



**Repositorio Institucional de la Universidad Autónoma de Madrid**

<https://repositorio.uam.es>

Esta es la **versión de autor** de la comunicación de congreso publicada en:  
This is an **author produced version** of a paper published in:

Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005. Chesapeake, VA: Association for the Advancement of Computing in Education (AACE), 2005. 3072-3079

**Copyright:** © 2005 AACE

El acceso a la versión del editor puede requerir la suscripción del recurso  
Access to the published version may require subscription

# Design and resolution of Exercises based on the Detection of Errors in Documents using the DeepTest tool

Roberto Moriyón  
Universidad Autónoma de Madrid  
C/ Francisco Tomás y Valiente, 11  
28049 Madrid (Spain)  
Phone: +34-91-497-2282  
Roberto.Moriyon@uam.es

Francisco Saiz  
Universidad Autónoma de Madrid  
C/ Francisco Tomás y Valiente, 11  
28049 Madrid (Spain)  
Phone: +34-91-497-2282  
Francisco.Saiz@uam.es

**Abstract.** We present DeepTest, a tool for the design and resolution through Internet of highly interactive exercises based on documents related to virtually any subject. Existing educational contents based on documents can be reused in order to design interactive exercises with DeepTest. We explain the reasons that indicate that in comparison with classical multiple choice or blank filling tests, the work with DeepTest exercises shows much better the level of global comprehension by the student of the matter under study, and encourages the reinforcement of this knowledge. Finally, we explain some ongoing experiences with DeepTest in different subjects.

## Introduction

In spite of the development of Internet related techniques during the last years, the style of computerized tests used to evaluate the work of students or to allow them to know better what they know has not had big changes. In this paper we introduce a novel type of interactive exercises that in some cases are easier to design and are more useful, from the point of view of the student, than traditional multiple choice or filling blanks tests.

We also present DeepTest, a tool for the design and resolution through Internet of highly interactive exercises on virtually any subject. The exercises generated by DeepTest allow students to reinforce the knowledge of the subject they are learning and to be more conscious of their misconceptions by noticing the conceptual errors they make. This is achieved by showing students documents with contextualized information, where they have to work according to some criteria established in each exercise, like detecting some kind of information or the mistakes that appear in the document. Moreover, DeepTest allows very naturally the design of exercises that make students use their global understanding of the whole picture with respect to the matter they are learning. Designing DeepTest exercises is very simple, since the only special knowledge that is required is the edition of documents in a WYSIWYG editor. Existing documents with educational contents can be reused in order to design interactive exercises with DeepTest. Moreover, communities of teachers can share sets of interactive exercises on arbitrary subjects through Internet. DeepTest is protected by a patent pending from the European Patent Office, on behalf of the Universidad Autónoma de Madrid.

Many of the tools that can be used by teachers in order to design interactive learning materials are devoted to the study of specific subjects. This happens for example with virtual laboratories, which can be used when learning experimental scientific disciplines, (Dwyer, 1997). The most widely used generic tools for the creation of interactive educational programs generate different types of tests based on explicit item manipulation, like multiple selection tests, which are the most popular form of such tests. In (McDonald, 1999) and (Zenisky, 2002) an introduction and a detailed description from a psychometric perspective of new types of exercises that have appeared in the last years is given. DeepTest exercises include no explicit items that make the student direct a special attention to predefined aspects of the document they are working with, but instead they encourage the student to examine the whole contents of the document and then elaborate on what they have seen. In this way, information is highly

contextualized in a way that is more natural than in tests based on predefined items, and lends itself to a more global approach to the reinforcement of the learning process.

DeepTest includes an authoring tool for the design of interactive exercises, and an environment for the resolution of the exercises that are designed with it. The authoring tool allows the definition of interactive exercises starting from arbitrary web documents. This is done in an environment that looks exactly as a text processor for the generation of multimedia documents, but with some specific extra functionality added to it. On the other hand, students work in a similar environment, but with a limited functionality that does not allow them to edit the exercises freely, so they can only accomplish some specific edition actions that have a special semantics adapted to the solving task they are doing. Once students finish solving the exercises, they get highly interactive feedback about their work.

