. The Approach

In this section we describe the approach to discover knowledge related to learner activities during on line tests, which can be used by tutors to produce new test strategies. In particular, we have devised a new symbolic data visualization strategy, which is used within a knowledge discovery process to graphically highlight behavioral patterns and other previously unknown aspects related to learners' activity in on line tests.

As we know from the literature, Knowledge Discovery (KDD) refers to the process of discovering useful knowledge from data [33]. In this context, data mining refers to a particular step of this process. Fig. 2 shows all the steps of a typical KDD process based on data mining [33]:

- data preparation;
- data selection;
- data cleaning;
- interpretation of results.

These steps are essential to ensure that useful knowledge is extracted from data. In what follows we describe the knowledge discovery process used in our approach.

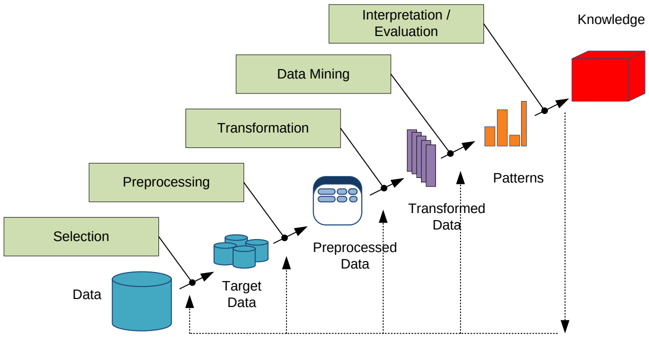


Fig. 2. The Steps of a KDD process.

### A. Data Collection

The proposed approach aims to gather data concerning the *learner's* activities during on line tests. As said above, a well known strategy to collect data concerning learner's activities is the *think out loud* method [1]–[5]. However, we have already mentioned the numerous drawbacks of this method. In fact, it is quite invasive, which can significantly influence user behavior. Further, it is difficult to use in practice, since it requires a considerable effort in terms of staff personnel to analyze tape recorded data.

One way of collecting data with a less invasive method is to track eyes' movements. However, this would require the use of expensive tools, which becomes even worse if large scale simultaneous experiments need to be conducted, like it happens in the e-learning field. Nevertheless, it has been shown that similar results can also be inferred by tracking mouse movements. In particular, an important experiment has revealed that there is a significant correlation between eyes' and mouse's movements [34]. This means that tracking the trajectory drawn by the mouse pointer could be useful for deriving the probable trajectory of user's eyes, which in our case might reveal the strategy the learner is using to answer multiple choice questions.

In the light of the above arguments, in our approach we have aimed at gathering data concerning mouse movements. In particular, the data schema is organized in learner test sessions. Each session is a sequence of page visits (each page containing a question). The following information are relevant for a page visit:

- Duration of the visit;
- Presence and duration of inactivity time intervals (no interactions) during the visit;
- Sequence of responses given by the learner during the visit;
- Estimation of the time spent by the learner in evaluating the stem (the text that states the question) and each of the options for the question.

These data are organized according to the schema depicted in Fig. 3. In particular, for each test session learner's activity is represented by a set of *ItemView* elements, which in turn are associated to a set of *Response* and *Observation* objects. Each *ItemView* represents the visit of the learner at the web page containing a given question. Usually, during a test session there is at least one *ItemView* per item, but in some cases the learner can skip or revise one or more questions. Each *ItemView* embodies a set of *Observation* elements, each of them representing the activity of item evaluation by the learner. With the term evaluation we mean the mouse activities of the learner on the web page containing the stem and the set of options. For each observation we store the type of user interactions (i.e. the kind of mouse movements) and the duration of that observation. Finally, in each *ItemView* the learner can give one or more responses, and for each of them we store its correctness and timestamp.

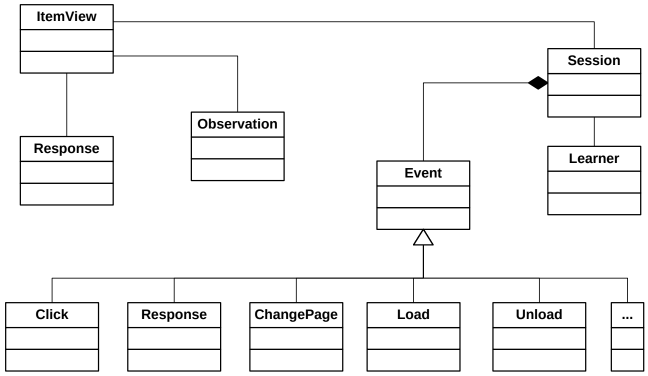


Fig. 3. Conceptual Schema of the Data collected during a Test Session.

The events captured during learner's interaction are the following:

- Events occurred in the browser window (open, close, resize, load, unload);
- Events occurred in the browser client area (key pressing, scrolling, mouse movements and clicks).

### *B. Data Visualization*

Once collected the above mentioned data, classical data mining algorithms can be used to detect behavioral patterns. However, in this paper we aim to enhance this evaluation activity by means of data visualization strategies to combine automatically inferred knowledge with that detected by the tutor through observations on graphically represented data. To this end, data collected during test execution are processed in order to derive a visual representation of the learner's activities. In particular, for each learner we produce a two-dimensional chart showing a chronological review of the test. An example of this chart can be seen in Fig. 4. The two axis show the time line (horizontal axis) and the questions submitted to the learner (vertical axis), respectively. The entire test execution is represented through a broken line, which is composed by a certain number of segments:

- An *Horizontal* Segment shows the *view* of an item for a given time interval. Consequently, the length of the segment is proportional to the duration of the visualization of the corresponding item;
- A *Vertical* Segment represents a browsing event. A segment oriented towards the bottom of the chart represents a backward event, that is, the learner has pressed the button to *View* a previous item. Analogously, a segment oriented towards the top represents a forward event;
- *Circles* represent the responses given by the learner on an item. The progressive number of the chosen option is printed inside the circle. The indication of correct/incorrect response is given by the circle's filling color: a blue circle represents a correct response, a red circle represents an incorrect one. Colors are used also for representing the position of the mouse pointer during the item *View*;
- The *Mouse* pointer presence in the stem area is represented by coloring the horizontal line in black. If the learner moves the mouse's pointer in one of the options areas, the line changes to a color that is specific of the selected option. Different colors have been used for 1 to 5 numbered options, respectively. The gray color is used to report the presence of the mouse pointer in a neutral zone.

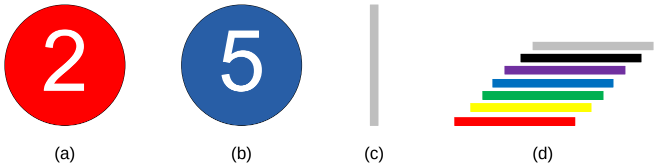


Fig. 5. Basic Elements of the chart: a) wrong response b) right response c) browsing event d) item view.

