Journal of Universal Computer Science, vol. 11, no. 9 (2005), 1482-1493
submitted: 15/2/05, accepted: 30/6/05, appeared: 28/9/05 © J.UCS

A Tool for the Reinforcement of Conceptual Learning: Description and Use Experiences

Roberto Moriyón
(Escuela Politécnica Superior, Universidad Autónoma de Madrid, Madrid, Spain
Roberto.Moriyon@uam.es)

Francisco Saiz
(Escuela Politécnica Superior, Universidad Autónoma de Madrid, Madrid, Spain
Francisco.Saiz@uam.es)

Abstract: In this paper we describe the DeepTest tool, which is intended to reinforce the conceptual learning of any subject by means of interactive exercises for the detection of incorrect texts. DeepTest can be used through Internet. The generic aspects of the tool are analyzed, and a first report on conclusions from the use of the tool by a group of students and teachers is presented. The main conclusion is that DeepTest can be used effectively in assessment tasks and its use is very simple and intuitive.

Keywords: Authoring tool, interactive exercise, conceptual learning
Categories: K.3.1

1 Introduction

In this paper we describe the DeepTest tool, which can be used through Internet and is based on a new type of interactive exercises, those designed for the detection of incorrect texts. This type of exercises is very different from the usual ones, which are still the same traditional ones generally solved by means of pencil and paper. The most frequent ones among them are multiple choice and free answer tests.

The state of the art in this field includes several other types of interactive exercises that have appeared within the last years, although their use is not very extended. An introduction to them and a detailed study from a psychometric perspective are given in [McDonald, 99] and [Zenisky, 02]. Among these types of tests, we shall comment on various types. Tests that consist of the selection and repositioning of graphical objects, which are used to some extent in primary and secondary schools since students find them attractive due to their graphical appeal. Tests for the selection of texts, like those described in [Tomico, 03], are used mainly in courses on reflective reading. These are especially relevant for this paper. Other types of tests are also used, like those based on graphical modeling and concept association. We shall point out the usefulness of tools for the interactive use of semantic networks, which are especially suited for conceptual learning. Their main limitation is that their use is inherently decontextualized as a consequence of the schematic way they represent knowledge.

Apart from the generic tools mentioned in the previous paragraph, nowadays there are more interactive systems for knowledge evaluation, which are based on specialized tools for specific subjects, like environments for the design and execution of mathematical problems, [Diez 04], for simulation and distant access to laboratory equipment, [Dwyer, 97], and for Architecture, [Katz, 88]. The Educational Testing Service of the USA is in charge of evaluating students and professionals from many different scopes at the state and federal level. This institution is responsible for the most novel contributions made to this type of applications and are based on ambitious research plans that have given rise to several patents and many scientific publications of the highest level, [Mills, 02].

The scenario described above shows that in spite of the appearance of new techniques and types of interactive exercises, there is a lack of computer systems and tools that allow students to reinforce different forms of learning. In this paper we shall mainly address conceptual learning. By conceptual learning we refer to deeper, transferable understandings of generalizable, abstract knowledge; that which has to do with logical thinking, the formation of scripts, stories, cases, mental models or constructs, concepts, associations, perspectives and strategies [Roussou, 05]. Conceptual knowledge includes the incorporation of knowledge that is deduced or explicitly related to other knowledge, in situations where these deductions or formal relationships with other concepts are part of the knowledge the student has to assimilate. Our work is also relevant for the learning of subjects with significant terminological components such as human or computer languages.

Conceptual learning can be framed in a broader scope on the basis of the Bloom Taxonomy, [Bloom, 56], which distinguishes six different dimensions related to the competencies acquired when learning: knowledge, comprehension, application, analysis, synthesis and evaluation. Conceptual learning is very closely related to analysis and comprehension ability, although it is also related to other competencies like application, since in many cases it is essential to analyze how actions are accomplished and to understand the mechanisms that govern them in order to learn how to apply the knowledge acquired.