DeepTest interactive exercises can check the knowledge of students both in terminological or grammatical aspects with respect to the study of any language (including computer programming languages and human languages) and in more conceptual aspects, like noticing relations among concepts or checking the comprehension of the concepts themselves. For example, students can be asked to check a document for incorrect expressions in a computer program, both from the syntactical and the semantic point of view, or in a text written in a foreign language. In both cases, the mistakes that have to be detected by the students can be associated to general explanations that relate them to the relevant concepts that have been studied by them. Hence, for example, if a student has been working on how to use the different past tenses in French, a document written in French can be shown to him where the tenses are not well used, and the student has to determine when the tenses are correctly used and when they are not. The interactive exercise can include explanations about the general rules that are used in each case. In the same way, if a student has just studied the management of memory in programs written in a specific programming language, and the difference between dynamic variables and static ones, a program can be shown to him with subtle mistakes that mix both kinds of memory and the corresponding ways to manage them.

This paper is organized as follows: Section 2 gives a description of the main aspects of DeepTest. Section 3 analyzes the specific ways in which DeepTest can be used in relation to conceptual learning, and compares them with the limitations of the usual types of interactive tests. It also analyzes the ways in which the bigger possibilities offered by DeepTest give rise to a bigger complexity for the work of the designer, and it shows how DeepTest gives support to minimize this complexity. Section 4 describes some experiences of use of DeepTest, and shows conclusions taken from them. Section 5 describes the architecture of DeepTest. Finally, we describe some plans for future work.

## **A Description of DeepTest**

DeepTest interactive exercises can be created by teachers and accessed and answered by students from any Internet browser by simply accessing a DeepTest server like the one in (DeepTest, 2005) and identifying themselves through a user id and password. This allows them to use the DeepTest access and service providing platform. The DeepTest platform allows free access to some sample interactive exercises, to general information about the system and to the registering process, through which any person can get a standard access to the services offered. Special registering is provided to teachers and designers, with the ability to work with students on courses based on interactive exercises.

DeepTest interactive exercises consist of digital documents related to the subject under study, which include assertions, references, words or images that are not correct. For example, an interactive exercise about photography can include a description of compounds that are used in holographic photochemistry, as in [Figure 1], some of which are either described incorrectly or their role in the process is described incorrectly.

Students are asked to detect the parts of the document that are not correct. They do it by just selecting parts of the document either with the mouse or by means of the keyboard, so they are marked as incorrect. After this, students get the following feedback:

- On one hand, the system shows students those choices made by them that are accepted, since they correspond to incorrect statements in the text that can include allowed surrounding text. These portions of the document are highlighted in green.
- The system also shows students the choices they made that do not correspond to incorrect statements. These are highlighted in red.
- Students are also shown the incorrect parts of the document they did not select, which are highlighted in purple.
- Students also get a grade that is based on a rating that corresponds to each inaccuracy, and a negative rating assigned also to inadequate selections made on different parts of the document. The

teacher can determine the number of points assigned to each adequate and inadequate selection. By default, all incorrect parts are assigned the same value.

- Each of the previously mentioned parts of the document can be linked to an explanation about the correctness or incorrectness of the answer, which can be accessed by means of a hypertext mechanism. These explanations are available when grading the answers given by the student, but not while the exercise is being solved.