Basic graphics elements composing the chart are shown in Fig. 5. The graphical chronological review of a sample test is shown in Fig.4. The test is composed of 25 items and the maximum expected duration is 20 minutes, even if the learner has completed and submitted the test in 17 minutes, approximately. The whole *View* of the chart in the figure shows the strategy adopted by the learner in the test: the test execution is visibly composed of two successive *phases*. In the first one (nearly 9 minutes), the learner has completed the *View* of all the items from 1 to 25. Responses to 19 questions have been given in this *phase*. Several items present more than one response, maybe due to learner's thinking, while a few items have been skipped. The mouse position for each item *View* is visible through a zoom, performed by using a suitable zoom tool of the analyzer.

## IV. The System

In this section, we describe the system implementing the proposed approach. The system is composed of a *Logging Framework* and a *Log Analyzer* application. The former, based on the *AJAX* technology [11], captures and logs all of the learners interactions with the *e-testing* system interface (running in the Web browser). It can be instantiated in any *e-testing* system and is further composed of a client-side and a server-side module. The latter is a stand-alone application that analyzes the logs in order to extract information from them and to graphically represent it.

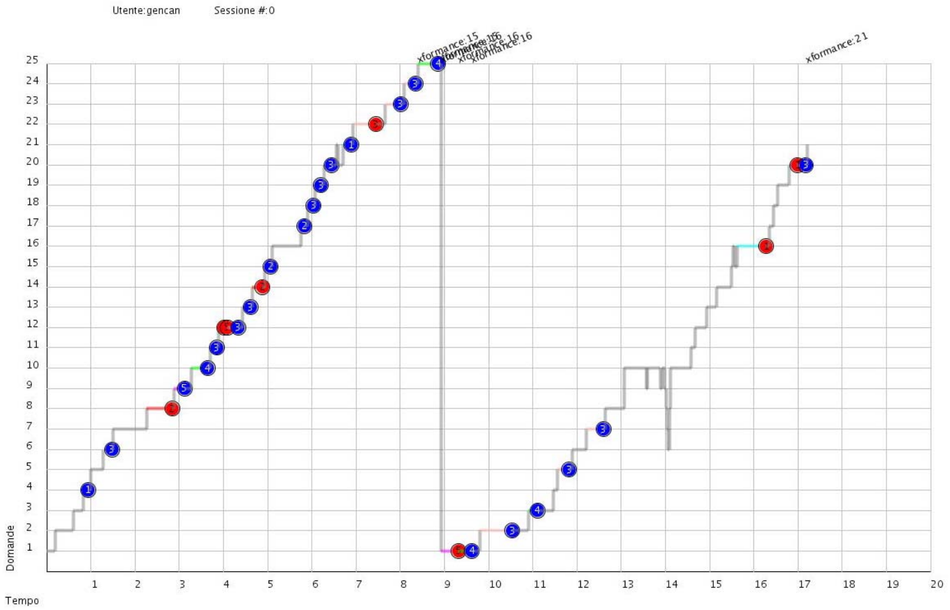


Fig. 4. Graphical Chronological Review of a Sample Test.

### A. The Logging Framework

The purpose of the *Logging Framework* is to gather all of the learner actions while browsing Web pages of the test and to store raw information in a set of log files in *XML* format.

The framework is composed of a client-side and a server-side module. The former module is responsible for "being aware" of the behavior of the learner while s/he is browsing the test's pages and for sending information related to the captured events to the server-side module. The latter receives the data from the client and creates and stores log files to the disk.

Despite the required interactivity level, due to the availability of *AJAX*, it has been possible to implement the client-side module of our framework without developing plug-ins or external modules for Web browsers. *JavaScript* has been used on the client to capture learner interactions and the communication between the client and the server has been implemented through *AJAX* method calls. The client-side scripts are added to the *e-testing* system pages with a light effort by the programmer.

The event data is gathered on the Web browser and sent to the server at regular intervals. It is worth noting that the presence of the javascript modules for capturing events does not prevent other scripts loaded in the page to run properly.

The server-side module has been implemented as a Java servlet, which receives the data from the client and keeps them in an *XML* document that is written to the disk when the learner submits the test. The *Logging Framework* can be instantiated in the *e-testing* system and then enabled through the configuration.

The information *Model* used for the log data is quite simple and is shown in Fig. 6. The information is organized per learner test session. At this level, user name (if available), IP address of the learner, session identifier, as well as agent information (browser type, version, and operating system) are logged. A *session* element contains a list of event elements. The data about user interactions are the following:

- Event type;
- *HTML* source object involved in the event (if present);
- Mouse information (pressed button, coordinates);
- Timing information (timestamp of the event);
- More information specific of the event, i.e. for a "response" event (a response given to a question), the question and option identifiers, with the indication whether the response was right or wrong are recorded.

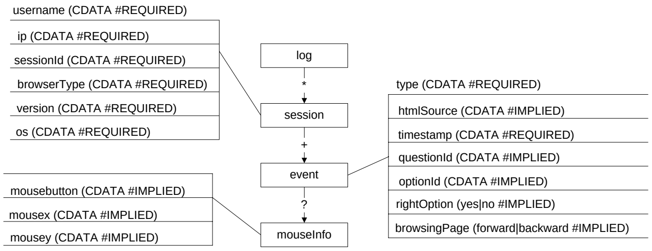

username (CDATA #REQUIRED)
ip (CDATA #REQUIRED)
sessionId (CDATA #REQUIRED)
browserType (CDATA #REQUIRED)
version (CDATA #REQUIRED)
os (CDATA #REQUIRED)
log
*
session
+
event
?
mouseInfo
mousebutton (CDATA #IMPLIED)
mousex (CDATA #IMPLIED)
mousey (CDATA #IMPLIED)
type (CDATA #REQUIRED)
htmlSource (CDATA #IMPLIED)
timestamp (CDATA #REQUIRED)
questionId (CDATA #IMPLIED)
optionId (CDATA #IMPLIED)
rightOption (yes|no #IMPLIED)
browsingPage (forward|backward #IMPLIED)

Fig. 6. The information *Model* for log data.

An important concern in logging is the log size. If an experiment involves a large set of learners and the test is composed of many questions, log files can reach big sizes. A configuration system, including the following configuration settings, has been conceived in order to reduce log sizes:

- List of events to capture;
- Sub-set of attributes for each event;
- Sections of the Web pages (*divs* or table cells) to be monitored as event sources;
- Time interval between two data transmissions from the client to the server;
- Sensitivity for mouse movements (short movements are not captured).

