A library of templates for exercise construction in an adaptive assessment system

EDUARDO GUZMÁN & RICARDO CONEJO

Department of Languages and Computer Science.
School of Informatics, University of Málaga. Apdo. 4114, Málaga 29080. SPAIN
e-mail: {guzman,conejo}@lcc.uma.es

Conventional computer testing systems usually offer items (questions) where examinees can only select one or more answers. In these systems, there is a fixed number of items to be administered to students. This implies that all students must answer the same number of questions, independently of their knowledge level. By contrast, adaptive testing systems are able to make more accurate predictions of student knowledge level with shorter tests, by choosing the most adequate item to ask next, depending on the current estimation of student knowledge level. In order to be reliable, adaptive testing requires a well-founded underlying theory. The theory mainly used is the *Item Response Theory*. On the other hand, items that usually appear in adaptive tests have a simple and maybe boring format: a stem and a set of answers. In this paper, a library of templates for the automatic construction of more sophisticated items is presented. These types of items are used in SIETTE, an adaptive web-based system for knowledge assessment by means of tests. This system also provides the capability of generating isomorphic items. With the same mechanism used in the construction of this library, SIETTE has been provided with the capability of restricting the time consumed by students while taking tests.

Keywords: assessment, cognitive diagnosis, Computerized Adaptive Tests, Item Response Theory, psychometrics, automatic item generation, Intelligent Tutoring Systems.

INTRODUCTION

Assessment is an important part in the process of instruction. Teachers need to measure somehow the knowledge acquired by students. It is especially important in Intelligent Tutoring Systems and generally in Computer Educational Systems, where the instruction process is adapted in terms of the student performance. Consequently, during the development of this kind of systems, adequate mechanisms for assessment are required. Testing is one of the most extended mechanisms for this purpose, mainly because it is relatively easy to

implement a test-based diagnosis module. The final goal of a test-based assessment system is to determine the proficiency or knowledge level of students.

There are great deal of commercial test-based tools, some examples are (Intralearn Software Corp., 2003), (Webassessor, 2001), (WebCT Inc., 2003), (WBT Systems, 2003). These systems offer powerful interfaces to teachers and students. One of the main lacks of these systems is that the assessment process is not well founded. In most cases they only present to examinees a set of fixed questions, and then, they return the percentage of items correctly answered, without regarding the greater or lower difficulty of the questions administered. On the contrary, educational systems based in well-founded theories do not take advantage of the multimedia capabilities available thanks to technology advances. As a result, in general, teachers and students prefer commercial systems that are more attractive and versatile, even though the assessment criteria they use are not well founded. This issue has been pointed out by authors like (Weber & Brusilovsky, 2001). They postulate that researchers may spend more effort in the development of powerful interfaces, but without forgetting the application of well-founded techniques.

A *Computerized Adaptive Test* (CAT) (Wainer, 1990) is a test where items (commonly known as questions) are posed to student in a personalized way. In general, in this kind of tests, items are shown to students one by one. The next item to administer will be selected in terms of the previous student response. In addition, the finalization, and hence the total number of items administered to students is dynamically adopted depending on the estimation of the student's knowledge level. SIETTE is an efficient adaptive web-based system for assessment. It delivers CATs. The mechanisms used to carry out the selection of the most suitable question (*item*) to administer and the test finalization criterion are based on a psychometric theory called *Item Response Theory* (IRT).

One disadvantage of test-based systems is that all items have the same or a very similar format: just a stem and a set of answers. It has been inherited from paper-and-pencil tests where it was very difficult to assess simultaneously different types of items using the Classical Theory (Thissen, 1993). This simple format may cause students get bored. Some efforts should be applied to improve the interface of this items but keeping unchanged the underlying theoretical model. Recently some IRT researchers are applying other kind of items (Osterlind, 1998) in adaptive testing. In this paper we present a library to automatically construct exercises that can be used as items in a CAT environment. These exercises are assessed in the same manner in which test items are. The goal of this library is to offer a powerful and amusing interface to student, but without losing the

rigorous assessment mechanism. These templates intend to be a complete collection of all generic exercises that teachers can pose. The use of this library provides the capability of posing exercises and tests questions in the same assessment session.

The introduction use of innovative items in IRT based systems has been widely discussed. (Huff & Sireci, 2001) pointed out that innovative items might enhance validity in computer-assisted assessment. They said that multiple-choice and paper-and-pencil items are not adequate to assess higher level skills such as reasoning, synthesis and evaluation. The introduction of innovative items accomplishes a best assessment of knowledge, skills and abilities. In (Boyle, Hutchison, O'Hare, & Patterson, 2002) a study which compares innovative items with classical items is shown. The test contained items from published sources and newly written items. It was composed of interactive complex items and simple dichotomous items, i.e. items only scored as right or wrong. The former were administered through a test delivery platform called TRIADs (*Tripartite Interactive Assessment Development system*) (Mackenzie, 1999). This system has been developed by the Centre for Interactive Assessment Development of the University of Derby. In this study items were analyzed from the Classical Test Theory and the IRT paradigms. It showed that simple items, i.e. multiple-choice items, had lower reliability and discrimination than complex items. That is, innovative items are more useful to make an accurate assessment.

In the next section, the SIETTE architecture is presented. Section 3 is devoted to briefly explain the modus operandi of an adaptive test. Following, in the same section, the student knowledge level estimation mechanism as well as the criteria used in item selection and in test finalization are approached. Later, the different types of items offered by SIETTE will be discussed. In section 5, the components of the library will be explained in detail. Section 6 focuses in the item automatic generation mechanism provided with SIETTE and how it can be combined with the library. Section 7 briefly shows how using the same mechanism to implement the components of the library, SIETTE has been supplied with temporized tests or items. Finally, the main contributions of our work is summarized.

THE SIETTE SYSTEM

SIETTE (this stands for *System of Intelligent Evaluation using Tests for Teleeducation* in Spanish) (Conejo, Guzmán, Millán, Pérez-de-la-Cruz, & Trella, (to appear)) is a test-based assessment system that has been designed to be used through WWW. By means of a web browser, teachers can create subject (or courses)

structured hierarchically in topics (or concepts). These topics are assessed through questions (named items in this context). Tests are specified by selecting the topics involved. Examinees can self-assess their knowledge through tests delivered. Some security mechanisms have been developed to avoid cheating, restrict the availability of tests in time, and to restrict tests to some groups of students.

