Journal of Universal Computer Science, vol. 14, no. 19 (2008), 3099-3113
submitted: 23/7/08, accepted: 29/10/08, appeared: 1/11/08 © J.UCS

SHARP Online: An Adaptive Hypermedia System Applied to Mathematical Problem Solving

Ana-Belén Gil, Raquel Rodríguez, Francisco J. García-Peñalvo
(Department of Computer Science, Sciences Faculty
University of Salamanca, Salamanca, Spain
abg@usal.es, raquelrh@gmail.com, fgarcia@usal.es)

Ricardo López
(Department of Mathematical Didactics and Experimental Sciences, Faculty of Education
University of Salamanca, Salamanca, Spain
riclop@usal.es)

Abstract: In this article we present the development of a web application called SHARP Online: An Adaptive Hypermedia System Applied to Mathematical Problem Solving. The pedagogical basis of this application is found in the support techniques for heuristic learning in mathematical problem solving developed according to the Schoenfeld model. The adaptivity of this tool is achieved by way of the utilization of an adaptive algorithm which has been developed for it and is described in this article. This algorithm implements mechanisms that make it possible for the user to construct mathematical knowledge adaptively using training methods. This application also provides the teacher with the following complete set of tools for managing the entire process: the inclusion of contents through a collaborative application with support; a shared work space; the adaptivity of the algorithm variables; and the supervision of the students' progress, etc. through specific modules. This application was originally developed for educational contexts in the area of teaching mathematics, and therefore includes a module for editing and visualizing mathematical formulas for a Web environment.

Keywords: Adaptive hypermedia, Interactive learning environments, Heuristic thought, Teaching/learning strategies, User-Computer Interaction
Categories: L.0, L.2.0, L.3.0, G.4

1 Introduction

The myriad of ways that users and computer systems can interact has made it increasingly necessary to develop adaptive systems ([De Bra, 1999], [De Bra, 2002]), systems that are able to adapt their contents, interface and even functionalities to the user who is interacting with them. This adaptivity can be approached automatically, guiding the user to fulfill certain previously-defined objectives, or by consensus with the user, with the latter marking the objectives.

Educational hypermedia was the first application area to see the introduction of adaptive hypermedia techniques before the research extended to other areas such as tourism, e-commerce and news portals. Classical projects such as ELM-ART [Brusilovsky et.al., 1996] and AHA! [De Bra et al., 1998]) are web-based systems used to develop adaptive online courses. But the process of teaching and learning mathematics is extremely complex and there is a significant lack of this domain

model courses. Throughout time human beings have developed diverse methodologies for making this process effective. With the arrival of new technologies, a new research field has opened up through modernistic learning environments and teaching methods that are taking advantage of the enormous potential of these electronic resources [Macias, 2007]. In this context, the adaptivity of a computer system is combined with the use of hypermedia technology, giving the system greater flexibility and interactivity.

Recent proposals for the mathematics curriculum recommend that learning should be organized around problem solving [NCTM, 2000]. It is thus important for students to develop different resources and strategies in order to think about and solve different types of problems. There is a need to create a learning environment in which students have the opportunity to check their knowledge, consolidate their understanding of mathematical contents, and strengthen their problem-solving skills. Learning mathematics is, according to Shoenfeld [Schoenfeld, 1998], a continuous process which is fostered in a problem-solving context, in which students have the opportunity to develop ways of thinking that are consistent with the tasks of the discipline.

The learning mechanism for training processes in interactive systems is based on the human mental process of knowledge acquisition, in which the acquisition and memorization of information correspond to associative processes [Boticario, 2001]. There are many authors who are developing strategies to promote and foster learning mechanisms in the area of mathematics. For example, Schoenfeld ([Schoenfeld, 1985], [Schoenfeld, 1985b]) in particular, has developed a series of important problem-solving strategies, consolidating the construction of a new method in the teaching-learning processes of mathematics: "Learning to think mathematically means (a) developing a mathematical point of view — valuing the processes of mathematization and abstraction and having the predilection to apply them, and (b) developing competence with the tools of the trade, and using those tools in the service of the goal of understanding structure — mathematical sense-making" [Schoenfeld, 1992].