On the client-side, everything can be done in the Web browser. The *JavaScript* modules for event capturing are *dynamically* generated on the server according to the configuration settings, are downloaded and run in the browser interpreter. Data is sent to the server through an *AJAX* request. On the server-side, a module called *RequestHandler* receives the data and sends them to a module called *LoggerHandler*, which organizes the *XML* document in memory and flushes it to the disk every time a learner test session finishes. The architecture of the framework is graphically represented in Fig. 7.

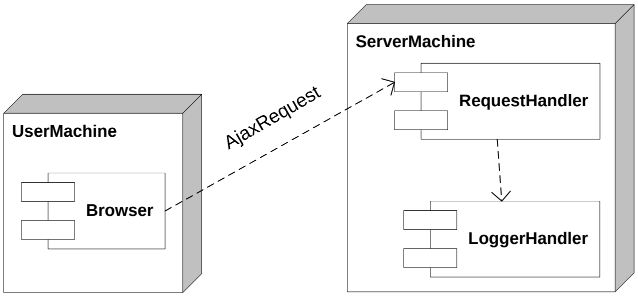


Fig. 7. The Logging Framework Architecture.

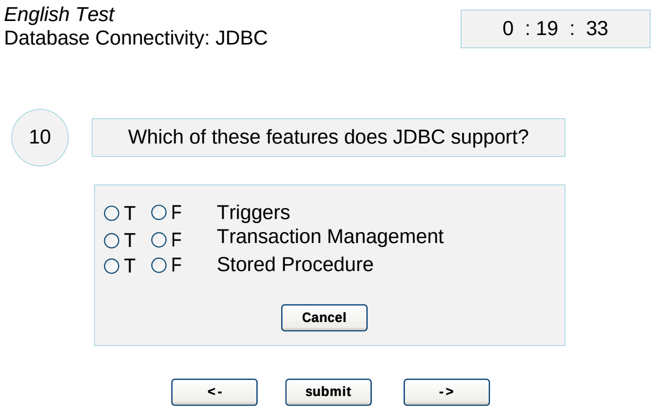
## B. Instantiation of the Logging Framework in the eWorkbook System

The above described framework has been instantiated in an existing Web-based *e-testing* system, named *eWorkbook* [12], which is used for evaluating learner's knowledge by creating (the tutor) and taking (the learner) on-line tests based on *multiple choice* question types. *eWorkbook* adopts the classical three-tier architecture of the most common J2EE Web-applications. The *Jakarta Struts* framework has been used to support the *Model* 2 design paradigm, a variation of the classic *Model View Controller* (MVC) approach. *Struts* provides its own *Controller* component and interacts with other technologies to provide the *Model* and the *View*. In our design choice, *Struts* works with *JSP*, for the *View*, while it interacts with *Hibernate* [35], a powerful framework for *object/relational persistence* and query service in Java, for the *Model*.

The application is fully accessible through a Web browser: no plug-in installations are needed, since it only delivers fully accessible HTML code and the included client-side scripts conform to the standard *ECMAScript* language [36]. It is worth noting that the system has been tested on recent versions of most common browsers (i.e., *Internet Explorer*, *Netscape Navigator*, *FireFox* and *Opera*).

The integration of the *Logging Framework* in the *eWorkbook* system has been rather simple. For the integration of the server-side component, the JAR (Java ARchive) file containing the framework classes has been imported as a library in the system. Furthermore, a modification to the system's deployment descriptor has been necessary. For the integration of the client-side component, it has been necessary to consider the structure of the *eWorkbook* system interface.

The test is launched in a child browser window of the system Web page. This window displays a timer to inform the learner of about the time left to complete the test and contains the controls to flow (*forward* and *backward* buttons) between the questions, and the button to submit the test. The *stem* and the form containing the *options* are loaded in an *iframe* window placed in the center of the page. A screen shot of the Web page showing the test interface is shown in Fig. 8. The *JavaScript* modules for capturing the events have been included in both the external window and the internal *iframe*, while the modules for communicating with the server have only been included in the main page. In order to identify the source element of the events, the HTML code of the *eWorkbook* interface has been slightly modified by adding *HTML* element identifiers to several elements (form interactive elements, such as labels and radiobuttons).



*English Test*
Database Connectivity: JDBC

0 : 19 : 33

10

Which of these features does JDBC support?

○T ○F Triggers
○T ○F Transaction Management
○T ○F Stored Procedure

Fig. 8. Screenshot of the Test Execution.

## C. The Log Analyzer Application

The data analysis and test visualization operations are performed by a Web-based stand-alone application, optionally hosted on the same server of the *Logging Framework*, which takes in input the *XML* log files. The *Log Analyzer* Application is composed of two modules: the *Query Engine* and the *Chart Generator*. The *Query Engine* module performs queries on the log files in order to obtain the desired data as an instance of the data model shown in Fig. 6. Since the log files are in *XML* format, the queries necessary to perform the above operation have been expressed in the *XQuery* language [37], and have been performed on the log files by using an implementation of the *JSR 225 - XQuery API for Java* [38]. Once obtained the data in the format shown in Fig. 6, these are given in input to the *Chart Generator* module. This module has been implemented through a Java Servlet, which takes the identifier of the learner whose test is going to be analyzed as a parameter. The module dynamically generates the chart and returns it in the PNG format. To produce the charts, the *Standard Widget Toolkit* (SWT) API has been used. The choice of dynamically constructing charts reduces the space necessary for the storage of the images: for caching purposes, only images obtained within the browser session are saved on the server and are deleted at the end of the session. With this approach, the computation necessary to construct each visualized chart is executed once per test analysis session, even when the same chart is visualized more times across the same session.

## V. Experimental Results

In order to demonstrate the effectiveness of the approach and of the proposed system we have used them in the context of the *Web Development Technologies* course in our faculty: the *eWorkbook* system [12], equipped with the new module for tracking learners' interactions, has been used to administer an on-line test to learners. These have not been informed of the experiment; they just knew that the grade obtained on the tests concurred to determine the final grade of the course exam. The test, containing a set of 25 items to be completed in a maximum of 20 minutes, was administered to 71 learners, who took it concurrently in the same laboratory. The assessment strategy did not prescribe penalties for incorrect responses, and the learners were aware of that. The logger was enabled and an approximately 4Mb sized *XML* log file was produced. The logging activity produced no visible system performance degrading.

Then, the *Log Analyzer* has been used for analyzing the logs in order to extract information from them and to graphically represent it in order to trigger a visual data mining process where the tutor plays a central role. In the case of the mentioned experiments, the visual analysis of charts enabled a tutor to infer interesting conclusions about both the strategies the learners used to complete tests, and the correlation between questions. In the former case, the objective was not only to understand the learners' strategies, but also to detect the most successful of them under the circumstances explained above, and possibly, to provide learners with advice on how to perform better on the next time. On the other hand, question correlation analysis aims to improve the final quality of the test by avoiding the composition of tests with related questions.