Although SIETTE has been conceived as a CAT delivering tool, it can also operate as a conventional assessment tool, i.e. using fixed tests with the same number and the same items in the same order for all students. These tests are assessed with criteria like the percentage of correct items over the total number of items, or with scored items, that are the classical heuristics used by other tools.

SIETTE can work itself as an independent tool for testing. It can be also integrated into web-based Intelligent Tutoring Systems. In these systems, SIETTE may play the role of a diagnosis module for generation and updating student models. In general, one of the main problems of tutoring systems is the initialization of the student model. Student models managed by SIETTE are overlay models over the curriculum defined by teachers. Adaptation must be done according to some preliminary data, which in this case are missing. Thanks to the hierarchically structured curriculum of SIETTE and to its complete assessment mode, a pretest of the whole subject to be instructed can be accomplished. As a result, SIETTE will give back to the tutoring system a detailed student assessment represented by means of a probability distribution curve of the student knowledge level for each one of the topics composing the curriculum (Guzmán & Conejo, 2002b). Additionally, after each instructional step, test-based assessment can be carried out in order to infer if student has enough knowledge to learn new concepts. Currently, SIETTE is being integrated into the architecture of MEDEA (Trella, Conejo, Guzmán, & Bueno, 2003), a web-based platform for the development of Intelligent Tutoring Systems.

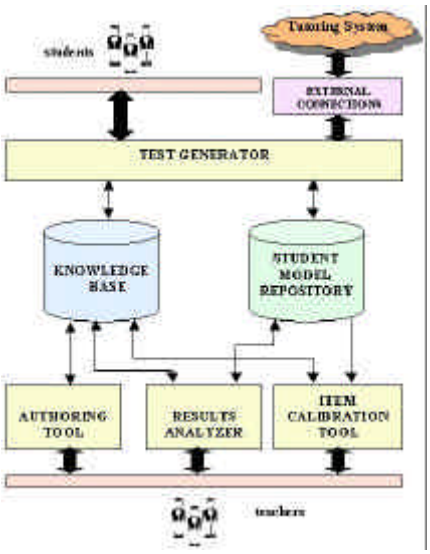


FIGURE 1 The architecture of SIETTE

The architecture of SIETTE is depicted in Figure 1. It is mainly organized in six modules:

- *Knowledge base.* It is mainly composed by the topic domain (*curriculum*), the specifications of the tests and the item pool.
- *Test generator.* This is the virtual dassroom where students take tests. It dynamically constructs test sessions. A test session is a tailor-made test for a student according to the specifications stored in the knowledge base.
- *The authoring tool.* This is a web-based utility only accessible for authorized teachers. It is used to add and update the contents of the knowledge base.
- *The student model repository.* It is a collection of student models. Each student model stores the information about a student's test session. It is dynamically updated after each response. These data are used by the test generator. The student model will play an important role in the item calibration process.
- *Result analyser.* This utility allows teachers to analyse the performance of students in tests. Moreover, it provides a tool to analyse the item parameters and the entire calibration process.
- *Item calibration tool.* When teachers add new items to the knowledge base, they must estimate heuristically the parameters of their items. These parameters are only an initial approximation, so a calibration process is required. This process is an essential process to ensure a suitable adaptation in testing. This module uses the information obtained from the student model repository.

- *External connection interface.* Through this part of the architecture, web-based tutoring systems can interact with SIETTE, using it as a diagnosis module inside their own architectures. A communication protocol has been developed to this end (Guzmán & Conejo, 2002a).

ADAPTIVE ASSESSMENT

As (Wainer, 1990) indicates the main idea of an adaptive test is to act in the same way a teacher would do. That is, if a teacher asks student an easy item and the student successfully answers, the next item to administered will be less difficult and vice versa. This process should go on until the teacher is able to estimate the qualification of student. In more precise terms, a CAT can be seen as an iterative algorithm that starts with an initial estimation of the examinee's proficiency level and has the following steps:

1. All the questions in the knowledge base (that have not been administered yet) are examined to determine which is the best item to ask next according to the current estimation of the examinee's knowledge level.
2. The question is asked, and the examinee responds.
3. According to the answer, a new estimation of the proficiency level is computed.
4. Steps 1 to 3 are repeated until the stopping criterion defined is met.

This procedure is illustrated in Figure 2.

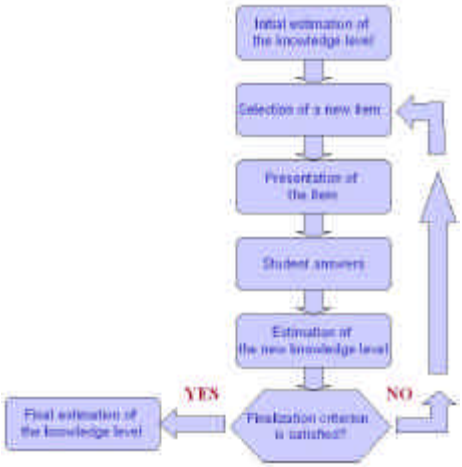


FIGURE 2 Flow diagram of an adaptive test (adapted from (Olea & Ponsoda, 1996))

The most important steps of a CAT are the estimation of the student knowledge level, the selection of the next item to pose as well as the decision of test finalization. These three steps are described in the two subsections below.

Student knowledge inference

Generally CAT systems implement IRT in order to estimate the student knowledge level. This theory is applied to determine the next item to be administered at each moment as well as to decide the finalization of the test. IRT {Van der Linden & Hambleton 1997 #196} was imposed as a better alternative to the *Classical Test Theory* (CTT) that was used in the early days of the 20th century. One of the most important is that the examinee ability is strongly joined to the test features. The ability of an examinee is defined as "the expected value of observed performance on the tests of interest". As a consequence, the ability only makes sense in the framework of a certain test. For more information about the disadvantages of CTT see (Hambleton & Swaminathan, 1985).