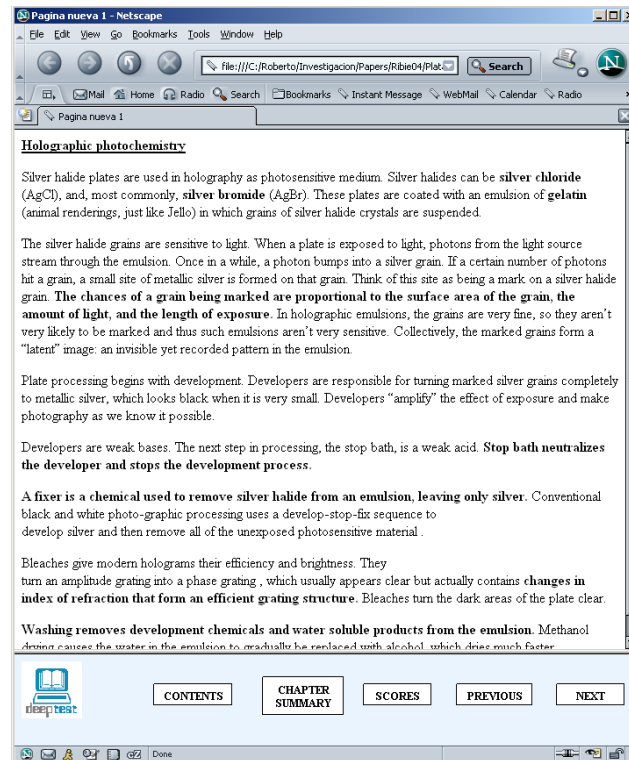


Figure 1: An interactive exercise

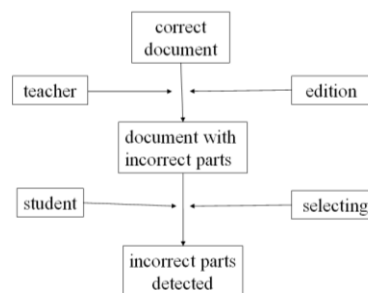


Figure X: The DeepTest users

Finally, students can interactively change the type of feedback they get. The options they have, besides the default feedback explained above are: seeing the correct version of the text, with the parts that correspond to the initial incorrect statements highlighted, seeing the original version of the exercise with the incorrect statements highlighted, and switching for each of these statements between one version to the other. [Figure 2] shows the grading of an exercise on holographic photochemistry based on the document shown in [Figure 1].

When the student notices his conceptual errors and analyzes them using the feedback possibilities we have mentioned, his wrong assumptions are rectified and the level of comprehension of the subject he is studying is reinforced. This is due to the form the wrong concepts and ideas are discovered, which is directly related to the mistakes made while solving the exercises. The level of

depth of the comprehension achieved depends very much on the auxiliary explanation that is given. Hence, the elaboration of suitable explanations is a fundamental step in the construction of successful interactive exercises.

Interactive exercises can be created very easily by teachers without the need of any special knowledge. All they have to do is to open the DeepTest editor, which looks like a standard HTML editor where different text styles, links, etc can be associated to the text that is edited. The DeepTest editor allows also teachers to define alternative incorrect statements by just selecting part of the text and replacing it with its corresponding incorrect version. Teachers can also assign a rating to both incorrect and correct parts of the text. Ratings assigned to incorrect parts of the document are positive and are added whenever the student detects that the part is incorrect; on the contrary, ratings associated to correct parts of the document are negative, and are subtracted each time the student makes a wrong assumption about what is incorrect in the document. Moreover, explanations can be assigned to different parts of the text, both correct and incorrect ones, just as regular links are created on standard HTML documents. Finally, designers can associate to each incorrect part of the text a larger acceptance area. When an acceptance area is defined, any selection of a student that includes an incorrect part and is included in the corresponding extended acceptance area is accepted. This allows teachers to avoid ambiguities occurring when there is more than one possible way to select an error in the text. Typical situations of this kind correspond to selecting part of a sentence or the whole sentence that encloses it.