### A. Strategies for Executing Tests

Some of the experiments found in literature, aiming to analyze learner's behavior during tests, have regarded the monitoring of learner's habit to verify the given answers and to change them with other, more plausible, options [1], [2]. These experiments try to answer the following two questions:

- Is it better for learners to trust in their first impression or to go back for evaluating the items again and, eventually, for changing the given answers?
- Are the strongest or the weakest learners those who change idea more frequently?

In the experiments, *right-to-wrong* and *wrong-to-right* changes have been recorded, and their numbers have been correlated to learners' final mark on the test. Similar experiments have been carried out by correlating the time to complete the test with the final mark [3], [4]. An interesting experiment focuses on the strategies used by several learners with different abilities ("A", "C" and "F" students) to complete the tests [5]. In particular, the frequency of several habits has been recorded and then correlated to learners' final mark on the test, such as: the habit of giving a rash answer to the question as soon as a plausible option is detected, often neglecting to consider further options; the habit of initially skipping questions whose answer is more uncertain, in order to evaluate them subsequently; and so on. The above experiments only consider some aspects of test execution. We have found no study in the literature that considers the learner behavior as a whole. By using our approach, we have replicated several experiments performed for traditional testing in an on-line fashion. In particular, we have analyzed the effects of response changing and question skipping. More importantly, we have tried to put these aspects together, obtaining a more general analysis of the strategies used by the learners to complete the tests. As for response changing, we have obtained the average number of response changes, including *wrong-to-right* and *right-to-wrong* ones, and their correlation with the final score. The results obtained in our experiments are summarized in table I. The table reports the following measures:

TABLE I
Extract of the data matrix and results of the response changing analysis.

| UserName | R2W | W2R | NOC | Score |
|---|---|---|---|---|
| achpape | 0 | 1 | 2 | 13 |
| adomig | 0 | 0 | 0 | 13 |
| alepet | 0 | 1 | 1 | 14 |
| andast | 0 | 1 | 1 | 20 |
| ... | ... | ... | ... | ... |
| MEAN RW2 | 0.91 | | | |
| MEAN W2R | 1.36 | | | |
| MEAN NOC | 3.46 | | | |
| CORR(NOC,SCORE) | 0.11 | | | |

- *UserName*: The name used by the e-testing system to uniquely identify a learner;
- *R2W*: Number or *right-to-wrong* changes made by the learner during a test session;
- *W2R*: Number of *wrong-to-right* changes made by the learner during a test session;
- *NOC*: The total number of changes made by learner during a test session. This number also includes wrong-to-wrong changes not considered in previous measures;
- *Score*: The number of questions correctly answered by the learner during the test session (out of 25).

Even in our experiment, *wrong-to-right* changes are slightly more frequent than *right-to-wrong* ones, as shown by the average values for these measures.

As for question skipping, we have analyzed its correlation with the final score: we have divided the learners participating in the experiment in three groups according to their score. The first group contains 27 learners with a *low score*; the second group 27 learners with a *medium score*; the third group 26 learners with a *high score*. The results obtained in our experiments are summarized in table II. The table reports the following measures:

- *Score*: The group to which the score belongs to;

TABLE II
RESULTS OF THE QUESTION SKIPPING ANALYSIS.

| Score | Average | SD |
|---|---|---|
| low | 5.11 | 4.96 |
| medium | 5.26 | 6.38 |
| high | 3.22 | 4.65 |
| 0 skipped Average Score = 15.67 | | |
| more than 1 skipped Average Score = 16.58 | | |

- *Mean*: The average number of questions initially skipped by the learner during a test session;
- *SD*: The Standard Deviation of questions initially skipped by the learner during a test session.

The *high score* group has the lowest average number of skipped items. This suggests that strong learners are less likely to skip questions than weak ones.

As for the strategies, from the analysis of the charts obtained through our experiment, the following strategies have been discovered:

- *Single Phase*. This strategy is composed of just one *phase*. By the term *phase*, we mean the time necessary to browse, and eventually answer, all the items in the test. The time available to complete the test is organized by the learner in order to browse all the questions just once. The learner tries to reason upon a question for an adequate time and then gives a response in almost all cases, since s/he knows that there will not be a revision for the questions. Possible *phases* subsequent to the first one have a negligible duration and no responses. A sample of this strategy is shown in Fig. 9(a);
- *Active Revising*. This strategy is composed of two or more *phases*. The learner intentionally browses all the questions in a time shorter than the one available, in order to leave some time for revising *phases*. The questions whose answer is uncertain are skipped, and the response is left to subsequent *phases*. A general rule that has been discovered is that the first *phase* lasts longer, whereas subsequent *phases* have decreasing durations. A sample of this strategy is shown in Fig. 9(b);
- *Passive Revising*. This strategy is composed of two or more *phases*. During the first phase, the learner browses and answers all the questions as fast as possible (the main difference with *Active Revising* lies in the lack of question skipping during this phase). The remaining time is used for one or more revising *phases*. While in the *Active Revising*, the learner has the "active" role of answering the previously skipped questions, in *Passive Revising* s/he only checks the previously answered questions. Also in this case we discovered the same general rule regarding subsequent *phases*: the first phase lasts longer and subsequent phases have decreasing durations. A sample of this strategy is shown in Fig. 9(c).

By visually analyzing the data of our experiment it came out that the most frequently adopted strategy is *Active Revising*, which was used by 40 learners over 71 (56.5%), followed by the *Passive Revising* strategy (20 learners over 71, 28.2%), and by the *Single Phase* one, used only in 9 cases over 71 (12.7%). Only two learners have adopted an atypical strategy (see Fig. 9(d)), which cannot be classified in any of the previously described patterns. The best results have been achieved by learners who adopted the *Passive Revising* strategy, with an average score of 17.6 exact responses over the 25 test questions. With the *Active Revising*, instead, an average score of 16.4 has been achieved. Lastly, the *Single Phase* strategy turned out to be the worse one, showing an average score of 15.1. Therefore, it appears that a winning strategy is one using more than one *phase*, and this is confirmed by the slightly positive linear correlation (0.14) observed between the number of *phases* and the score achieved on the test. For both strategies using more than one *phase* (active and passive revising) the score is often improved through the execution of a new *phase*. The improvement is evident in the early *phases* and tends to be negligible in subsequent *phases*: starting from a value of 14.3 obtained after the first *phase*, the average score increases to 16.2 after the second *phase*. The presence of further *phases* brings the final score to an average value of 16.5. The average duration of the first *phase* in the *Passive Revising* strategy (14'50") is longer than the one registered for the *Active Revising* strategy (11'51"). This result was predictable, since, by definition, the *Active Revising* strategy prescribes the skipping (= less reasoning) of the questions whose answer is more uncertain for the learners. Another predictable result, due to the above arguments, is that the *Passive Revising* strategy has less *phases* than the *Active* one, on average (2.55 and 3.2, respectively).