IRT is based on the hypothesis that the answer given to each item of the test depends probabilistically on certain *latent trait (q)* that can be measured by means of an unknown fixed numerical value. This theory has been successfully applied to examinee knowledge level estimation in adaptive tests. In educational environments the *latent trait* is the *knowledge level* of the student. In this theory, conditional probabilities of the successful answer to the item by a student with a certain *knowledge level* must be previously known for each item. This probability is expressed by means of a function $f:(-\infty,+\infty)\rightarrow[0,1]$ that is called *Item Characteristic Curve* (ICC). The calculus of the ICC can be accomplished by several models. SIETTE uses a model of three parameters based on the logistic function (Birnbaum, 1968):

$$P_i(q) = P(u_i = 1 \mid q) = c_i + (1 - c_i)\frac{1}{1 + e^{-1.7a_i(q - b_i)}} \quad (1)$$

In equation 1, u_i=1 represents that the answer to the item *i* is correct. The opposite case distribution function, i.e. $P(u_i=0|q)$ is equals to $1-P(u_i=1|q)$. The three parameters that characterize this ICC model are:

- *Difficulty* (b_i): It corresponds to the knowledge level in which the probability of success is equal to the probability of failure.

- *Discrimination factor* (a_i): It is proportional to the slope of the curve. A high value indicates a high probability of success for students with knowledge higher than the difficulty of the item.
- *Guessing factor* (c_i): It is the probability that a student with no knowledge gives a correct answer to the item, and can be computed as the proportion of right answers over the total number of answers.

In SIETTE, a discrete version of IRT has been implemented. For this reason, the latent trait q can only take K discrete values (from *0* to *K-1*), and therefore ICCs are given by vectors of K components $P_i(u|q) = (p_i(u|q=0), p_i(u|q=1),... p_i(u|q=K-1))$ calculated using equation 1.

The value of q is estimated using the response to each item. There are several methods to get this value. In SIETTE a Bayesian method (Owen, 1975) is used. In this method, the knowledge level is inferred from the posterior knowledge distribution of the student. If the final knowledge level is the expected value of this distribution, this technique is called *Expected A Posteriori* (EAP). Else, if the estimated level corresponds to the mode of this distribution, it is called *Maximum A Posteriori* (MAP). The probability distribution of the student's knowledge level is calculated applying Bayes' rule:

$$\overline{P(q|u)} = \left\| \prod_{i=1}^{n} \overline{P_i(q)}^{u_i} (\bar{1} - \overline{P_i(q)})^{(1-u_i)} \overline{P(q)} \right\| \qquad (2)$$

Equation 2 shows that, thanks to the discretization used, the Bayesian estimation can be simplified to a product of ICC vectors of the administered items and the prior normalized density vector. Vertical bars represent that the distribution obtained must be normalized, i.e. the sum of all its value must be one. Note that in this product, ICC values taken can be either positive ($\overline{P_i(q)}$) or negative ($\bar{1} - \overline{P_i(q)}$) in terms of the correct or incorrect response given, that is in terms of the value of u_i. In this paper, to simplify, a dichotomously scored version of the assessment mechanism of SIETTE is being considered. Consequently, items only can be evaluated as correct or incorrect. No partially correct items are considered. In addition, in order to calculate the prior distribution *P(q)* several methods has been proposed (Thissen & Mislevy, 1990). In SIETTE, if there is no previous information stored in the knowledge base about the student, a uniform distribution with equally likely values is used as the initial estimation.

Item selection and finalization criteria

In SIETTE, teachers must indicate the item selection criterion of each test. After the selection, the chosen item is removed from the temporal pool of the current test session. As a result, each item will appear once at much. There are three alternatives. Note that only the first two criteria are adaptive:

- *Bayesian criterion.* Starting from the distribution of the estimated knowledge level of the student, the selected item is that one which minimizes the sum of the *a posteriori* variances resulting from a correct/incorrect answer to the item.
- *Difficulty-based criterion.* This method selects that item whose difficulty is closer to the estimated knowledge level of the student. The selection made by this method is equivalent to the one made by the previous criterion. Hence, both techniques were developed by (Owen, 1975). This last criterion is advantageous because requires less computational cost to be performed.
- *Random criterion.* This method randomly selects an item.
- Difficulty ordered criterion. It orders, before the first selection, all items in the pool in a increasing ordered difficulty. Each time the less difficult available item is selected.
- Explicit ordered criterion. Sometimes teachers prefer that the selection will be accomplished following a preestablished order indicated by them.

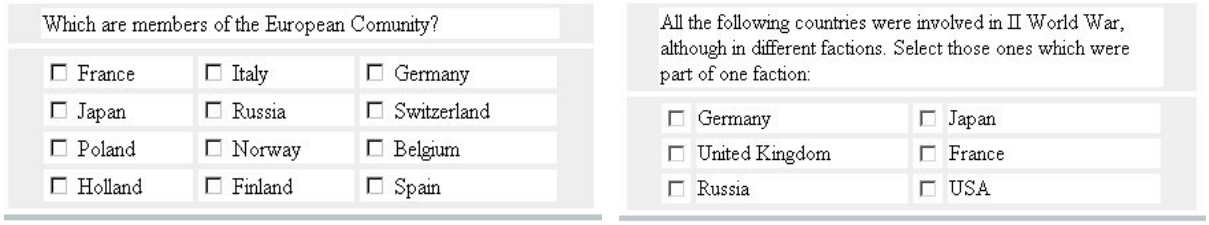
The finalization criterion is configured in each test too. First, in the edition phase, teacher must indicate a minimum and a maximum number of items. Assessment will not finish until the minimum number of items is administered. When the maximum number is reached, the current estimation of student's knowledge level becomes the final estimation. This upper bound is set in order to prevent very large tests. Additionally SIETTE offers the following adaptive finalization criteria: a) the most likely value of the estimated knowledge distribution is bigger than a certain threshold; or b) the variance of the estimated knowledge distribution is lower than a certain value. These two adaptive criteria refer to the precision with which the teacher wants the student knowledge level to be inferred. Anyway they are taken into account only when the number of items administered is between the two boundaries.

TYPES OF ITEMS IN SIETTE

In SIETTE, an item is associated to one or more topics. This association represents that this item provides information about the student knowledge level in these topics. Teachers are allowed to administer different types of items, all of them into the same test session. There are several ways of categorizing items, but none of them can be completely fitted to items available in SIETTE. In this paper, the taxonomy proposed in (Parshall, Davey, & Pashley, 2002) will be mainly followed. The types of items currently available are:

- *True/false items*: These items ask about the certainty of a statement. It can only have two answers: true (correct) or false (incorrect).
- *Multiple-choice items*: This kind of items presents more than two possible answers (choices) Examinees may select one of these answers, or even none of them.
- *Multiple response items*: These items are multiple-choice items with more than one correct answer. Examinees must select all correct answers in order to pass the question. This kind of items can be also classified into:

a) *Items with independent answer.* Answers are mutually independent. This type of item is equivalent to a set of true/false items. For instance, the item shown in Figure 3 (a) is equivalent to the following true/false items: *Is France member of the European Community?*, *Is Italy member of the European Community?*, etc.