Exercises for the detection of incorrect texts, EDITs, are documents that include types of erroneous concepts, reasoning or information that the students must detect. They can be solved interactively when learning any subject in any language that uses roman characters. Compared with other traditional tests, their degree of interactivity improves the learning process. The main reason is that the use of the tool makes it easier for students to learn from the mistakes they make. The main limitations of the tool have to do with the use of mathematical formulae and non standard characters. These limitations are strictly technical, and future versions of DeepTest will be able to handle documents based on any alphabet that can include mathematical formulae.

EDITs present information in the form of interactive documents that look like static ones, which allows a contextualized work. EDITs designers have absolute freedom to use arbitrary parts of documents in order to establish a context for the solving of exercises, and other parts, which are indistinguishable a priori for the students. In this way students are asked in an implicit way to check for the correctness of the knowledge they have acquired. DeepTest includes a tool for the creation of EDITs, and an environment for their interactive resolution. The authoring tool allows interactive exercises to be defined starting from arbitrary HTML documents. It also permits teachers to collaborate by sharing collections of exercises. These aspects are fundamental, since it is well known that one of the greatest barriers for the expansion of the use of computers in education is the enormous development cost of high quality interactive contents.

Since EDITs are based on documents that include text and images, the appearance of the authoring tool and the exercise resolution environment is similar to that used with a document editor, thereby reaching a high level of usability. The evaluation performed with a group of students described in section 3 below confirms this. In addition, teachers who use the tool in order to create EDITs do not need any specialized knowledge about computer usage. Moreover, the user interface for the design of interactive exercises is very similar to that used for their resolution. This is very advantageous for teachers when designing exercises, since they can anticipate the difficulties students are going to find in their work.

The DeepTest system is protected a patent pending from the European Patent Office, hold by the Universidad Autónoma of Madrid. DeepTest can be integrated as an additional service in a platform for computer assisted education, like WebCT, [WebCT, 05] or Moodle, [Moodle, 05], using standards for communication among educational applications like IMS, or SCORM, [ADLNET, 05].

This paper is organized as follows: the next section is devoted to the description of the main generic aspects of DeepTest. The following section describes the experience of DeepTest being put to use at the Universidad Autónoma de Madrid by means of a collection of EDITS in a course on Object Oriented Programming, [Alfonseca, 04]. Finally, ongoing work and future plans are briefly described.

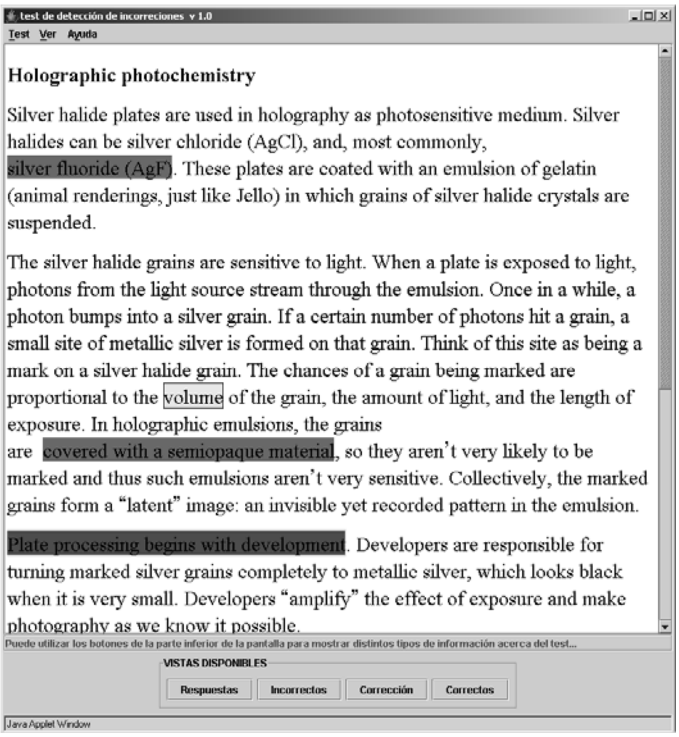
2 Description of DeepTest

The DeepTest tool is available for use through Internet, [DeepTest, 04]. Users can register and accomplish their work through courses that include interactive exercises. Each course includes a group of students that are enrolled in it. In this context, a teacher can start a program for the design of EDITs to be integrated with static contents, and the students that are associated can follow the course and start another program for the resolution of the exercises.