### B. Detection of Correlation Among Questions

By visually inspecting the charts, we can also be assisted in the detection of correlated questions. In some cases, a visual pattern similar to a stalactite occurs in the chart, as shown in Fig. 11. The occurrence of such a pattern says that while the learner is browsing the current question s/he could infer the right answer by a previous question. In the example shown in the figure, while browsing question "2", the learner understood

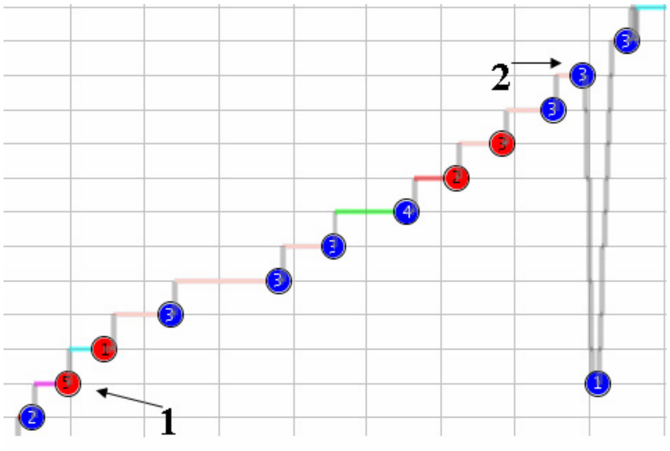
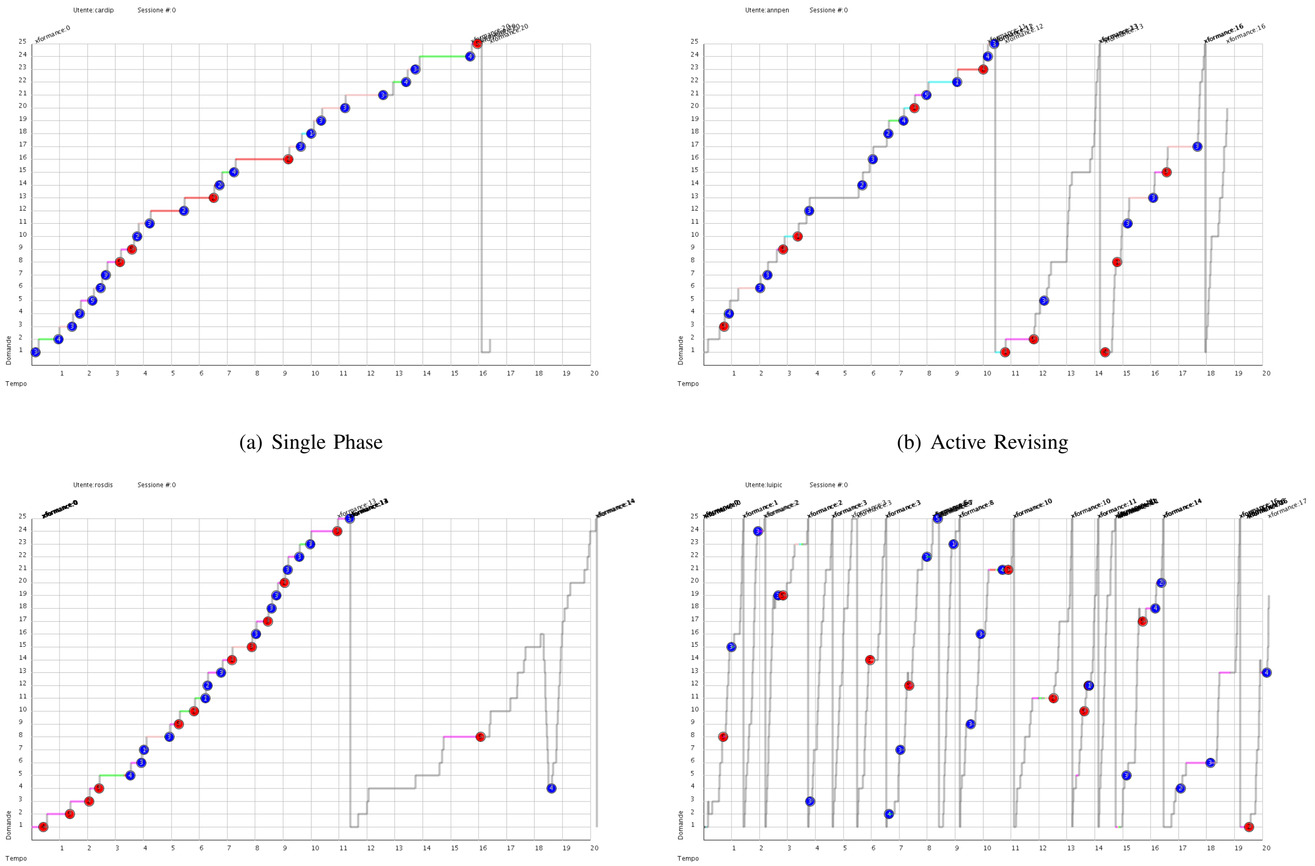


Fig. 11. A *Stalactite* in the Chart Shows the Correlation Between Two Questions.



(a) Single Phase

(b) Active Revising

(c) Passive Revising

(d) Atypical

Fig. 9. Sample of Test Execution Strategies.

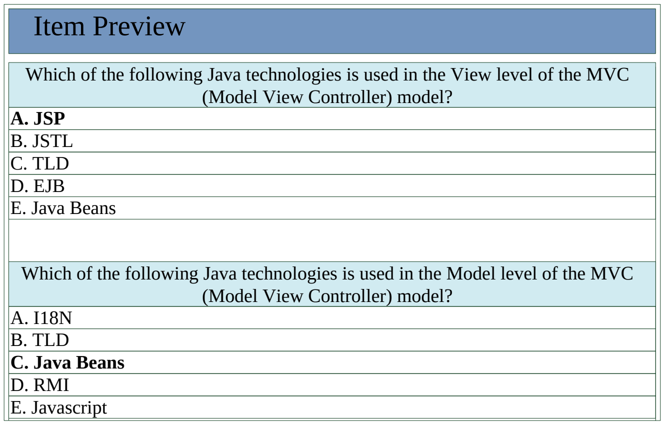
that s/he had given a wrong response to question “1”, and went back to change the response, from option 5 to option 1.