(a) *(b)*

FIGURE 3 Multiple response items: (a) with independent answer, (b) with dependent answer

b) *Items with dependent answer:* The correct answers are combinations of the set of possible answers. For the answer to be correct, all members of a combination must be selected. For example, in Figure 3 (b), two combinations are correct. The first one is formed by the options Germany and Japan, and the second one is the combination of United Kingdom, Russia, France and USA. This kind of items is equivalent to a

multiple-choice item where the answers are all possible combinations. That is, if the item has *n* possible answers, it is equivalent to a multiple-choice item with factorial of *n* possible answers.

- *Self corrected items.* Items interfaces are pieces of HTML code in SIETTE. Therefore little programs embedded in web pages like Java applets or Flash movies can be added to the stem or to any of the choices to enhance presentation. SIETTE provides another kind of items where the assessment mechanism is accomplished by a little embedded program itself. This type of questions does not offer a list of possible responses. In this case the student must interact with the little program. The student actions are caught and processed by the program to determine whether the answer is correct or not. Several specific tests have been developed in SIETTE using this kind of items, like a Piagetian test for cognitive ability estimation (Arroyo, Conejo, Guzmán, & Woolf, 2001), a test of European trees geographical distribution, etc.

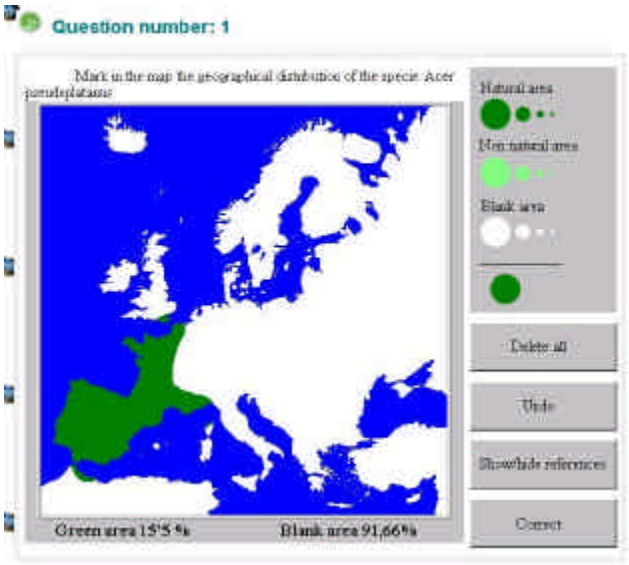


FIGURE 4 A figural item controlled by means of a Java applet

Figure 4 shows an example of a figural item from the *European trees geographical distribution* test. The goal of this question is, by means of a paintbrush, to select the European regions where certain species of tree can be found. Once the examinee has selected a region in the map, he must click on the "*Correct*" button. Then the applet will compare the region selected by the student with the correct region (allowing certain degree of error). When the applet has classified the answer of the student in terms of its correctness, it gives the result of this assessment to SIETTE.

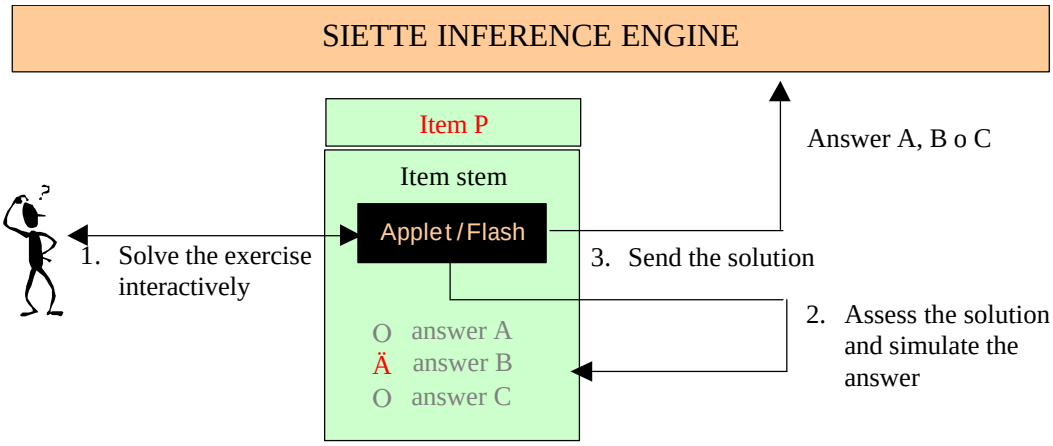


FIGURE 5 Assessment process by a self-corrected item

These items have a set of possible answers, but in this case they are hidden and only known by the little program. Figure 5 shows a schema of this assessment mechanism. First, the item is shown to the student. He interacts with the program inside the item. Once the student has solved the problem, the applet corrects his performance, and then it internally selects the corresponding (hidden) answer. It is automatically sent to the assessment mechanism of SIETTE. See (Arroyo et al., 2001) for a complete description of this mechanism.

These kinds of items can also be classified as true/false or multiple-choice items, depending on the number of possible responses that the little program uses to assess the student's answer. Thanks to this type of items, SIETTE offers the possibility to include virtually any kind of item (as long as it can be implemented by means of a web embedded program), keeping the assessment mechanism unchanged. Certainly, this possibility is restricted to test developers with some programming skills.