One of the main uses of incorrect images in DeepTest is as a mean to use mathematical formulae. As it is well known, most documents in the web that include mathematical formulae do it by inserting images that represent the formulae themselves. Designers who want students to show they have understood the formulae involved in the matter they study can define incorrect alternatives to them by marking the formulae they want to change, activating the *create alternative* command, and inserting the alternative formulae either by cutting and pasting them from another document or from a file.

DeepTest also allows for the design of two other simpler types of exercises. Detection exercises ask students just to detect parts of the document that are characterized according to some criteria. For example, an interactive exercise on some language can ask the student to mark all verbs in future tense. Detection exercises are simpler to specify and grade than exercises for the detection of incorrectness, since on one hand there are no alternative versions to specify, and on the other hand the student gets less feedback after his work is done. The second type of exercises supported by DeepTest besides detection exercises are the classical multiple choice exercises. The interfaces for the design and resolution of these exercises are similar to the corresponding ones for the other types, and they are similar in their final functionality to usual tools for the development and resolution of multiple choice tests.

DeepTest servers can communicate with learning management systems in order to allow teachers to associate interactive exercises to courses or groups of students, and to give students access at each instant to the tests that correspond to their state in the course they are working on. DeepTest has its own learning management system that allows the management of contents and exercises to be solved by students, and the revision by teachers of the work done by the students.

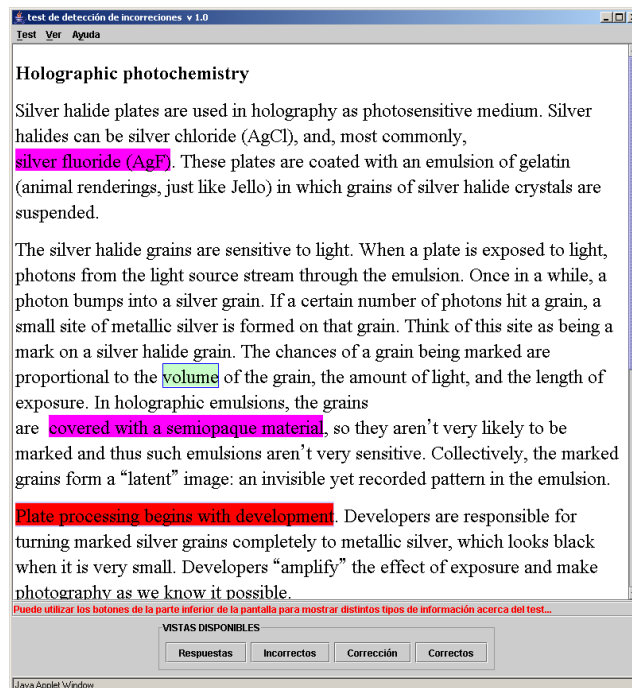


Figure 2: Exercise grading

## Learning with DeepTest

In this section we shall describe how DeepTest can enhance the learning process by giving the student bigger insight about his own degree of knowledge of the subject under study. We start by putting DeepTest under a suitable theoretical framework. This will allow us to have a clearer idea of the possibilities of the system as a tool for learning.

Activity theory is a framework that tries to explain the relationship between humans and objects of the environment as being mediated by cultural means, tools and signs. When applied in the context of learning with computers, it preconizes the use of constructivist techniques. Although it was developed during the first half of the XXth Century by Vygotsky, Y. Engeström, (Engeström, 1999), (Engeström, 1987), has expanded the fields where it can be applied by giving a special role to contradictions. Engeström claims that within any human activity, contradictions emerge and evolve within and between each of the six "corners" of the activity triangle: subject, tool/artifact, object, rules, community, and division of labor. Moreover, he characterizes contradictions saying that they are not points of failure or deficits in the activity system in which they occur, they are not obstacles to be overcome in order to achieve goals, but rather than ending points, contradictions are starting places. According to Engeström, in order to become a source of substantial learning motivation, contradictions must be noticed, faced, and experienced by the learner as a personal challenge. DeepTest interactive exercises can be seen as a simple instrument for strengthening learning by helping students to know their own contradictions.