To give the reader a more precise idea of what happened, the two correlated items have been reported in Fig. 12. For question “1”, the learner initially gave an erroneous response, selecting the option number 5 (“JavaBeans”) as correct. About 7 minutes later (see figure 11), the learner has browsed question “2” and, by excluding the less plausible distractors (“I18N”, “TLD”, “RMI” and “Javascript”), succeeded in providing the correct response (option C: “Javascript”). At the same moment, s/he remembered s/he had given the same response to question “1”.

In a short time s/he sequentially browsed all of the items until reaching question “1” and revised it by providing the correct response (option A: “JSP”). The stalactite in the chart indicates the rapid sequence of “backward” events followed by a response and a rapid sequence of “forward” events.

The learner, yet having a bad preparation about the subject, achieved a full score on the two questions concerning the “MVC Model”, due to the presence of two strongly correlated questions.

Starting from the observation of the charts, we detected the correlated questions and derived that the test could have had a better quality if it included just one of the two items shown in Fig. 12.



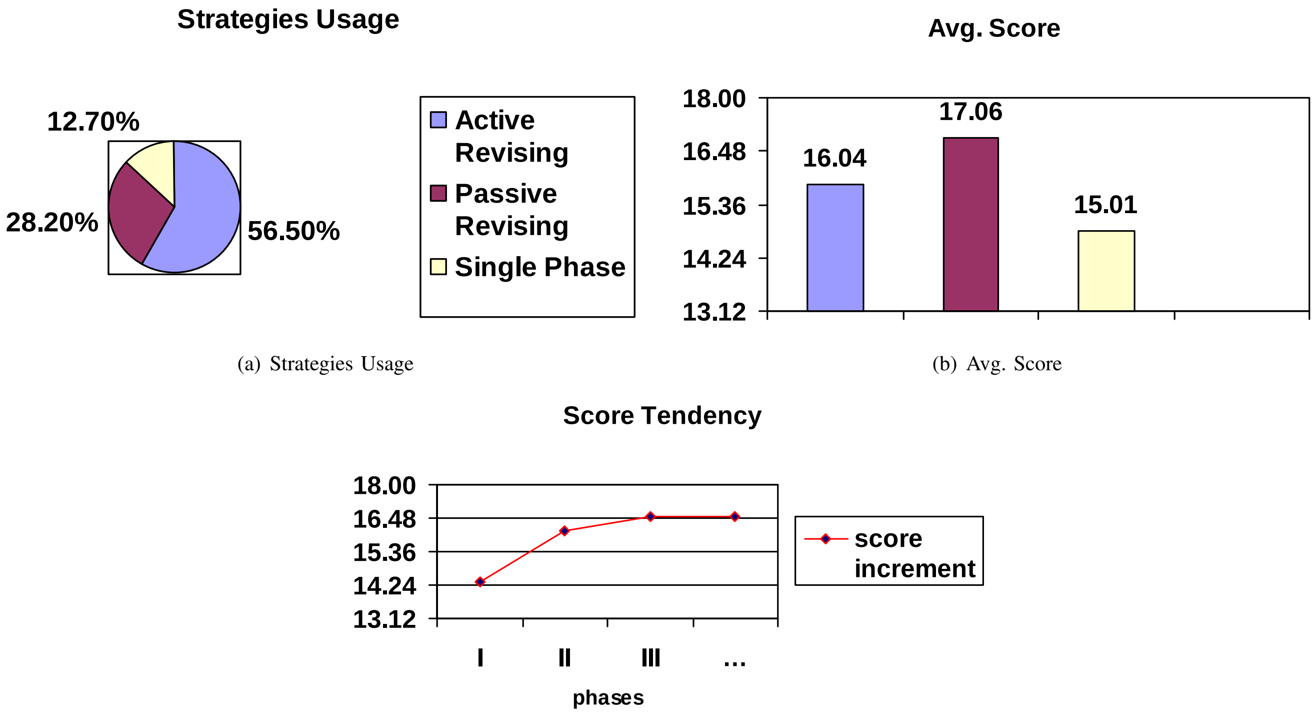
Item Preview

| Which of the following Java technologies is used in the View level of the MVC (Model View Controller) model? |
|---|
| **A. JSP** |
| B. JSTL |
| C. TLD |
| D. EJB |
| E. Java Beans |
| |
| Which of the following Java technologies is used in the Model level of the MVC (Model View Controller) model? |
| A. I18N |
| B. TLD |
| **C. Java Beans** |
| D. RMI |
| E. Javascript |

Fig. 12. Two strongly correlated question items.

## *C. Detection of Cheating*

As proven by several studies in the education field, many learners cheat during exams, when they can [39], [40]. Cheating detection in assessment tests is not an easy task: most of the techniques employed so far have been based on the comparison of the results obtained in the tests [41]. These techniques cannot give the certainty of the guilt, since a high



(a) Strategies Usage

(b) Avg. Score

(c) Score Tendency

Fig. 10. Strategies Comparison

similarity of two tests can be due to coincidence.

Furthermore, as in all fraud detection systems, the task is complicated by several technological and methodological problems [42]. It could be useful to gain information on the learners' behavior during the test. Analysis on these data can be integrated to results comparison in order to have a more comprehensive data set as input for a data mining process aiming at detecting cheating. For example, let us consider the following situation: during the test, learner A answers *true* to a question and learner B, who is seated behind the former, answers the same few instants later. The tracking of this information, available through the charts of our system, could be useful to prove that the learner has cheated, looking on the screen of his classmate.

Copying from the colleague set in front of someone is not the only frequently encountered cheating exploit: in some cases several attempts of *gaming the system* have been reported [10]. This exploit consists of executing a large number of tests with the only scope of having as many correct responses as possible from the *e-testing* feedback system. When some suspicious cases are detected from the frequency of self-assessment test access and the strange scores achieved, our system can confirm the cheating attempts, by revealing atypical patterns of test execution.

## VI. Related Work on the Use of Data Visualization in E-Learning

The data gathered in years of use of e-learning systems contain knowledge that can provide precious information for improving the approach to on-line educational methodologies. To this end, several methods and tools for Data Visualization have been frequently applied to e-learning in recent years, since they can rapidly convey information to the user. The same information is often incomprehensible, if provided in a tabular or in other formats [9].