THE LIBRARY OF EXERCISE TEMPLATES

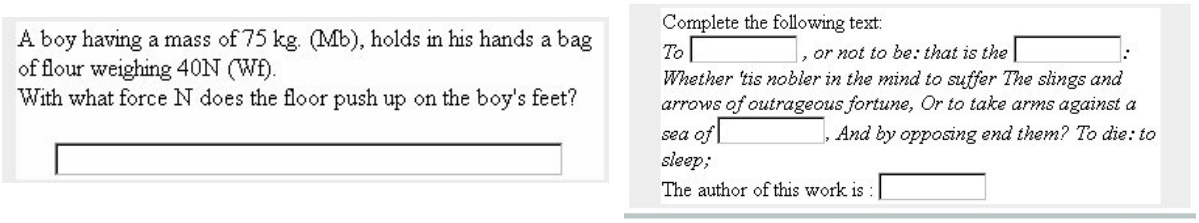
Additionally to the items shown in the section above, SIETTE includes a library of templates for items. This library allows teachers to administer different kinds of exercises with powerful interfaces. It does not only offer typical items of simple picking the correct answers like the items shown in Figure 3, but intends to be a complete collection of all different types of exercises that usually appear in text books.

All these exercises are self-corrected items and have been implemented by means of Java applets, but in this case, the library allows teachers to automatically construct items for any kind of test. They only have to properly instantiate the templates of the library. This task can be easily accomplished by using a wizard in the authoring tool provided with SIETTE. Accordingly, in order to use this library, teachers do not need to have any

programming skills. In addition, all types of items constructed with this library or not can be merged in the same test.

The items that can be constructed by means of the library are the following:

- *Fill-in-the-blank items*, in which the examinee has to fill some blanks in a text. The correctness of the answer is checked by means of regular expressions (in the case that the blank needs to be filled with text) or formulas that allow a selectable percentage of error (in the case that the answer is a number), or a combination of both. Examples are shown in Figure 6.



(a) *(b)*

FIGURE 6 Fill-in-the-blank items: (a) numeric (b) text

The teacher using regular expressions can introduce the solution. For the example in Figure 6(a) the set of possible answers can be described by the following regular expression:

```
#(Mb+Wf/9.81)3% ([Kk][Gg]|Kilograms) |
#(9.81*Mb+Wf)3% ([Nn][Ww]|Newtons)
```

while the student's answer might be any valid combination (*e.g.* `11,72 Kg`).

These questions are internally transformed into *n* true/false items, where *n* is the number of blanks in the composed item.

- *Ordered response items.* In these items students have to sort a set of *n* elements by a drag-and-drop mouse operation. Elements can be text or images. These items are transformed into multiple-choice items with *n!* possible answers. An example can be seen in Figure 7.

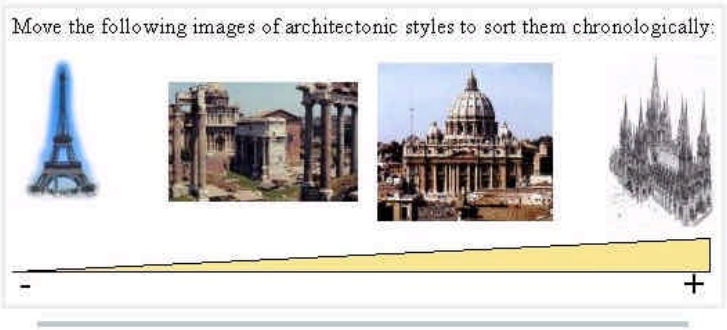


FIGURE 7 An ordered response item

- *Inset items.* In this kind of items examinees must select the elements that fulfil a condition expressed in the stem. These elements must be inserted into a set by means of drag-and-drop operations. Element can be text or images. These items are equivalent to multiple response items with independent answers. Figure 8(a) shows an example of this type of items. This example is equivalent to the example of Figure 3(a), but it offers a more attractive interface.
- *Matching items.* Two columns of *n* elements are presented, text or graphics. The students must link each element of the left column with one element of the right column. These items can be transformed into a multiple-choice item with *!n* choices. The objective of the example of Figure 8(b) is to link each country with its capital. In this case the item is equivalent to a multiple-choice item with 5! possible answers. In this case it is important to manage each answer independently to cover partially correct answers.

As shown, each exercise that can be constructed by using these templates has an equivalent among the items presented in the section above. Therefore the addition of this library has involved neither significant modifications in SIETTE architecture nor in its adaptive assessment and item selection mechanisms.

Pastime items

In order to improve the presentation of items, SIETTE offers some other exercise templates in the style of pastimes. These items do not seem to be very useful from a rigorous assessment point of view. They have been included to make test session more amusing specially when self-assessment is carried out. In general, these items will have very low difficulties and will not be very reliable to determine the examinee knowledge level. Some of these types of templates are:

- *Word search items.* These items consist of locating the missing word in a matrix of letters by selecting a set of aligned letters with the mouse. In the example shown in Figure 8(c) the students have to locate the numbers (1 to 8) written in Spanish.