SHARPO, the Spanish acronym for "Adaptive Hypermedia System Applied to Online Problem Solving" is a Web application that integrates the design norms of the adaptive hypermedia system. It incorporates the functionalities of materials management and monitoring of student learning to permit online training. With this tool, students can reinforce their knowledge in the field of mathematics using the Internet; it also permits the implementation of new learning strategies based on e-learning. The basis of the instructional design applied to mathematical problems was established in a previous study [Gómez, 2003], and the development of the tool presented here is framed within the area of mathematics for pre-university level students or students in the first years of university mathematics.

This paper begins with this brief introduction, followed by a description of the architecture and functionality of the application, and of the user support in relevant aspects of the interface. The Web application developed is extensive, but throughout this paper we shall highlight two fundamental aspects of its functionality: the adaptive algorithm which supports the application's engine, which is dealt with in part three; and the module for editing and viewing mathematical notation in a Web environment,

described in part four. We will finish with some conclusions and suggestions for possible future lines of research.

2 Description of the Tool

SHARPO is an application based on learning by training in the area of mathematics education. It was developed with Web architecture (See Figure 1), which supports the tasks by assigning functionality through three roles: teacher, student and administrator.

The teacher is in charge of providing and managing the contents as well as preparing tests and monitoring the students throughout the teaching-learning process. The students improve their practical knowledge by solving math problems directed by the heuristic of each of the areas formalized into mathematical topics. Finally, the role of the administrator is added to that of the teacher, that of the creation, modification and elimination of the set of configurations which assign values to the parameters that the algorithm utilizes for its adaptive management. The administrator also manages the users who have access to the system, registering them or modifying their data.

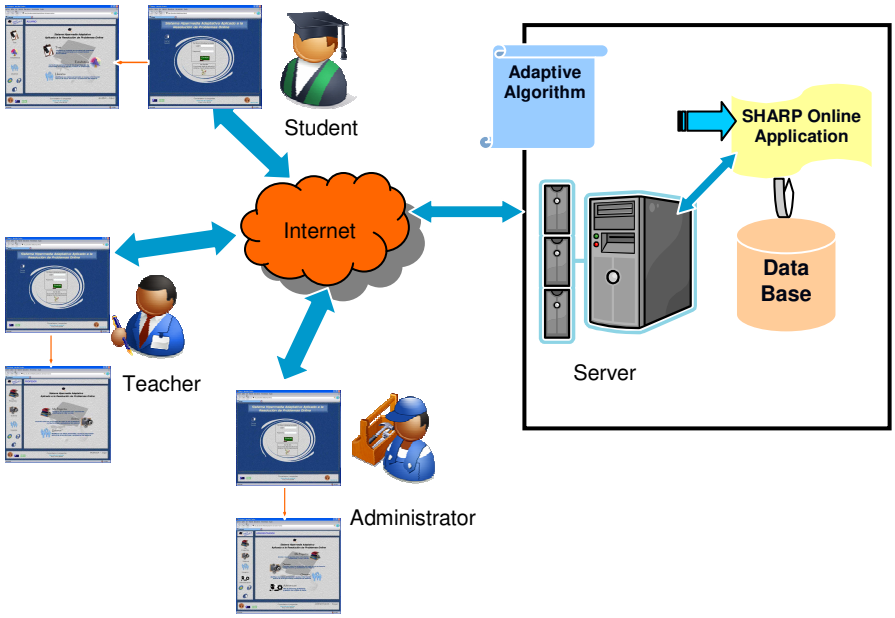

Figure 1: ***Architecture of the system***

The application works with an initial battery of questions whose characteristics define diverse decision-making heuristics developed in the Schoenfeld model. Regarding the didactics of this model, the system is created in four phases with a series of detailed tasks for each one of them in mathematical problem solving: analysis, exploration, execution and checking.

The initial set that we worked with was based on the collection of mathematical tests in the S.A.T. exam (*Scholastic Assessment Test*), used for admission into universities in the United States. These questions can be considered to be a good example of the set of decisive strategies compiled ([Barker, 1991], [Hammond, 1993]). These strategies are part of the variables implemented in the development of an adaptive algorithm, which we will explain in subsequent points and which is designed to select, by way of rules, the questions to be solved and to coordinate the hypermedia system and manage the navigability between the different nodes of hyperspace in order to increase, by way of training, the heuristic strategies in mathematical problem solving.

An adaptive algorithm with the selection of questions makes it possible to parameterize and regulate the hypermedia system developed according to the student's situation at each moment, in addition to managing the navigability between the different nodes of hyperspace that constitute the questions in the database.