2.1 Exercises for the detection of incorrect texts

EDITs consist of interactive digital documents related to the subject under study, which include statements, references or words that are not correct. These parts of the text can be used to reflect typical fundamental mistakes. In this way DeepTest can reinforce conceptual learning. By means of the mouse or the keyboard, students have to detect those parts of the text where these incorrect statements appear. When finished and while grading their answers, DeepTest provides them with feedback about their knowledge, using different colors to show their selections as well as those incorrect statements that have not been detected, as depicted in figure 1. The colors used to show these parts of the text depend on their correspondence to incorrect statements or not. A mark is also assigned on the basis of a positive value associated by the designer to each erroneous statement or word, which is applied to all the zones properly selected by the student. A negative value is similarly assigned to each text selection, which is applied to the inadequate selections. The global mark is the sum of all positive and negative values.

In addition, all parts of the documents cited previously can include a link to an explanation about their correctness or incorrectness, which can be accessed by means of a hypertext mechanism. Moreover, students can interactively switch the type of information shown to them, like for example the visualization of the correct/incorrect version of each incorrect/correct alternative.



Holographic photochemistry

Silver halide plates are used in holography as photosensitive medium. Silver halides can be silver chloride (AgCl), and, most commonly, silver fluoride (AgF). These plates are coated with an emulsion of gelatin (animal renderings, just like Jello) in which grains of silver halide crystals are suspended.

The silver halide grains are sensitive to light. When a plate is exposed to light, photons from the light source stream through the emulsion. Once in a while, a photon bumps into a silver grain. If a certain number of photons hit a grain, a small site of metallic silver is formed on that grain. Think of this site as being a mark on a silver halide grain. The chances of a grain being marked are proportional to the volume of the grain, the amount of light, and the length of exposure. In holographic emulsions, the grains are covered with a semiopaque material, so they aren't very likely to be marked and thus such emulsions aren't very sensitive. Collectively, the marked grains form a "latent" image: an invisible yet recorded pattern in the emulsion.

Plate processing begins with development. Developers are responsible for turning marked silver grains completely to metallic silver, which looks black when it is very small. Developers "amplify" the effect of exposure and make photography as we know it possible.

Puede utilizar los botones de la parte inferior de la pantalla para mostrar distintos tipos de información acerca del test...

Figure 1: Grading of an interactive exercise

Exercises for the detection of incorrect texts are prone to a high degree of ambiguity. This is especially relevant in portions of the text that are statements. For example, the statement *Nero was born in 150 A.D.* may be a mistake because of the wrong date or the wrong name of the emperor. Similarly, the statement *two plus three equals six* can be considered a mistake in five different ways: either because the word *two* appears instead of *three*, or because *plus* appears instead of *times*, because *three* appears instead of *four*, because *equals* appears instead of *is less than*, or because *six* appears instead of *five*. We call this disjunctive ambiguity. In order to cope with disjunctive ambiguity, designers can use two mechanisms: on the one hand, they can use the document as a way to define a context for the exercise; for example, in the first case if the subject of the document is *Nero*, the mistake will be in the year and not in the emperor's name. On the other hand, DeepTest allows designers to create exercises with specific types of errors like words or sentences exclusively. For example, if the last exercise above asks for the detection of incorrect sentences, there is no disjunctive ambiguity.

EDITs can also present what we call cumulative ambiguities. An ambiguity of this kind also appears in the first example above, even after being disambiguated by the subject as explained in the previous paragraph. In this case the student might consider that the region of error is the whole sentence or just the year. In this case the semantics of the contents of error are not ambiguous like in disjunction ambiguities, but the exact location is ambiguous. Once more there are two different mechanisms that can be used by the designer when dealing with this type of ambiguity: after defining an incorrect version that corresponds to some correct statement or work, designers can specify a part of the text that contains it as the largest acceptance zone, so that any selection by the student that contains the original mistake (the number *150* in our example) and is contained in the biggest acceptance zone (the whole sentence would be an appropriate one in this case) is accepted. Like in the case of disjunctive ambiguities, the second possibility restricts the type of possible mistakes to sentences or words.