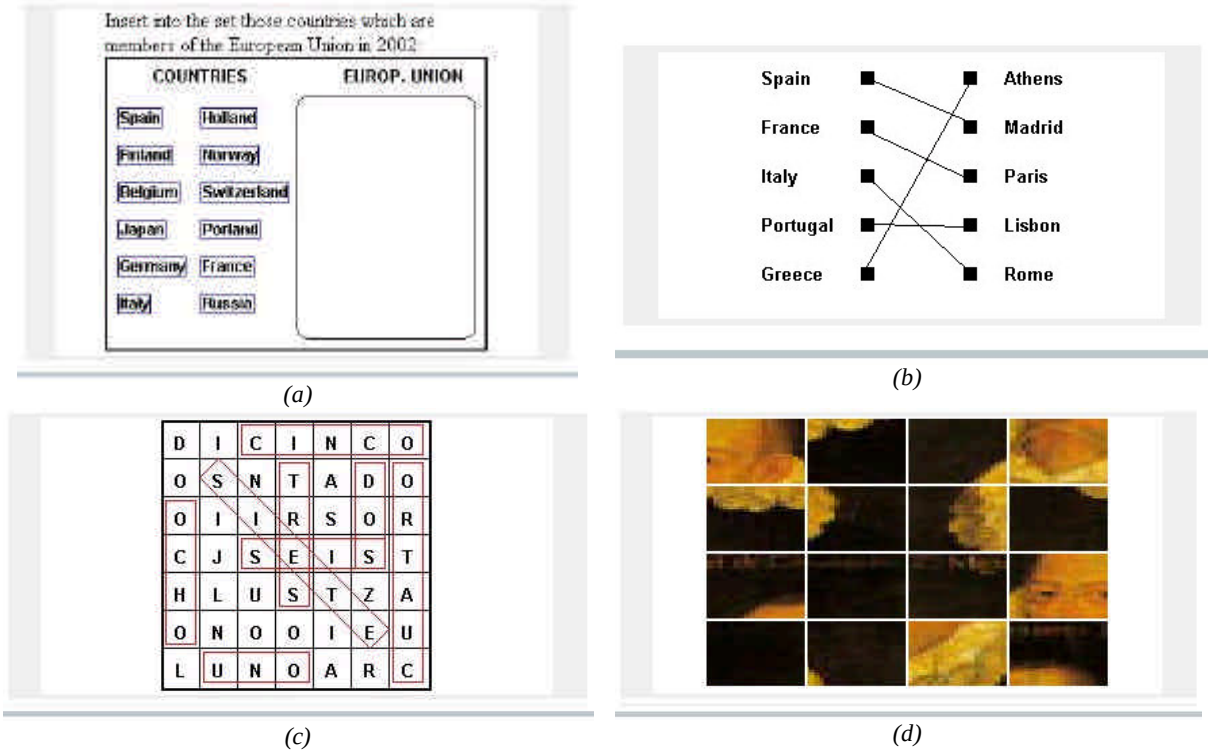


FIGURE 8 (a) Inset item. (b) Connection item. (c) Word search item. (d) Puzzle item.

- *Puzzle items*. This is the classical Frank Lloyd's puzzle. In these kinds of items students must order the pieces of an image to get the correct shape of the object represented in the image. In Figure 8(d) students must sort the pieces of the puzzle in order to get the picture of the renowned writer Miguel de Cervantes.

At a first sight, this library has a disadvantage. In general, the code of a web page can be easily seen using an option usually provided by most of the navigator tools. Consequently, the code of an applet embedded in a page should be seen too. This means that examinees can guess the correct answers by inspecting the source code of the item web page. To avoid this cheating, when an item is going to be shown, all its answers are coded. This codification is included in the web page as applet parameters. Inside the applet, this codification can be decoded, since methods have been developed to this end. All the applets of the library as well as all self-corrected items of SIETTE inherit from an abstract class. This class, developed inside the kernel of SIETTE, is responsible of implementing all the communication with the inference engine and the codification/decodification capabilities.

As a result, when the applet accomplishes the assessment, it internally decodes the answers corresponding the student performance and returns the correction to SIETTE in a decoded way.

AUTOMATIC GENERATION OF ITEMS

To have a valid CAT delivering system, the item pool should contain a very large number of items (it is recommended at least 500 items) (Flaugher, 1990). An excessive item exposure might cause that examinees learn the correct answers and even share it with the remainder examinees. This must be avoided to guarantee a valid assessment. Therefore, it means that teachers must add a huge set of items to the pool. This is a very hard and time-consuming task. A partial solution to this problem is the automatic generation of items. A review of the generation problem can be seen in (Belmonte, Guzmán, Mandow, Millán, & Pérez-de-la-Cruz, 2002).

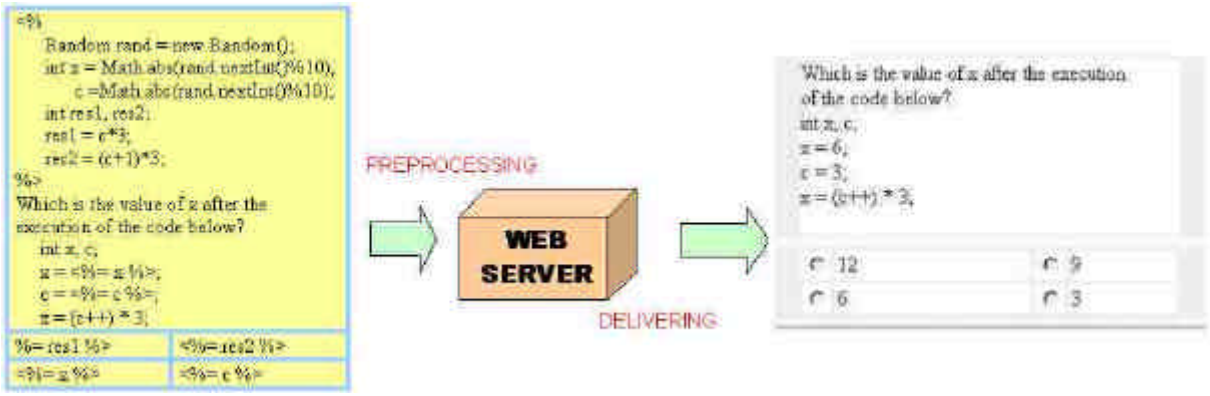


FIGURE 9 Generation and delivering of an item template

SIETTE makes possible the creation of templates to generate isomorphic items in real time. When teacher develops a template, each time it is administered to an examinee, a different item is generated. The templates can be implemented by using HTML-embedded languages, such as PERL, PHP, JSP, and so on. The inclusion of a generative item is transparently done. In the creation of an item, teacher must indicate whether the item is generative or not, by marking an option in the authoring tool. Instead of adding HTML code in the stem and in the answers, teacher must add pieces of the HTML-embedded languages with random value generation sentences. These sentences are in charged of generating the correct and incorrect answers according to the stem. Before the item is shown, a pre-processing step is accomplished. A buffered web page is composed with the stem and the answers. This page is sent to the web server with the appropriated plug-in to support the embedded language. The web server generates a page with just HTML code. This page instead of being shown, is returned

to SIETTE. Thereby, the stem and answers are separated again and composed in the final web page that will be shown to the examinee.

An example of a template in JSP is shown in Figure 9. It is a question about Java language. The main code of template as well as statement are stored in the stem. The code that generates each answer is stored in the place of the answers. The values of *x* and *c* variables are generated every time the item is administered. Note that this mechanism allows generating distractors, i.e. answers that even though they are not correct, they are similar to the correct answer. In the item generated placed in the right of Figure 9, the correct response is the second one, but if the student has some, but not enough notions of Java, there is a higher probability of selecting this incorrect answer than the others. Some IRT-based models are able to take into account the response selected by students and make an estimation according to the degree of error. Currently SIETTE does not support these models.