In order to analyze with more detail the possibilities offered by DeepTest to exercise designers, we shall use the classification of content types proposed by Merrill, (Reigeluth, 1999), and the taxonomy of cognitive operations proposed by Bloom for knowledge evaluation, (Bloom, 1984). The content types considered by Merrill are Facts, Concepts, Principles and Procedures, and the generic cognitive operations from Bloom are Remembering, Understanding, Predicting, Applying and Evaluating. Remembering is especially related to being aware of how well facts and concepts are learned, since usually this type of knowledge is not linked by itself to reasoning, prediction, application or evaluation. Similarly, principles are related to remembering and understanding, while procedures are related to application and, in a more sophisticated setting, to prediction and evaluation.

Among the simplest types of knowledge that can be reinforced by means of DeepTest interactive exercises are concepts. In general, DeepTest allows the design of exercises related to Remembering by just substituting in a text a part of it by another text that is not appropriate under the underlying global context. For example, in a document related to History, the name of a person or place, or the year in which an event took place, can be changed by another one.

Facts, and especially explicit or implicit relations among different concepts, can be treated in DeepTest by more efficient methods from the point of view of learning reinforcement. For example, a

mistake about the association of a date and a specific historical event like the fact that a particular person is the king of a country can be wrong in several ways. We shall show them by means of a specific case: when trying to understand what is wrong in the sentence “Nero lived between the years 31 B.C. and 14 A.D.”, there is a chance that the person who states this sentence is referring to Augustus, who lived during the indicated period of time, instead of Nero, but he might also refer to the life of Nero instead of the interval between 31 B.C. and 14 A.D. If the sentence is part of a document about earlier stages of the Roman Empire, the first interpretation is the right one. This can be deduced by the student from the global context of the document. This type of interactive exercise is especially interesting, since in order to give a correct answer students have to be aware of the global context posed by the document, which can be very structured and complex, put each specific statement of the document in relation with this global context, compare this with his knowledge, and decide what part of it is not compatible with each statement and the global context. Hence, noticing what is wrong in the document associated to the exercise can need naturally a deep knowledge of the subject under study. Moreover, in case the student is not able to find out a mistake in the document, the interactive feedback that is given to him is very striking and motivated, so the student will easily remember in the future what his mistake was.

On the other hand, principles are usually related to a context or hypothesis under which they are accepted or valid, and this gives rise to the same kind of possibilities for the creation of interactive exercises based in the detection of mistakes as in the previous case. Finally, procedures and, in general, application operations, are also subject to validity for the achievement of specific goals, besides to being by themselves actions to remember, so the same ideas can be applied in order to make students practice and discover their mistakes and limitations.

The main difference between DeepTest interactive exercises and other existing types of tests is that the document associated to interactive exercises is the natural working context, both as a source of information and as a place where the answer has to be embedded. When working, students have to understand deeply all the ideas expressed in the document, since they do not know where the information they have to highlight is or what kind of misconceptions are expressed in it. This makes students work more deeply at the conceptual level, especially if the tests are well designed.

If we compare with bigger detail the standard computerized tests with those designed with DeepTest, we see two main differences: on one hand, traditional tests are based on the use of items, which makes students be centred on the specific questions represented by them. On the other hand, DeepTest exercises can be created easily from pre-existing educational materials, like summaries of themes under study. This is impossible with standard tests.

As a counterpart to the advantages of interactive exercises we have described, their use is more complex than that of regular tests. The main difference relies on the possibility of ambiguities in the exercises. The typical example of this complexity is the ambiguity cited above, where Nero is claimed to have been born before Christ. Noticing the occurrence of these ambiguities can be a complicated task, since they can be anywhere within the document. Moreover, there is a different type of ambiguity that can take place: some times the student can select as a mistake one part of the text or another part that contains it. For example, in the sentence “John starts to eat”, a student can think that “to eat” is wrong, or that “starts to eat” is the mistake.