The main difference between exercises generated by means of DeepTest and those created by means of other systems is that DeepTest exercises can be solved by students in a highly interactive context, giving rise to a clear reinforcement of conceptual learning. This is due to the fact that besides learning by discovering concepts that may be wrong, they have to understand in depth the ideas that are expressed in the document as a consequence of the fact that they do not know where the wrong information is, or the type of mistakes that are included in the document.

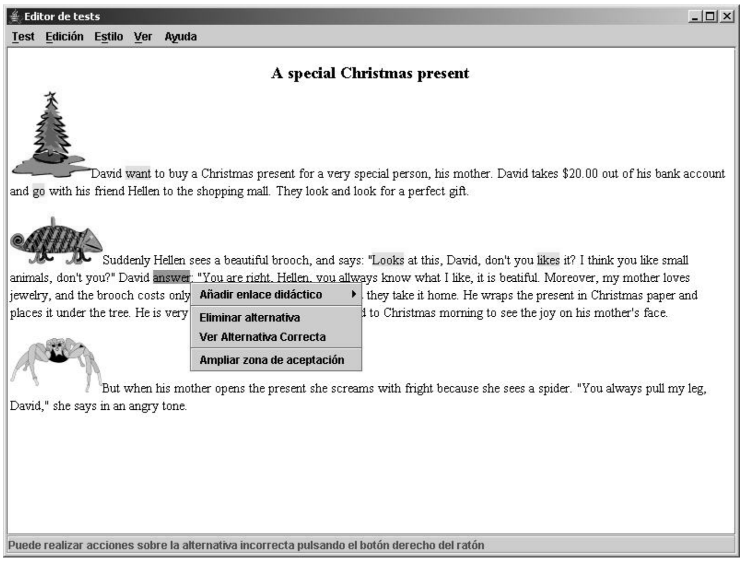
Moreover, the design of EDITs and their resolution are accomplished in similar environments by using mechanisms that are similar to the ones used when working with a text processor. The DeepTest design tool is actually an HTML editor that allows designers to define incorrect parts of documents by just selecting the corresponding part of the original documents one by one and activating a command that allows them to be replaced by their incorrect counterpart. Designers can also assign a value to each correct or incorrect part of the text, as well as links to explanations that correspond to correct or incorrect parts of the text. Finally, designers can specify the biggest acceptance zone that corresponds to an erroneous area by just activating the corresponding design command and marking the acceptance zone with the mouse.

2.2 User interface of the authoring tool

The DeepTest authoring tool has a very simple user interface. It consists of a standard edition area where the document is shown, a lower bar that shows help messages that are adapted to the working context of the designer of interactive exercises, and a menu area with the usual menus in an editor, together with other menus to deal with the specific information on erroneous areas. As depicted in figure 2, all the editor functions that correspond to these specific aspects of EDITs are accessible through the right button of the mouse, by means of contextual menus. This is a design decision for the user interface that is a key for the usability of the tool, since users can use the resulting application exactly in the same way they use a standard editor for HTML documents, which is something they are used to. Only when dealing with the specific tasks related to the interactivity of the final exercises do they have to access it through two menus where all these tasks are available. Thanks to this design, the excellent results concerning the usability of the tools are shown in section 3.

The contextualized help given to users by DeepTest displays a list of tasks that can be carried out at each moment in the lower part of the windows as well as explaining how to do so. In addition, a comprehensive help system complements the previous mechanism and is available from the help menu at any moment.

The implementation of the authoring tool is based on an HTML document editor whose functionality has been extended in order to add the interactivity that corresponds to interactive exercises to it. The main task designers can accomplish is the creation of new incorrect alternatives corresponding to a text that has been selected. When activating this command, the tool hides the selected text and the designer can introduce the desired incorrect version.



A special Christmas present

David want to buy a Christmas present for a very special person, his mother. David takes $20.00 out of his bank account and go with his friend Hellen to the shopping mall. They look and look for a perfect gift.