Many aspects related to on-line learning can be rendered in a suitable graphical format. Several works found in literature are aimed at understanding learner's behavior through the analysis of the data tracked during learners' activities, such as, communication in discussion forums and other tools [9], [43], [44], accesses to the e-learning system pages, with particular emphasis on the accesses to on-line lecture material [9], [45]. Some of the above papers have also focused on the data analysis of assessment tools, but only in relation to the participation of the learner to the on-line activities and to the results achieved in on-line tests. Although it has roused a remarkable interest in the classical forms of learning, the analysis of learner's behavior during tests has no precedents in e-learning literature. More aspects of e-learning treated using data visualization techniques include: the structure of learning material [46] and the use of 3D animation graphics in educational activities [47]. Other, more ambitious projects, leading to more debatable results, focus on the visualization of the relations among the e-learning system's data. In one case [48], the relations are shown in order to better understand the structure of the system interface. In another approach the visualization is generalized to all the aspects related to learning, letting the user choose the data of interest [49]. Another important work focused on the use of Data Visualization in E-learning is the one underlying the CourseVis system [9]. This system analyzes and visualizes data regarding social, cognitive, and behavioral factors of the learning process. We have considered the evaluation of multiple aspects as an extremely interesting feature, and the example of this work has induced us to analyze both the aspects of learner's behavior and test quality in our work. The study proposed by Mochizuki et al. is also noteworthy [44]. In this work, the learners (primary school young learners) visualize a chart in order to understand their level of involvement in the main arguments treated in the discussion forums and modify their behavior accordingly. For the above reasons, the authors have made the interpretation of the visual metaphor as simple as possible. The interactive visualization of information regarding the activity in the discussion forums is also a concern of the work presented by May et al. [43]. Particular attention in the paper has been devoted to the description of technological aspects, such as: the integration of the data tracked on the client-side with those recorded on the server-side, and the possibility of instantiating the data logging framework on many e-learning systems, as we have done in our work too. The technique of tracking client-side events has also been employed by Bates et al. [45], in reference to the paths followed by the learners in the browsing of learning material. Typical data gathered in this activity are the frequency and the duration of the accesses to on-line material. Another activity performed by the authors is the analysis of the correlation between the behavioral data and the results achieved in the exams. The same thing has been done in our work. Nevertheless, while the learner's behavior during test execution seems to be correlated to the final score of the tests, the authors have not found any interesting result about the correlation of the data on the browsing of courseware material with the score achieved on the final exam of the course.

## VII. Conclusion

We have presented an approach and a system to let tutors monitor learners' strategies during on-line tests. The approach exploits data visualization to draw the data characterizing the learner's test strategy, in order to trigger tutor's attention and to let him/her discover previously unknown behavioral patterns of learners, and conceptual relationships among test items. In this way the tutor is provided with a powerful tool that let him/her review the whole assessment process, and evaluate possible improvements.

We have extensively used the implemented system experimentally to evaluate on line test's strategies in courses of our faculty, in order to assess the whole approach. This let us discover several relevant patterns regarding test quality, characteristics of used strategies, and the impact on the final score.

In the future we would like to devise new visual representations and perform further experiments, possibly in combination with classical data mining algorithms. Moreover, since the approach lends itself to the application to other application fields, such as e-commerce, in the future we would like to evaluate its use in these contexts.

Finally we wish to explore more deeply the problem of cheating detection, that here we just treat in broad terms, since we believe that an approach based on logging and visualization can be promising and effective to this aim.

## References

[1] J. Bath, "Answer-changing behaviour on objective examinations," *The Journal of Educational Research*, no. 61, pp. 105–107, 1967.

[2] J. B. Best, "Item difficulty and answer changing," *Teaching of Psychology*, vol. 6, no. 4, pp. 228–240, 1979.

[3] J. Johnston, "Exam taking speed and grades." *Teaching of Psychology*, no. 4, pp. 148–149, 1977.

[4] C. A. Paul and J. S. Rosenkoetter, "The relationship between the time taken to complete an examination and the test score received." *Teaching of Psychology*, no. 7, pp. 108–109, 1980.

[5] L. McClain, "Behavior during examinations: A comparison of "a", "c," and "f" students." *Teaching of Psychology*, vol. 10, no. 2, pp. 69–71, 1983.

[6] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. The MIT Press, Cambridge, Massachusetts, 2001, adaptive Computation and Machine Learning Series, A Bradford Book.

[7] N. Ye, *Introduction In: The Handbook of Data Mining*. Lawrence Erlbaum Associates Publishers, 2003.

[8] C. Plaisant and B. Shneiderman, "Show me! guidelines for producing recorded demonstrations," *vlhcc*, vol. 00, pp. 171–178, 2005.

[9] R. Mazza and V. Dimitrova, "Student tracking and personalization: Visualising student tracking data to support instructors in web-based distance education," in *Proceedings of the 13th International World Wide Web conference on Alternate Track papers & posters*, 2004, pp. 154–161.

[10] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner, "Off-task behavior in the cognitive tutor classroom: when students "game the system"," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press, 2004, pp. 383–390.

[11] S. D. N. (SDN), "Asynchronous javascript technology and xml (ajax) with the java platform," 2007. [Online]. Available: http://java.sun.com/developer/technicalArticles/J2EE/AJAX/

[12] G. Costagliola, F. Ferrucci, V. Fuccella, and F. Gioviale, "A web based tool for assessment and self-assessment," in *Proceedings of ITRE 04*, 2004, pp. 131–135.

[13] U. Demšar, "Data mining of geospatial data: combining visual and automatic methods," Ph.D. dissertation, Department of Urban Planning and Environment, School of Architecture and the Built Environment, Royal Institute of Technology (KTH), 2006.

[14] U. Fayyad and G. Grinstein, *Information Visualisation in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers, 2002, ch. Introduction.

[15] D. A. Keim and M. Ward, *Visualization. In: Berthold M and Hand DJ (eds) Intelligent Data Analysis*, 2nd ed. Springer Verlag, 2003.

[16] D. A. Keim, *Visual Exploration of Large Data Sets*, 2nd ed. Springer Verlag, 2003.

[17] G. Grinstein and M. Ward, *Information Visualisation in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers, 2002, ch. Introduction to Data Visualization.

[18] P. E. Hoffman and G. G. Grinstein, "A survey of visualizations for high-dimensional data mining," *Information visualization in data mining and knowledge discovery*, pp. 47–82, 2002.

[19] M. Ankerst, "Visual data mining," Ph.D. dissertation, Ludwig Maximilians Universitat, Munchen, Germany, 2000.

[20] D. A. Keim, W. Müller, and H. Schumann, "Visual data mining," in *STAR Proceedings of Eurographics 2002*, D. Fellner and R. Scopigno, Eds. Saarbrücken, Germany: Eurographics Association, sep 2002, eurographics 02 STAR.

[21] D. A. Keim, "Information visualization and visual data mining," *IEEE Trans. Vis. Comput. Graph.*, vol. 8, no. 1, pp. 1–8, 2002.

[22] D. A. Keim, C. Panse, M. Sips, and S. C. North, "Pixel based visual data mining of geo-spatial data," *Computers & Graphics*, vol. 28, no. 3, pp. 327–344, 2004.

[23] K. Cox, S. Eick, and G. Wills, “Brief application description - visual data mining: Recognizing telephone calling fraud.” *Data Mining and Knowledge Discovery*, vol. 1, pp. 225–231, 1997.