DeepTest gives limited support for some of the complexities just mentioned. First of all, it gives support to alternative errors by means of the context in which the errors appear. It also gives support through the possibility given to the designers to enlarge the area of acceptance that corresponds to any incorrect zone. In this way, the complexity of the design of interactive exercises is smaller than it would be otherwise.

## **Experiences with DeepTest**

DeepTest has been available in the web since the end of 2003. Systematic tests have been accomplished with a group of 60 students of Object Oriented Programming during the spring of 2004. All the exams of this course since the year 1996 included two programs with mistakes that the students had to mark. This was done with paper and pencil up to this year. The exams from previous years are available in (Alfonseca, 2005).

The test took place in the computers lab during the last week of the course. The main goal of the test was to analyze the usability of the tool, and its suitability in order to reinforce the learning in courses on computer programming. The students participated in a survey once the test had finished. Next we give a summary of the answers of the students.

- Most students consider that using the system is simple (95%; 20% of them consider it is very simple).

- There was a minority that missed a complete and systematic help document. This deficiency has been fixed already.
- Most students find DeepTest useful as a tool intended to reinforce learning of subjects related to computer programming (90%; 75% consider it is very useful).
- A limited majority of students find that the tool is useful in order to evaluate them automatically (35% in favour; 20% of them are very much in favour). Studies under way will try to arrive to a conclusion about this fact. Moreover, an expert on psychometrics is going to perform during the next months an objective evaluation about the possibilities of the tool to distinguish different levels of assimilation of contents.

Besides the previous experience, DeepTest is being used in the TRAC project of teaching innovation at the Universidad Autónoma de Madrid. TRAC stands for “Tests for the Reinforcement of Conceptual Training”. 26 teachers and more than 1000 students corresponding to 17 courses with very different characteristics participate in the project. The project is devoted to studying the degree of suitability of DeepTest for the study of each subject. The tool is being used in courses on languages, Ecology, Biology, Biochemistry, International Law, Didactics of Mathematics, design for the development of multimedia material, Biophysics, Civic Education, Geography, Philosophy, History of political theory, Data Bases, programming, Chemistry, Automata Theory and translation.

DeepTest is being integrated into different e-learning platforms like WebCT and Blackboard by using standards for the communication among e-learning applications like IMS and SCORM, (IMS, 2005). DeepTest also includes its own platform for the management of e-learning courses. Moreover, the DeepTest design tool and the environment for the resolution of interactive exercises can be used on any platform as a consequence of their Java based architecture. As a consequence of this, DeepTest interactive exercises and courses that include them can also be used on any platform without the need of any adaptation.

## **DeepTest Architecture**

DeepTest is implemented in Java 2 and uses extensively the Swing framework, (Robinson, 2003). The implementation is based on the use of documents with semantic annotations, which correspond to standard documents where some parts of them have an associated semantics. So, whenever a designer creates an erroneous alternative to part of a document, a hidden annotation is created that indicates that the part created is an alternative to the original one. [Figure 3] shows a scheme of a class diagram for a collection of classes for the representation of documents with semantic annotations. The fundamental classes represent semantic annotations and filters, and will be described below.