Suddenly Hellen sees a beautiful brooch, and says: "Looks at this, David, don't you likes it? I think you like small animals, don't you?" David answer: "You are right. Hellen. you allways know what I like, it is beatiful. Moreover, my mother loves jewelry, and the brooch costs only . they take it home. He wraps the present in Christmas paper and places it under the tree. He is very d to Christmas morning to see the joy on his mother's face.

But when his mother opens the present she screams with fright because she sees a spider. "You always pull my leg, David," she says in an angry tone.

Puede realizar acciones sobre la alternativa incorrecta pulsando el botón derecho del ratón

Figure 2: Design of an interactive exercise

Once an incorrect version has been created, the user can add an explanation to it, see the corresponding correct alternative, and expand its corresponding acceptance zone, as explained before. The incorrect alternative can also be deleted, in which case the corresponding part of the document is restored to its initial form. All these actions are available through contextual menus, as shown in figure 2.

2.3 Implementation aspects

The authoring tool represents exercises by means of HTML documents that include intervals with an associated semantic annotation. The decision to use semantic annotations associated to parts of documents was based on the requirement to define extensible mechanisms that can be used in future types of interactive exercises, like the ones described in the section on conclusions and future plans at the end of the paper. DeepTest semantic annotations are data structures that can have an arbitrarily complex structure. This allows the inclusion of complex semantics, like the syntactic role played by a part of a sentence. In the case of intervals with erroneous information, the semantics is represented by a chain of characters (“True” or “False”).

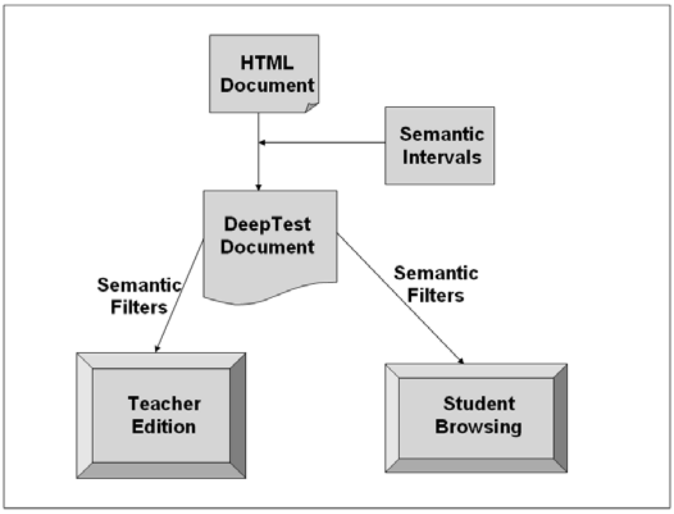


Figure 3: Architecture and working mechanisms of DeepTest

Semantic filters are an essential component in the design of DeepTest. Filters are formed by semantic structures or patterns of them, together with associated presentation styles. They are applied to interactive documents, as depicted in figure 3. They show intervals with semantic annotations using the presentation styles associated to them. For example, figure 2 shows an interactive document during its design, when a filter is applied that shows “False” semantics intervals with a yellow background and hides those with “True” semantics. In order to see the correct version of the document instead of the previous one, a different filter that shows the intervals with “True” semantics and hides the ones with “False” semantics can be applied. The use of filters simplifies both the modularization and reuse of the code related to the reaction of the system under inputs from the user. The DeepTest tools show a document and apply different semantic annotations to parts of it depending on the user’s actions. Depending on the state of the tool successive filters can be applied. For example, figure 1 shows the grading of an interactive exercise using a grading filter that is activated when the system passes on to the corresponding state.

DeepTest was developed in Java 2. Swing and its Model-View-Controller architecture were used in the development of the editor, while the implementation of the distributed system is based on J2EE. The main functional requirement behind this decision was the need for a framework for the definition of HTML editors that allowed the redefinition of user interaction with several aspects of visualization. Although it was not critical and its use was never necessary, we also wanted to have the source code of the framework just in case we needed to accomplish something that was not possible in its implementation. After looking for editors that satisfied these requirements, we ended up with two suitable systems for our needs, and stable enough to be trusted. These were the Mozilla HTML editor and the HTML editor included in Java 2 under javax.swing.text.html. Despite having advantages like covering a more recent version of HTML, the Mozilla editor is the result of the integration of components written in different languages, and uses a specific programming language for the specification of the user interaction, so we decided to use the Swing editor, which is a pure Java editor. This in turn automatically gave rise to the use of a rich and elaborated Model-View-Controller architecture.