[24] A. Inselberg, “Visualization and data mining of high-dimensional data.” *Chemometrics and Intelligent Laboratory Systems*, vol. 60, pp. 147–159, 2002.

[25] L. Chittaro, C. Combi, and G. Trapasso, “Data mining on temporal data: a visual approach and its clinical application to hemodialysis.” *Journal of Visual Languages and Computing*, vol. 14, pp. 591–620, 2003.

[26] H. Miller and J. Han, *Geographic Data Mining and Knowledge Discovery*. London and New York.: Taylor and Francis, 2001, ch. An overview., pp. 3–32.

[27] I. Kopanakis and B. Theodoulidis, “Visual data mining modeling techniques for the visualization of mining outcomes.” *Journal of Visual Languages and Computing.*, no. 14, pp. 543–589, 2003.

[28] M. Kreuseler and H. Schumann, “A flexible approach for visual data mining.” *Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 39–51, 2002.

[29] G. Manco, C. Pizzuti, and D. Talia, “Eureka!: an interactive and visual knowledge discovery tool.” *Journal of Visual Languages and Computing*, vol. 15, pp. 1–35, 2004.

[30] S. Kimani, S. Lodi, T. Catarci, G. Santucci, and C. Sartori, “Vidamine: a visual data mining environment.” *Journal of Visual Languages and Computing*, vol. 15, pp. 37–67, 2004.

[31] I. Kopanakis, N. Pelekis, H. Karanikas, and T. Mavroudkis, *Visual Techniques for the Interpretation of Data Mining Outcomes.* Berlin-Heidelberg.: Springer Verlag, 2005, pp. 25–35.

[32] P. Buono and M. Costabile, *Lecture Notes in Computer Science: Integrated Publication and Information Systems to Information and Knowledge Environments*. Berlin-Heidelberg.: Springer Verlag, 2005, ch. Visualizing Association Rules in a Framework for Visual Data Mining., pp. 221–231.

[33] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI Magazine*, pp. 37–54, 1996.

[34] M. C. Chen, J. R. Anderson, and M. H. Sohn, “What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing,” in *CHI ’01: CHI ’01 extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM Press, 2001, pp. 281–282.

[35] Hibernate, “Hibernate framework,” 2007. [Online]. Available: http://www.hibernate.org

[36] ECMAScript, “Ecmascript language specification.” [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf

[37] “Xquery 1.0: An xml query language.” http://www.w3.org/TR/XQuery/, January 2007, w3C Recommendation.

[38] “Xquery api for javatm (xqj).” http://jcp.org/en/jsr/detail?id=225, November 2007.

[39] M. Dick, J. Sheard, C. Bareiss, J. Carter, D. Joyce, T. Harding, and C. Laxer, “Addressing student cheating: definitions and solutions,” *SIGCSE Bull.*, vol. 35, no. 2, pp. 172–184, 2003.

[40] T. S. Harding, D. D. Carpenter, S. M. Montgomery, and N. Steneck, “The current state of research on academic dishonesty among engineering students,” in *Proceedings of FIE 01*, vol. 3, 2001, pp. 13–18.

[41] S. Mulvenon, R. C. Turner, and S. Thomas, “Techniques for detection of cheating on standardized tests using sas,” in *Proceedings of the Twenty-Sixth Annual SAS Users Group International Conference*, 2001, pp. 1–6.

[42] H. Shao, H. Zhao, and G.-R. Chang, “Applying data mining to detect fraud behavior in customs declaration,” in *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 3, 2002, pp. 1241–1244.

[43] M. May, S. George, and P. Prévôt, “Tracking, analyzing and visualizing learners’ activities on discussion forums,” in *Proceedings of the Sixth IASTED International Conference on Web Based Education*, vol. 2, 2007, pp. 649–656.

[44] T. Mochizuki, H. Kato, K. Yaegashi, T. Nagata, T. Nishimori, S.Hisamatsu, S. Fujitani, J. Nakahara, and M. Suzuki, “Promotion of self-assessment for learners in online discussion using the visualization software,” in *Proceedings of the 2005 Conference on Computer Support for Collaborative Learning*, 2005, pp. 440–449.

[45] J. Hardy, S. Bates, J. Hill, and M. Antonioletti, “Tracking and visualization of student use of online learning materials in a large undergraduate course,” in *Proceedings of the 6th International Conference on Web-based Learning*, 2007, pp. 280–287.

[46] M. Sasakura and S. Yamasaki, “A framework for adaptive e-learning systems in higher education with information visualization,” in *Proceedings of the 11th International Conference on Information Visualization (IV’07)*, 2007, pp. 819–824.

[47] G. K. L. Tam, R. W. H. Lau, and J. Zhao, “A 3d geometry search engine in support of learning,” in *Proceedings of the 6th International Conference on Web-based Learning*, 2007, pp. 248–255.

[48] Q. V. Nguyen, M. L. Huang, and I. Hawryszkiewycz, “A new visualization approach for supporting knowledge management and collaboration in e-learning,” in *Proceedings of the Eighth International Conference on Information Visualisation (IV’04)*, 2004, pp. 693–700.

[49] C. G. da Silva and H. da Rocha, “Learning management systems’ database exploration by means of information visualization-based query tools,” in *Proceedings of the Seventh IEEE International Conference on Advanced Learning Technologies*, 2007, pp. 543–545.

**Gennaro Costagliola** received his Laurea degree in Computer Science from the University of Salerno, Italy, in 1987. Following four years post-graduate studies, starting in 1999, at the University of Pittsburgh, PA, he joined the Faculty of Science at the University of Salerno where, in 2001, he reached the position of Professor of Computer Science. His research interests include web systems and e-learning, and theory, implementation and applications of visual languages.

He is a member of the Association for Computing Machinery and the IEEE Computer Society.

**Vittorio Fuccella** received the Laurea degree in Computer Science (cum laude) from the University of Salerno (Italy) in 2003. In 2007 he received the PhD in Computer Science at the University of Salerno. He is currently a Research Fellow in Computer Science at the University of Salerno. He won the Best Paper Award at ICWL’07. His research interests include Web engineering, Web technologies, Data Mining and e-learning.

**Massimiliano Giordano** received his Laurea degree in Computer Science from the University of Salerno, Italy, in 2003. Currently he is a PhD student in Computer Science at the University of Salerno. His research interests include Access Control, Data Mining, Data Visualization and e-learning.

**Giuseppe Polese** is an associate professor in the Department of Mathematics and Computer Science at the University of Salerno, Italy. His research interests include visual languages, databases, e-learning, and multimedia software engineering. He received the Laurea degree in Computer Science from the University of Salerno, a Master in Computer Science from the University of Pittsburgh, USA, and a PhD in Computer Science and Applied Mathematics from the University of Salerno.