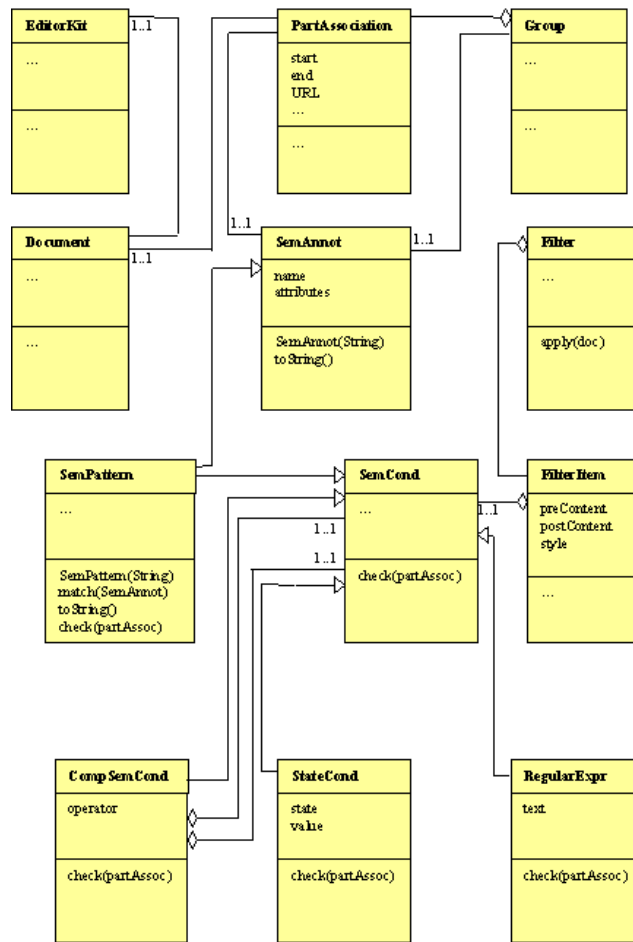


Figure 3: UML diagram for documents with semantic annotations

Semantic annotations, represented by the class `SemAnnot` in [Figure 3], can be represented by means of their name followed by a list of attribute names and values as in `noun(gender → masculine, number → plural)`. Attribute values can also be semantic annotations. The semantic annotations used in `DeepTest` have the form `alternative (incorrect → anyHTMLcontents)` and `answer()`. Semantic annotations are attached to parts of the documents both by the designer at design time and by the student at resolution time. Correction consists in a comparison of the semantic parts defined by the student with those ones defined by the designer of the exercises.

The representation of documents with semantic annotations uses another class in order to display the documents in different ways at different moments. This class represents filters that allow parts of documents to be hidden or to be highlighted in a way that depends on their associated semantic annotations. Filters consist of an ordered set of filter items. Each filter item is formed by a semantic acceptance condition, some optional content to be inserted before corresponding parts of documents, some other optional content to be inserted after them, and a presentation style (a list of specifications of background colour, frame, and character attributes) to be applied to corresponding parts of documents.

Semantic acceptance conditions can be semantic patterns, regular expressions, state conditions or composed acceptance conditions. Semantic patterns are syntactic annotations some of whose attributes can have no value. A semantic pattern matches another semantic annotation if the attributes of the pattern that have a value that is a string are equal to the corresponding ones of the annotation, the ones that have a value that is a semantic pattern match the corresponding ones of the annotation, and the ones without value are defined in the annotation. State conditions impose a value on the state of the system when the condition is tested. Composed acceptance conditions are formed from other acceptance conditions by means of conjunction, disjunction and part comparison (equal, contained, contains, disjoint, overlaps), or by negation of a semantic pattern or state condition. The truth value of an acceptance condition on a part is defined as follows: if the condition is a semantic pattern, it is true if the part is associated to a semantic annotation that matches the pattern. If it is a regular expression, it is true if it matches the contents of the regular expression. If it is the negation of a semantic pattern, it is true if the part is not associated to any semantic annotation that matches the semantic pattern. Finally, if it is the conjunction, disjunction or part comparison of other conditions, it is the conjunction, disjunction or part comparison of their truth values.

For example, a filter can map the semantic annotation alternative(incorrect  $\rightarrow$  A) to the contents A with a given style. When applying this filter, parts that have an incorrect alternative will be substituted by their corresponding alternatives, with the given background.

In general, filter application is done by means of a loop on the pairs that form the filter; for each pair, if the corresponding set is empty, the parts of the document associated to semantic annotations that match its semantic pattern are omitted; otherwise, the corresponding pieces of document are inserted before and after each part of the document whose associated semantic annotation matches the semantic pattern of the pair, and the format of these parts is modified according to the presentation style specified in the second part of the pair.