This generation mechanism has the same lack as the self-assessment items, that is, teachers must have programming skills. In order to provide automatically in some way this feature, generative capabilities have been added to some exercise templates. Ordered response, inset and matching items can be created like simple items or like generative templates. The main idea is that teacher must supply a big set of elements (answers) to the item, and indicate the number of them that should be shown to an examinee every time. As a result, in a test session, when one of these items is selected, the system will generate an item by choosing the predetermined number of elements among the set initially provided.

For instance, let us suppose a teacher has included in his subject an item such as the one of Figure 8(b). That is, through the authoring tool, he has selected the exercise template of matching items and supplied the following correct combinations: Spain-Madrid, France-Paris, Italy-Rome, Portugal-Lisbon, Greece-Athens, USA-Washington, UK-London, Russia-Moscow, Germany-Berlin, Japan-Tokyo, etc. He has also indicated he wants the item to be generative, and that every time the item will be posed only five combinations will be shown. When this item is selected by the adaptive engine, five pairs of elements will be selected, randomly ordered and shown to examinee. An analogous mechanism is applied in the remainder two types of generative exercise templates.

TEMPORIZED TESTS AND ITEMS

Most psychometric models tries to collect as much information as possible about the interaction with students. The main goal is, with this information, to make estimations accurately. From the early days of psychometry, there have been diverse attempts to use the response time as a part of the student response. Firstly this time was related to the correction of items (White,). Recently some IRT models that restrict the response time used have been developed (Verhelst, Verstralen, & Jansen, 1997), (Roskam, 1997).

In addition, the supporting library provides the capability of making temporized tests. SIETTE is used to estimate the *knowledge level*, not any psychometric *latent trait*, so it uses a simplified approximation. Although SIETTE stores the item response time used by each student, it does not include time as a parameter of the ICC yet. Anyway it provides a mechanism to add time constraints. There are three possibilities:

- *Non-temporized tests*. In these tests the examinees have as much time as they need to finish the test. It is recommended that adaptive tests use this option.
- *Temporized-item tests.* In this type of tests, each item is assigned a maximum time (in the edition stage). If the student does not answer the item on time, the generator will automatically show the next one. To calculate the new estimated knowledge level of the student, the generator assumes that the student has not answered this item.
- *Temporized tests.* These tests have a maximum global time to answer all questions.

This feature has been implemented using a clock applet with the same mechanism used in the implementation of the templates of the supporting library. The clock begins to count down when the HTML page, with the first item, has been loaded. In a temporized test, if the student clicks on the *Next question* button, the clock stops and the time value is passed as an initial value for the next item. If the time finishes before the student answers the item, the clock applet will force the finalization of the availability of the item, firing the final estimation of the knowledge level of the student, and therefore the test ends. In temporized-item tests, if the student clicks on the *Next question* button, the clock stops but there is no need to store the remainder time, because each item has its own independent clock time. Figure 10 shows the look of a temporized test. The clock is shown in the upper right side of the page.

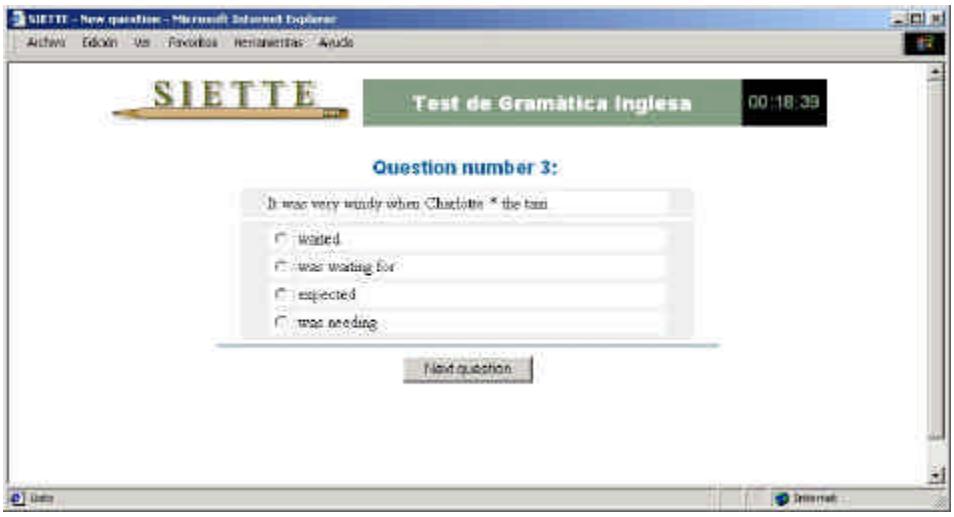


FIGURE 10 A temporized item

Teachers can design tests where some items are temporized and other are not. They only have to select, in the test editor, the temporal option for each item. Thanks to the dynamic addition of this temporal characteristic, items can be easily converted into temporized items and vice versa. Equally, complete tests can be easily set or unset as temporized by means of the test editor options. Of course the ICCs change if a different maximum answering time is selected. There is no special feature yet implemented to deal with time varying ICCs.

CONCLUSION

Conventional assessment systems based on tests pose to the examinees a fixed number of items, not making any distinction between students in terms of their ability. Advanced and inexperienced students take the same tests. This feature enforces teachers to design tests for students with a medium level of knowledge. Also, all these items are usually administered in the same order. Therefore, students may get bored/disappointed while taking tests because questions are too easy/difficult for them.

Fortunately, adaptive test-based assessment systems can improve these drawbacks. The number of items in a test depends on the ability of the student. The system estimates, after the presentation of each item, the new knowledge of the student and the selection of the next item is done according to this level. Consequently, the next item posed to the examinee is the most informative. This process continues until the system considers it has enough information to correctly diagnose the student knowledge level.

A priori, one of the disadvantages of using CAT is that traditionally these systems only deliver tests with a format very near to the paper-and-pencil tests. But as (Thissen, 1993) argues, the items to which the models of IRT may be applied may not be individual questions, they may not even appear to be test items, and they may vary in format within a single measurement instrument. This makes the combination of IRT and CAT a powerful tool to assess any kind of exercises in a well-founded way.