2.1 General Description of the Interface

The design of the interface (see Figure 2) is based on aspects of usability ([Preece, 1993], [Nielsen, 2000]). The icons chosen represent an extended metaphor in the field of education, symbolizing typical elements such as a book, a calculator, a bookshelf, so that the user can gain access to the different functionalities in an intuitive fashion. Each button and item associates with an action with the different events of the pointer, so that if the mouse passes over a clickable area, this modifies its appearance because it has an associated action. Likewise, the interface provides communication with the user at all times by way of panels for information, alert or contextual assistance as well as events with responses provoked by actions, errors, verifications, etc.

The interface is divided into 5 highly-differentiated areas (see Figure 2):

- Location / Breadcrumbs Bar: Located at the top, it contains an indicator with a breadcrumb trail to avoid user disorientation in the application and gives options for exiting the system.
- Options Panel: Located on the left part of the interface, from which all the application modules can be accessed. Each of the three types of users (Administrator, Student and Teacher) will have some different options. At the bottom, this panel includes some *Common Options* for all the users, which makes it possible to access the contextual help and navigation (previous screen, return to the main menu, etc.).
- Content: This is the central and main interface, where the content is shown on the screen that the user works on. It varies depending on the user and the option selected.
- System Information: It is located in the lower part of the content panel; data is displayed about the active users at the moment, as well as system information.

2.2 Role of the Student

Using the options panel, the student can access each one of the options offered by the system.

- **Test**: In this window, the student can interact with the system, visualizing and responding to the test configured by the adaptive algorithm (see Figure 3).
  In the location / breadcrumbs bar the student will see information about the level that the question corresponds to, the topic it is framed in, or the time remaining to finish the block. In the *options panel,* a graph is shown with information about the number of responses given to questions which have been asked at this level. In *the content section*, the question to be answered is displayed along with a series of options to respond to.
- **Statistics**: The student can consult data of personal use for training. This information includes graphs generated dynamically for each consultation, and the student can interact with them by way of pull-down lists.
- **Users**: Here, information can be accessed about other users registered in the system, and personal data can be modied.

Figure 2: *Structure of the Interface*

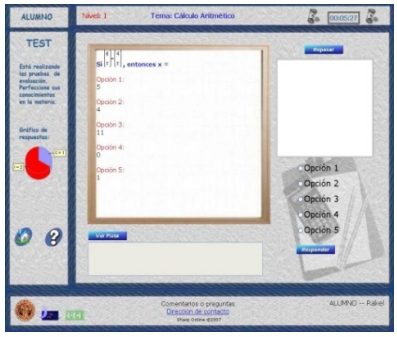

Figure 3: Test Interface

2.3 Role of the Teacher/Administrator

This user is in charge of managing, creating, modifying or eliminating questions, in addition to having access to other characteristics of the system such as the overall statistics, the configurations, the questions of other users, and everyone else's personal information.

- **My Questions**: The user will be able to create, modify or eliminate his or her questions using three options: assistant, advanced and consultation.
  - Assistant Mode. In this mode, the question is created through a series of screens, in each of which only a single piece of information is asked for about the question being inserted. It is possible to navigate from one to another thanks to the buttons with the right and left arrows located in the lower part of the content area (see Figure 4)
  - Advanced Mode. In this mode, all the necessary fields are gathered for the insertion of the question, on the same screen with text fields that permit editing and groups of buttons with the options of grading the responses and selecting the level, as well as a pull-down list for selecting the topic.
  - Consultation. Where the user will be able to consult his or her questions, modify them or eliminate them.