The DeepTest tool for the design of interactive exercises allows users to edit exercises and add semantic annotations of the type explained above to them. This is done in a seamless process where the user only has to think in the terms used in section 2 above. Moreover, as the exercise is being designed, the underlying document is shown and updated according to a design filter. The designer can change the view of the document by changing the filter. In this way, he can see the original version of the document instead of the one the student will see at the end. Similarly, when the student is solving an interactive exercise, the system updates its semantic annotations. When correcting the exercise, a new correction filter is applied, which is changed depending on the students actions.

## Conclusions and Future Plans

We have presented DeepTest, which allows the creation by teachers of interactive exercises that can be accessed and answered by students. The exercises consist of documents related to virtually any subject, including assertions, references, words or images, where the students are asked to detect the parts of a document that are not correct. Interactive exercises can be created very easily by teachers, who define alternative incorrect statements by just selecting parts of the text and replacing them with their corresponding incorrect versions. Finally, there is an ongoing effort to cover more kinds of exercises based in this preliminary experience, by allowing teachers to specify different semantics to the initial documents.

## Acknowledgements

The DeepTest environment is a product of the work developed within the HyperTest project, funded by the Fundación General de la Universidad Autónoma de Madrid, and projects Ensenada, TIC 2001-0685-C02-01, and Arcadia, TIC2002-01948, from Spain's Plan Nacional de Investigación.

The environment has been designed by the authors of this communication. The development team has been formed by Carlos Alonso, Alberto Andrés, Sergio Jiménez, Javier Martínez, Iván Meléndez, Daniel Mellado, Miguel Pazos, and Almudena San Martín.

The problems of computer programming used in the tests that we have described have been elaborated by Manuel Alfonseca.

## References

- [1] Dwyer, T. M., Fleming, J. Randall, J. E., & Coleman, T. G. (1997): *Teaching Physiology and the World Wide Web: Eletrochemistry and Electrophysiology on the Internet*. American Journal of Physiology (Advances in Physiology Education) 273 (6), S2-S13
- [2] McDonald, R. P. (1999): *Test Theory: A Unified Treatment*. Lawrence Erlbaum Associates, Inc
- [3] Zenisky, A. L., Sireci, S. G. (2002): Technological Innovations in Large-Scale Assesment. In *Applied Measurement in Education*, 15, 4, pp. 337-362
- [4] DeepTest (2004): Access to DeepTest. <http://DeepTest.ii.uam.es>.
- [5] Engeström, Y., Miettinen, R. & Raija-Leenam, P.(1999): Perspectives on Activity Theory. In Ed. Pea, R., Seely Brown, J. & Hawkins, J., *Learning in Doing Social, Cognitive and Computational Perspectives*, Cambridge: Cambridge University Press.
- [6] Engeström, Y.(1987): *Learning by expanding: An activity-theoretical approach*. Helsinki: Orienta-Konsultit.
- [7] Reigeluth, C.M. (1999): *Instructional-Design Theories and Models: A New Paradigm of Instructional Technology* (Volume II). Lawrence Erlbaum Associates, Inc.
- [8] Bloom, B.S., & Krathwohl, D.R. (1984): *Taxonomy of Educational Objectives*. Handbook 1: Cognitive Domain. Addison-Wesley Pub. Co.

- [9] Alfonseca, A. (2005): <http://www.ii.uam.es/~alfonsec/curso4i.htm>. Web de la asignatura Programación Orientada a Objetos 2, Escuela Politécnica Superior, UAM
- [10] IMS (2005): <http://www.IMSProject.org>. Web de IMS Global Learning Consortium
- [11] Robinson, M., & Vorobiev, P. (2003): *Swing*. Manning Pub. Co.