3 A first experience using DeepTest

DeepTest was tested with a group of 60 students from a second course on Object Oriented Programming offered by the Higher Polytechnic School at the Universidad Autónoma of Madrid. The course has three parts: the first part is devoted to the study of Smalltalk programming language, the second to the analysis and design of object oriented computer programs, with a special emphasis on Unified Modelling Language, UML, and in the third part the computer programming language C++ is studied. Since the 1995-96 academic year both the mid-term and final exams in this course have always included some questions under the form of Smalltalk or C++ programs that include selected mistakes that the student must detect and comment on. The resolution of this type of problem is very suitable by means of DeepTest. The exams given in this course from the beginning can be found in [Alfonseca, 04]. The test included interactive exercises made from pre-existing exam questions, and some others designed specifically using DeepTest. Two teachers were involved in the design work.

The test was carried out at the computer lab during the last week of classes. The students worked with DeepTest in groups of two. Before starting to work, the students were given a ten minute explanation of how to use DeepTest. During the test five people were ready to solve the problems that might arise.

The goal of the initial test was to detect possible usability problems with the tool and to check its suitability to reinforce learning in computer programming courses. The students were asked to fill in a questionnaire after having solved the problems. Twenty four groups answered the survey. The most relevant questions are described below, together with the global results represented in figure 4. The opinions from the teachers were collected by means of an interview, which allowed us to analyze in greater detail aspects that were discovered to be more important. The following subsections show the results of the survey answered by the students. At the end of this section, the opinions of the teachers are analyzed.

3.1 Usability related questions

1. Simplicity of use: Seventy per cent of the answers indicated that the system is very simple to use. Twenty percent indicated that its use is simple. Five percent indicated that it is complicated.
2. Adequacy of available information: Twenty per cent of the students missed a systematic explanation of the way the tool should be used. The rest considered the information given to be adequate. It should be noted that at the time the survey was taken only the contextual help indicated in the previous section was available. This problem has been solved with the comprehensive help system.

3.2 Suitability of DeepTest for knowledge assessment

3. Usefulness of DeepTest for self assessment of acquired knowledge: The results for this question were very similar to the ones for the first question. Seventy five percent of the students said DeepTest is very useful for self assessment, while fifteen percent said it is useful, five percent claimed it is more or less useful, and five percent did not find it useful.

   Two students indicated that the type of exercises posed is closely related to the information a compiler gives, something they use every day. This opinion was not shared by other students. We deduce from it that this type of test needs a more profound design since the mere addition of more or less random errors does not give the students more information than a compiler does.
4. Usefulness of DeepTest for assessment by the teacher of the knowledge acquired by the students: The answers to this question were more diverse. Thirty five percent of the students thought DeepTest is very useful for assessment by the teacher, while twenty percent of them considered it is useful, ten percent moderately useful and should be complemented by other forms of assessment, five percent considered it not useful and thirty percent were completely against the use of DeepTest to evaluate their knowledge.

   During the next years we are planning to keep using DeepTest at mid-term and final exams, and we will carry out a more detailed study.

3.3 Suitability of DeepTest for learning reinforcement

5. Adequacy of DeepTest for learning Computer Science subjects different from computer programs: Once more the answers were varied in this case. Only half of the students had a clear opinion on this. Sixty percent of them were very much in favor of using DeepTest in all Computer Science courses, five percent were in favor of it, five percent of them were moderately favorable, and thirty percent were completely against it.
6. Adequacy of DeepTest for learning human languages. Eighty percent of the students were convinced DeepTest would be very useful for learning languages, five percent thought it would be useful, five percent said it would be useless, and ten percent that it would be completely useless.
7. Potential interest of DeepTest for the learning other subjects: Half the students did not express any opinion on this. Seventy percent of those who did said it would be very interesting, while thirty percent said it had no interest.
8. Specific aspects that need to be changed: The students made several proposals, which are indicated below.