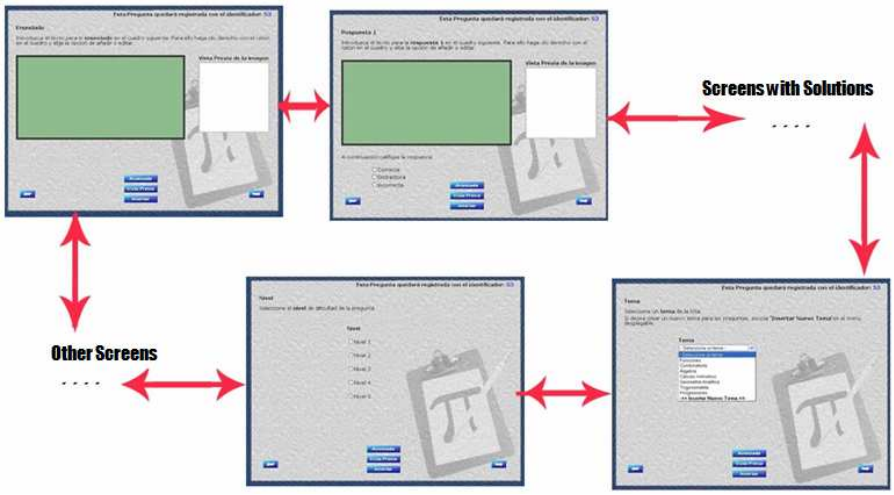

Figure 4: Use Option in Assistant Mode

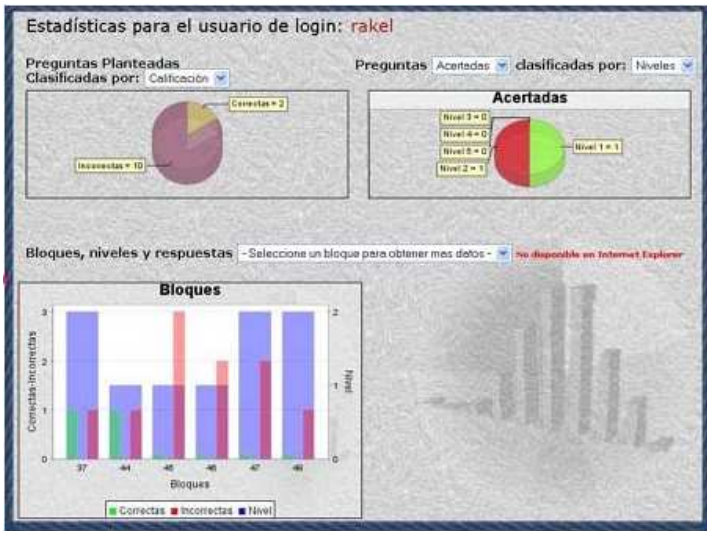

Figure 5: S(tatistics at system Role

- **System**: The administrator will have access to all the characteristics, where he or she will see the questions of the whole system, the use *statistics* (see figure 5) and will have the configuration, where there will be access to the parameters which the algorithm utilizes. The teacher can consult them and the administrator, in addition to that, can modify them, create new ones, update them and eliminate them.

- **User:** With this option, the rest of the users can consult data and modify their personal information. If it is an administrator, he or she can also modify the information of the rest of the users.
- **Administration** (only administrators): In this option, the administrator can register teachers or students and can allow students who have finished the tests to interact with the system again by reactivating their accounts.

2.4 SHARPO Architecture

In order for the tool implemented to function in an online mode, it is hosted in a server that interprets dynamic content. The system is in constant communication with a database containing all the information that the algorithm needs (See Figure 1). Through his or her navigator, the user accesses the application, identifying him or herself and interacting according to his or her working role.

The technology used to develop the system is based on the platform J2EE. As a prerequisite, we attempted to use only software that was available for free. Java was used as a programming language along with JSP on the side of the server in order to implement both the algorithm and the rest of the application, in addition to SQL for the database and DHTML, XML, and JavaScript on the client side. A JSP container was worked with for the management of the dynamic contents. The server used was Jakarta-Tomcat. The Web server used was Apache, and the management system for the database was MySQL.

3 SHARPO's Adaptive Algorithm

The adaptation module implemented by SHARPO contains the description of the adaptive functionality. It constitutes the intelligent part of the system, in charge of adjusting it to the student's knowledge according to a set of logical rules that have been defined and whose parameters can be configured by the teacher.

The adaptive model was developed to support the learning process and the evolution of the student's knowledge on a hierarchy of three levels, so that the control for passing from one level to another will be determined by a filter controlled by logical rules. These rules activate different states of user support according to the numerical projection carried out by the scale of the algorithm. If the responses are not correct, the system applies a correction module that allows the student to develop other ways of fulfilling the assessment criteria. If the pre-established objectives are still not met, the corresponding state is activated, a stoppage event. Likewise, if it is proven that the learning of the contents has been achieved in a satisfactory fashion, according to pre-established criteria, the process will be concluded successfully.