SIETTE has been presented as a platform to support the inclusion of virtually any kind of item. It extends considerably its capabilities. This platform offers to teachers the possibility of assessing by using adaptive criteria based on IRT, or conventional criteria such as percentage of item successfully answered. An entire collection of exercises that usually appear in textbooks has been included in a library. Now teachers without programming skills can design tests with different kinds of questions and exercises. As a result, tests can be richer and more amusing. Anyway this is an open system in the sense that teachers can development their own items and easily include them in their tests. This feature is available thanks to the implementation of these templates by means of Java applets, which can be inserted into HTML pages. Any test developer with programming skills can add new templates to the library using the authoring tool. The same mechanism has been used to provide SIETTE with temporized items and tests. In this line, other libraries are under development. They are libraries of *Psychotechnical test* and other libraries specific for subjects like *Logic, Compilers*, etc.

CAT delivering systems require huge item pools in order to avoid that students memorize item and even share them with other students making invalid the assessment process. But item construction is a hard task and requires great efforts by teachers. An attempt to solve this problem is the use of generative items. SIETTE brings teachers the opportunity of automatically generating items using some exercise templates. In addition, teachers with programming skills can implement their own generative items and easily include them in their tests.

SIETTE can be accessed and tested at http://www.lcc.uma.es/SIETTE . Items constructed with this library are shown in several tests of *Demo* subject.

References

Arroyo, I., Conejo, R., Guzmán, E., & Woolf, B. P. (2001). An Adaptive Web-based Component for Cognitive Ability Estimation. *Proceedings of the 9th World Conference of Artificial Intelligence and Education AIED'01* Amsterdam: IOS Press.

Belmonte, M. V., Guzmán, E., Mandow, L., Millán, E., & Pérez-de-la-Cruz, J. L. (2002). Automatic Generation of Problems in Web-based Tutors. In L. C. Jain, R. J. Howlett, N. S. Ichalkaranje, & G. Tonfoni (Eds.), *Virtual Environments for Teaching & Learning* (Vol. 1pp. 237-281). London: World Scientific .

Birnbaum, A. (1968). Some Latent Trait Models and Their Use in Inferring an Examinee's Mental Ability. In F. M. Lord, & M. R. Novick (eds), *Statistical Theories of Mental Test Scores* . Reading, MA: Addison-Wesley.

Boyle, A., Hutchison, D., O'Hare, D., & Patterson, A. (2002). Item Selection and Application in Higher Education. *Proceedings of the 6th International Computer Assessment Conference (CAA)*. (pp. 269-284).

Conejo, R., Guzmán, E., Millán, E., Pérez-de-la-Cruz, J. L., & Trella, M. (to appear). SIETTE: A web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education*.

Flaugher, R. (1990). Item Pools. In H. Wainer (ed.), *Computerized Adaptive Testing: A Primer* . Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Guzmán, E., & Conejo, R. (2002a). An adaptive assessment tool integrable into Internet-based learning systems. *Sociedad de la Información: Educational Technology: International Conference on TIC's in Education* (pp. pp. 139-143).

Guzmán, E., & Conejo, R. (2002b). Simultaneous evaluation of multiple topics in SIETTE. In *Lecture Notes in Computer Science 2363. Proceedings of the 6th International Conference on Intelligent Tutoring Systems ITS 2002* (pp. 739-748). Berlin: Springer Verlag.

Hambleton, R. K., & Swaminathan, J. (1985). *Item response theory: principles and applications.* Boston: Kluwer.

Huff, K. L., & Sireci, S. G. (2001). Validity issues in computer based testing. *Educational Masurement: Issues and Practice, 20*(3), pp. 16-25.

Intralearn Software Corp. Intralearn [Web Page]. URL http://www.intralearn.com [2003, April 21].

Mackenzie, D. M. (1999). Recent developments in the Tripartite Interactive Assessment Delivery System (TRIADs). In *Proceedings of the 3rd International Computer Assessment Conference (CAA)*.

Olea, J., & Ponsoda, V. (1996). Tests adaptativos informatizados. In J. Muñiz (ed.), *Psicometría* (pp. 731-783). Madrid: Universitas.

Osterlind, S. J. (1998). *Constructing Test Items: Multiple-Choice, Constructed-Response, Performance and Other Formats* (Second Edition). Evaluation in Education and Human Services . London: Kluwer Academin Publishers.

Owen, R. J. (1975). A Bayesian Sequential Procedure for Quantal Response in the Context of Adaptive Mental Testing. *Journal of the American Statistical Association, 70*(350), 351-371.

Parshall, C. G., Davey, T., & Pashley, P. J. (2002). Innovate item types for computerized testing. In W. J. van der Linden, & C. A. W. Glas (eds.), *Computerized Adaptive Testing: Theory and Practice* (pp. 129-148). Dordrecht (Netherlands): Kluwer Academic Publishers.

Roskam, E. E. (1997). Models for Speed and Time-Limit Tests. In W. J. Van der Linden, & R. K. Hambleton ((eds.)), *Handbook of Modern Item Response Theory* (pp. 187-208). Berlin : Springer-Verlag.

Thissen, D. (1993). Repealing rules that no longer apply to psychological measurement. In N. Frederiksen, R. J. Mislevy, & I. Bejar ((eds.)), *Test Theory for a New Generation of Tests* . Hillsdale, NJ: Lawrence Erlbaum Associates.

Thissen, D., & Mislevy, R. (1990). Testing Algorithms. In H. Wainer (ed.), *Computerized Adaptive Testing: A Primer* (pp. 103-136). Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Trella, M., Conejo, R., Guzmán, E., & Bueno, D. (2003). An educational component based framework for Web ITS development. *Lecture Notes in Computer Science. Proceedings of the 3rd International Conference on Web Engineering (ICWE 2003)* Berlin: Springer Verlag.

Verhelst, N. D., Verstralen, H. H. F. M., & Jansen, M. G. H. (1997). A logistic model for Time-limit tests. In W. J. Van der Linden, & R. K. Hambleton ((eds.)), *Handbook of Modern Item Response Theory* (pp. 169-186). Berlin: Springer-Verlag.

Wainer, H. (1990). *Computerized Adaptive Testing: A Primer*. Hillsdale, NJ: Lawrence Erlbaum.

WBT Systems. TopClass [Web Page]. URL http://topclass.uncg.edu/ [2003, April 21].

Webassessor [Web Page]. URL http://www.webassessor.com/webassessor [2003 June 6].

WebCT Inc. WebCT [Web Page]. URL http://www.webct.com [2003, April 21].

Weber, G., & Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education, 12*, pp. 351-383.

White, S. M. C. I. G. I. In H. J. Eysenck ((ed.)), *A model for Intelligence* . New York : Springer-Verlag.