   Several students complained about selecting a whole instruction as an error instead of the specific word or token that was erroneous, but the system did not accept their choice. As explained in section 2.1. this difficulty can be solved by disambiguating the exercises. It can be concluded that designers need some training on the use of the authoring tool from a pedagogical point of view.

   Students also said they missed the availability of static contents that can be used to alternate between learning and exercise solving. This has already been solved.

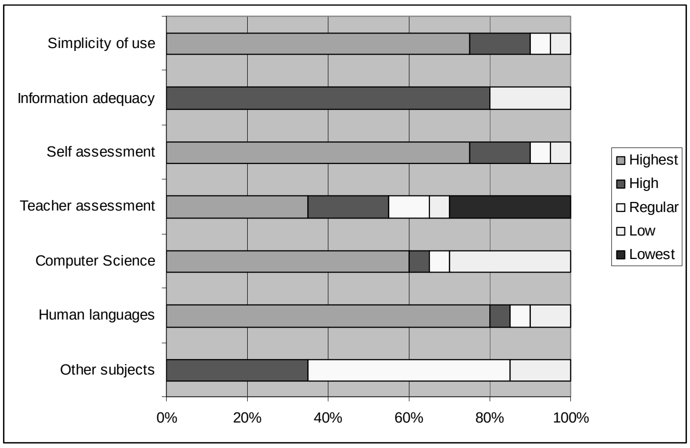


Figure 4: Students evaluation summary

Hence, the survey gave clear answers to the questions related to the simplicity of the tool's use and its utility for self assessment. However, there are divergent opinions about its use for evaluation by the teacher, in spite of the fact that similar exercises have been used for years. We should point out that the opinion of the teachers in this respect was very positive. During the next months, an expert in psychometrics will carry out a more objective evaluation of the possibilities of DeepTest in order to distinguish different levels of contents assimilation. The answers to other questions mentioned above are being contrasted in 2005 with the results from the teaching innovation project mentioned below.

3.4 Opinions from the designers

There were three different types of experience related to the design of EDITs used in the test: as already noted, on the one hand a teacher had designed and used similar exercises solved with paper and pencil for many years. On the other hand, two teachers used the DeepTest tool to design additional exercises. Finally, one teacher used the tool to convert the original paper and pencil exercises into EDITs.

We asked the teachers similar questions to those posed to the students, with reference to the design task instead of the resolution task. The main emphasis was on the usability of the tool and enhancement of conceptual learning and assessment.

There was unanimity about the usability of the design tool, which was considered to be very high, and the usefulness of DeepTest for different forms of assessment in courses on computer languages. The opinions were less unanimous about its use in other Computer Science courses. As a consequence of this test, the tool is now being used in several subjects related to Computer Science by a group of teachers including those involved in the test. No elaborated results are available yet, but the first impressions are positive. It should also be noted that in general, the teachers found it necessary to have a more extensive help system that would cover the subtle aspects of design related to preventing ambiguities, like the definition of extended acceptance areas. The latest version of DeepTest includes a renewed help system for the designing tool, and we still have work to do in order to improve it according to the experiences of designers.

From a teaching point of view teachers consider that two of the most useful aspects of DeepTest are the possibility to easily know what the weak points of the knowledge students acquire from a conceptual point of view are, and the simplicity of designing interactive exercises with the tool.

4 Conclusions and future plans

Using DeepTest to complement education in the classroom can reinforce conceptual learning. DeepTest allows teachers to create tests to assess a deeper knowledge acquired by students than more traditional ones. This has been proved in the evaluation of the tool reported above. Since the tool encourages learning by pointing out mistaken concepts, it can be useful when students learn concepts that are difficult to assimilate correctly.