The algorithm is structured in five phases (see Figure 6) which we describe below:

- Start-up of the algorithm when a student begins the test
- Selection of the block: Obtain block>Calculate the number of questions that make up the block>Initiate the question timer for the block
- Question selection: Topic selection> Question selection > Obtain question data
- Correct the student's response

    - Process response >Show the grade to the student>Assess the block
- Change blocks

The algorithm functions by adjusting the variables which are configured by way of parameters and which determine the choice of question, the number of questions that are formulated, criteria for the assessment of levels, of correction, response time, etc. These configuration parameters are grouped into three criteria in the following way:

a. *Number of questions*: Percentage of questions per block and minimum questions per block.
b. *Criteria for assessment and for changing level*: percentage of correct responses necessary for going up a level, percentage of correct responses for going down a level; combined percentage of correct responses for going up a level, combined percentage of distractor responses for going up a level; weighting of consultation of hints and maximum of repetitions.
c. *Times:* average estimated time per question, threshold for warning time and threshold time for going down a level.

For a correct configuration of these parameters, it is necessary to know the concepts (Questions, Responses, Block, Assessment, Times, Topics and Hints) that the algorithm is working with, which are described below.

Questions. Each question has 5 possible responses and each one of them has an assessment parameter associated with it: correct, incorrect and distractor. The grade of each response is determined by the teacher when he or she introduces the questions. The distractor responses are responses which, although they are not correct, are 'less incorrect'. That is to say, their content can lead the student to a response error.

Responses. Each response is graded with three different states. These states correspond to the grades that were associated with a given question when it was created.

Block: the concept of a block is defined along with the concept of a response. A strong relationship exists between both elements and affects the functioning of the algorithm as explained below:

- A block corresponds to a set of responses given and is the minimum unit that will be considered when assessing the student.
- In order to grade a block, the grades obtained by the student in all the responses given in that block are considered.
- A block can be divided into various sessions; it is a unit in continuous development. The system knows how many responses are necessary in order to proceed to the conclusion of the block, but not what questions will be formulated for it. The number of responses that make up a block can vary before it is finished due to special situations that can occur in the database and to particular configurations defined by the algorithm.

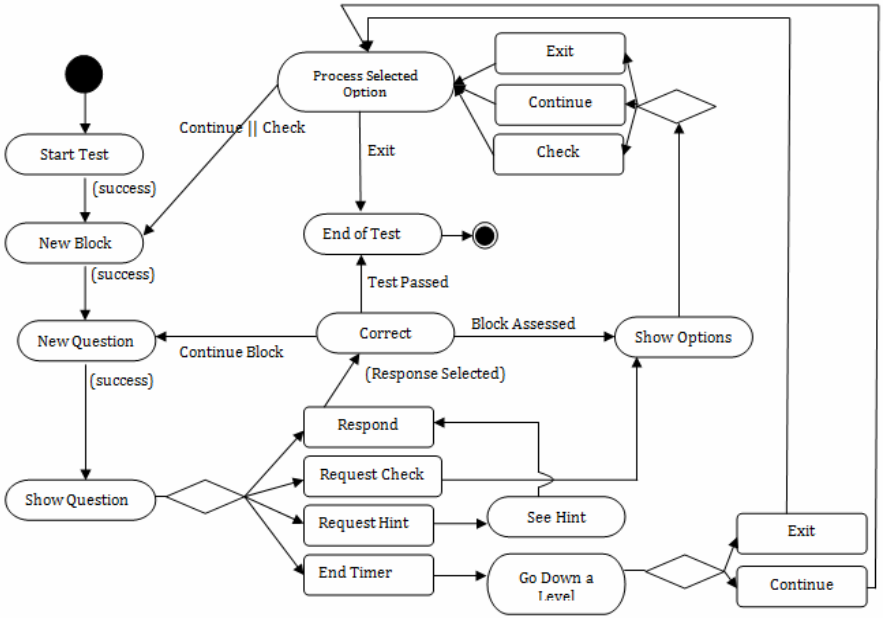

Figure 6: Adaptive Algorithm

Assessment. For the assessment of the responses given, the questions to be formulated are divided into blocks; the number of questions that constitute a block are determined by a configuration parameter variable, "percentage of questions per block". After responding to the determined number of questions, the next step is to proceed to the analysis of the block and check if the student should go up one level of difficulty, go down, or stay on the same level. To do this, the threshold parameters for going up or down are used.

To analyze a block, the algorithm utilizes the following configurable parameters:

- *Minimum threshold for going up a level*: determined by the percentage of correct responses given for going up.
- *Maximum threshold for going down a level:* determined by the percentage of correct responses given for going down.
- *Special threshold for going up a level:* determined by the percentages of combined correct and distractor questions for going up a level.