A teaching innovation project, TRAC, is being conducted during the 2004-05 academic year at the Universidad Autónoma of Madrid. Twenty six professors are using DeepTest in seventeen courses from different subjects. A comprehensive study of its usefulness will be carried out at the end of this experience. The subjects involved in the project are Languages, Ecology, Biology, Biochemistry, International Law, Didactics of Mathematics, Design for the Development of Teaching Contents, Biophysics, Civic Education, Geography, Philosophy, History of Political Theory, Data Bases, Computer Programming, Chemistry, Automata Theory and Translation and Interpreting.

We have many future plans for the tool, including new types of interactive exercises. The extensibility of the mechanisms of semantic annotations we use will allow us to do this while reusing large portions of the work we have done up to now. More specifically, we are working on a resolution environment where students can correct the mistakes they find in documents. The system will use heuristic inference mechanisms in order to detect the areas of the text that have been corrected appropriately and those for which there are errors in the correction.

Acknowledgements

The DeepTest environment for the design and resolution of interactive exercises for the detection of incorrect texts is the result of work in the HyperTest project, funded by the Fundación General de la Universidad Autónoma de Madrid, and the Ensenada, TIC 2001-0685-C02-01, and Arcadia, TIC2002-01948, projects, within the Plan Nacional de Investigación, Spain.

The design of the tool was made by the authors. A. Andrés, D. Mellado, S. Jiménez, A. San Martín, M. Pazos, I. Meléndez, J. Martínez and C. Alonso have collaborated in the development of DeepTest. Smalltalk and C++ problems used in the test described in section 3 were elaborated by M. Alfonseca.

References

[ADLNET, 05] Web de Advanced Distributed Learning, SCORM, http://www.adlnet.org

[Alfonseca, 04] M. Alfonseca et al., Course on Object Oriented Programming 2, Higher Polytechnic School, UAM, http://www.ii.uam.es/~alfonsec/curso4i.htm

[Bloom, 56] B. S. Bloom, (ed.), Taxonomy of educational objectives: The classification of educational goals, Handbook I, cognitive domain. New York, Longmans, 1956

[Diez, 04] F. Díez, R. Moriyón, Design and interactive resolution of exercises of Mathematics that involve symbolic computations, Computers in Science and Engineering, January 2004

[Dwyer, 97] T. M. Dwyer, J. Fleming, J. E. Randall, T. G. Coleman, Teaching Physiology and the World Wide Web: Electrochemistry and Electrophysiology on the Internet, In American Journal of Physiology (Advances in Physiology Education) 273, 6, 1997, S2-S13

[DeepTest, 04] DeepTest, http://DeepTest.ii.uam.es

[IMS, 05] Web de IMS Global Learning Consortium, http://www.IMSProject.org

[Moodle 05] Moodle Course Management System, www.moodle.org

[Katz, 88] I. R. Katz, M. E. Martínez, K. M. Sheehan, K. K. Tatsuoka, Extending the Rule Space Methodology to a Semantically-Rich Domain: Diagnostic Assessment in Architecture, In Journal of Educational and Behavioural Statistics, 24, 1988, 254-278

[McDonald, 99] R. P. McDonald, Test Theory: A Unified Treatment, 1999, Lawrence Erlbaum Associates, Inc.

[Mills, 02] C. Mills, J. Fremer, M. Petenza, B. Ward, (eds.), Proceedings of the ETS colloquium, computer-based testing: Building the Foundation for Future Assessment. Mahwah, NJ: Lawrence Erlbaum Associates

[Roussou, 05] M. Roussou, Can Interactivity in Virtual Environments Enable Conceptual Learning? In Proceedings of the 7th Virtual Reality International Conference, VRIC, First International VR-Learning Seminar, Laval, France, 2005, 57-64

[Tomico, 03] V. Tomico, D. Bolaños, J. Tejedor, N. Morales, I. Orjales, J. Colás, J. Garrido, Interactive Web Platform for Encouraging Reader Comprehension, In Proceedings of the IADIS International Conference WWW/Internet 2003, ICWI 2003, Algarve, Portugal, IADIS 2003, 711-718

[WebCT, 05] WebCT e-learning systems, http://www.WebCT.com. Web de WebCT

[Zenisky, 02] A. L. Zenisky, S. G. Sireci, Technological Innovations in Large-Scale Assessment, In Applied Measurement in Education, 15, 4, 2002, 337-362