These parameters function as 'cut-off marks' divided between 0 and 100. Answering all the questions in the block correctly gives the value of 100, whereas answering all of them incorrectly is a 0. The percentage of questions answered correctly in relation to the total of correct responses is compared with these thresholds. If the minimum threshold for going up a level is surpassed, the analysis of the block makes the level of difficulty increase. Should the maximum threshold for going down a level not be surpassed, the student must drop down to the next level. And if the assessment falls between these two extremes, the level is repeated.

Times. During the configuration, the parameter "average estimated time per question" is defined. Based on this, the time for the block is obtained.

Topics. The individual selection of each question is adjusted to the topics of the database and to the percentages defined by the teacher in each test. A topic with 15% associated with it indicates that the adaptive algorithm will attempt to have 15% of the questions in a block belong to that topic.

Hints. For each question, the student can consult a small hint. This consultation has a negative effect the result of the assessment of the block. The parameter of penalization for consulting the hint is also configurable, "weighting the penalization for each hint consulted".

The algorithm is codified in Java in the application by way of a set of rules which lend support to the process described. Below, we present an example in pseudocode of the rule utilized for the assessment of a block:

```
start evaluateBlock
  questionsInBlock = countResponseesInBlock()
  if questionsInBlock < numberOfQuestionsB then
    selectQuestions()
  otherwise
    numHints = countHintsConsulted()
numQuests = calculateNumQuests (numHints)
    numQuestC = countQuestionsCorrect ()
    numQuestI = countQuestionsIncorrect ()
    numQuestD = countQuestionsDistractor()
    if numQuestC >= thresholdMinimumGoUpLevel then
      newLevel = GoUpLevel ()
        otherwise if numQuestI <= MaxThresholdGoDownLevel
then
          newLevel=GoDownLevel ()
          changeLevel = true
 otherwise  if  numQuestC  >=  MinThresholdCCorrect  &&
numQuestD >= MinThresholdCDistractors  then
    newLevel=GoUpSpcialLevel ()
    otherwise
       Repeat Level
   end
   end
   updateStateOfBlock()
   if changeLevel then
      updateLevel (newLevel)
      if newLevel = 6 then
    warnEndOfTest ()
    exit()
      end
    otherwise
      changeBlock()
   end
end assessBlock
```

4 Module for Editing and Visualizing the Mathematical Language

The SHARPO application in teacher mode makes it possible to generate and edit questions. This task is supported by the editor of the tool which includes a module for working with mathematical notation in the Web environment.

After analyzing the problems of editing/visualizing mathematical language in the Web and the set of editors and converters which exist, the option ASCIIMathML [Jipsen, 2007] was chosen. Not only is this one of the few free tools available, but it also makes it possible to carry out the translation from the client side, freeing up the server from this task. ASCIIMathML is a JavaScript that carries out the conversion of ASCII to MathML. The translation is carried out from the client side and the process is conducted by going over all the content of the page on which the text is found using the model DOM of JavaScript.

Given that MathML is a standard, it can be expected that any navigator should be able to interpret the tags that ASCIIMathML generates, but this is not the case. The relative newness of MathML, the need to have an interpreter for Javascript, and the way in which the navigators evolve means that, as of today, it is only supported by a few navigators under certain conditions (*Firefox* with the fonts appropriate for Linux, MacOS and Windows; *Mozilla* (v. 1.3a) and Camino in Mac OS X and *Internet Explorer* (v. 6.x) with the corresponding *plugin* (MathPlayer) in Windows).

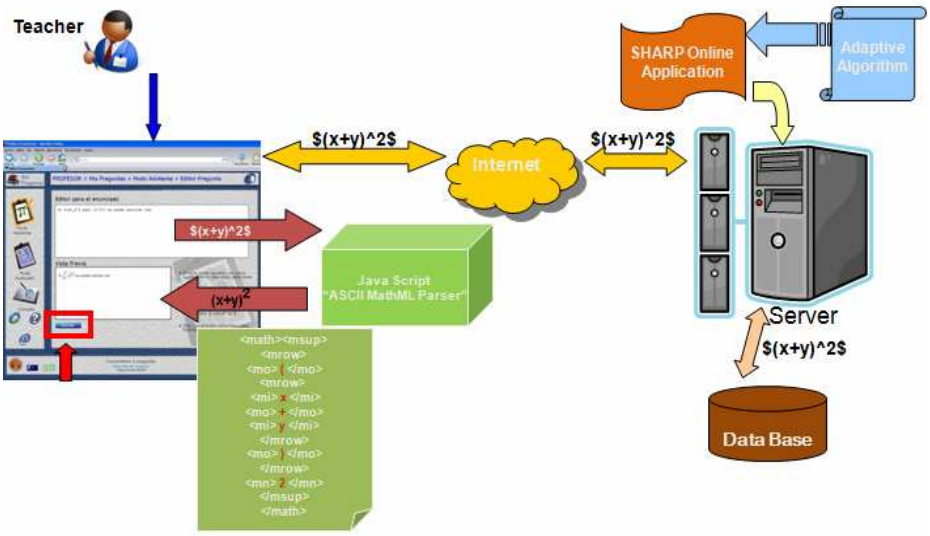

Figure 7: Levels of access and visualization for mathematical notation

4.1 Editing the Mathematical Equations

ASCIIMathML was the starting point, having been adapted and integrated to the needs of the application defined in the file JavaScript *ASCIIMathML.js*.

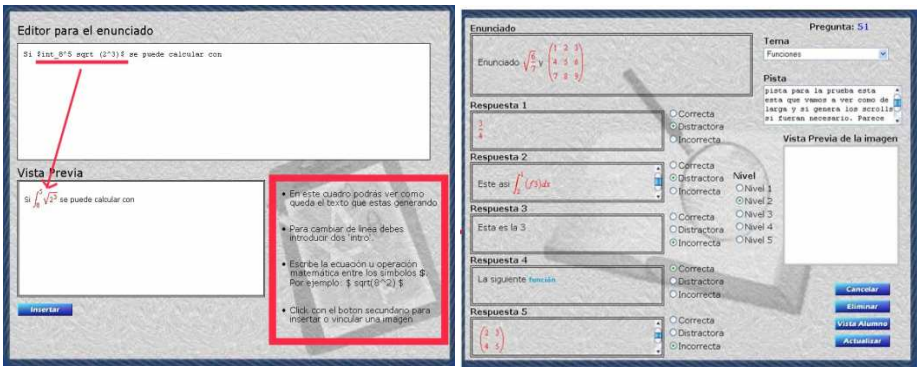

Figure 8: Mathematical Formula Editor

The editor of the application recognizes as mathematical notation elements which are located between two $ symbols (for example, x^2 = x^2). It has various areas in the editing module, see Figure 8. The area for editing is where the text of the question is written along with formulas, images, and other elements. These are displayed in the lower box, which is called the visualization area. The editing module also offers contextual assistance.

4.2 Parser

ASCIIMathML is a JavaScript and therefore it is necessary to have a navigator which is able to interpret this language (and which is activated). This JavaScript carries out the translation (from pseudo-LaTeX to MathML) of formulas embedded in HTML documents. The translation is carried out making use of the function *"translate()"* which this JavaScript implements. It goes over the page beginning with the element in which the function has been called applying the translation function recursively to all of the sub-elements.

For each element to be dealt with, it looks for the character that delimits the mathematical formulas and substitutes each of the entities with its counterpart in MathML. In this process, the definition of the grammar is applied that ASCIIMathML uses to parse the expressions (See Table 1).

c ::= [A-z] | numbers | Greek letters | other symbols for constants
u ::= 'sqrt' | 'text' | 'bb' | other unary symbols for font commands
b ::= 'frac' | 'root' | 'stackrel' binary symbols l ::= (| [| { | (: | {: keys to the left
r ::=) |] | } | :) | :} keys to the right S ::= c | lEr | uS | bSS | "any simple expressions
E ::= SE | S/S |S_S | S^S | S_S^S expression (fraction, sub-, super-, subsuperscript)

Table 1: Grammar utilized for the ASCIIMathML parser

By way of the translation rules, each terminal symbol is translated to its corresponding node in MathML. The constants are converted into their *Unicode* symbol. The remaining expressions are converted as shown in Table 2:

lSr	<mrow>lSr</mrow>
sqrt S	<msqrt>S'</msqrt>
text S	<mtext>S'</mtext>
"any"	<mtext>any</mtext>
frac S1 S2	<mfrac>S1' S2'</mfrac>
root S1 S2	<mroot>S2' S1'</mroot>
stackrel S1 S2	<mover>S2' S1'</mover>
S1/S2	<mfrac>S1' S2'</mfrac>
S1_S2	<msub>S1 S2'</msub>
S1^S2	<msup>S1 S2'</msup>
S1_S2^S3	<subsup>S1 S2' S3'</msubsup> or <munderover>S1 S2' S3'</munderover>

Table 2: Translation Rules

4.3 Sending and Storing Data

Once the question has been edited by the teacher who generates the content from his or her navigator, the storage process for saving the question in the system's database is that indicated in Figure 7.

Each formula generated is sent through the Web in the form of the ASCII code in which it was written. The application hosted in the server will mark this formula as a "mathematical element" and will associate it with the option of the question for which it was created.

All of the information will be stored in the database with the specific ASCII code. In this way, when the visualization of the student's test is generated, the system will retrieve the question data for it to be loaded in the client's navigator, and the parser which is implemented will display this data in visualization mode.

5 Conclusions

In this article, we have described a system for supporting online teaching which makes it possible for students to train themselves in mathematical problem solving. The modular architecture of the application includes functionalities for managing teaching and the creation of contents. The management of the educational process is regulated by way of an adaptive algorithm which incorporates parameters with which the training sessions can be programmed and monitored. The system has the ability to work through the Internet, which facilitates its use and incorporation as a complement to other educational technologies.

The functioning of the algorithm with regard to management of the hyperspace is not visible to the student, but the algorithm is responsible for adapting the content of the problems to but the algorithm is responsible for adapting the content of the problems to each student's own mathematical shortcomings each student's own mathematical shortcomings, offering alternative paths in each case. The design of the algorithm allows for adaptation not only to the student's shortcomings, but also to the state of the database of the questions, thus lending flexibility to the incremental and cooperative development on the part of the teachers in charge of generating the batteries of questions.

This study makes it possible to analyze the advantages that hypermedia technology provides for the development of applications which implement learning processes which are favourable to heuristic thought. A personal "training tool" has been constructed for solving problems in the Web environment, SHARPO, with adaptive techniques in the teaching of mathematics, including didactic techniques applied to the field, such as the Shoenfeld model.

A Web application with the functionalities presented (and due to its modular design) makes it possible to continue incrementing the development of diverse aspects, essentially along three axes of expansion: the first in regard to its adaptation to other types of educational contexts; the second action will be to expand the statistics module, adding more functionality such as searches which are intelligent and more flexible and adapted to other educational criteria; and finally, the inclusion of the internationalization of the tool.

Acknowledgements

We would like to thank to the group GRIAL (Research Group in Interaction and eLearning) of the University of Salamanca for their contributions and ideas for the development of this work.

This work was partly financed by Ministry of Education and Science as well as FEDER Keops project (TSI2005-00960).

References

[Barker, 1991] Barker, P.: "Interactive Electronic Books". Interactive Multimedia 2, (pp. 11-28).1991

[Boticario, 2001] Boticario, J. G. y Gaudioso, E.: "Aprender y Formar en Internet" Thomson

[Boticario, 2001]Learning Paraninfo, Madrid, Spain, 2001.

[Brusilovsky et al., 1996] Brusilovsky, P., Schwarz, E., and Weber, G. 'ELM-ART: An intelligent tutoring system on World Wide Web', in C. Frasson , G. Gauthier and A. Lesgold (eds.), Third International Conference on Intelligent Tutoring Systems, ITS-96 (Lecture Notes in Computer Science, Vol. 1086), Berlin: Springer Verlag, pp. 261-269.1996

[De Bra et al., 1998] De Bra, P. and Calvi, L., 'AHA: a Generic Adaptive Hypermedia System', *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia*, Hypertext 1998, Pittsburgh, USA, 1998.

[De Bra, 1999] Paul De Bra, Peter Brusilovsky, and Geert-Jan Houben. "Adaptive hypermedia: from systems to framework" in ACM Computing Surveys 31(4), December 1999.

[De Bra, 2002] Paul M. de Bra, Peter Brusilovsky, Ricardo Conejo. Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002, Malaga, Spain, May 2002.

[Gómez, 2003] Gómez A., Sosa F., Gil A., López R., García F.:"SHARP: Sistema Hipermedia Adaptativo Aplicado a la Resolución de problemas". En las actas de Interacción 2003, ISBN: 84-932887-4-8. Vigo, June 11 to 13, 2003.

[Gómez, 2003] Gómez., M. A. "Usabilidad, Interfaces de Usuario e Inteligencia Artificial. Diseño e Implementación de un Interfaz Adaptativa para una Web." September. 2003.

[Hammond, 1993] Hammond, N.: "Learning with Hypertext: Problems, Principles and Prospects." Eds. Mcknight, C., Dillon, A. Y Richarson, J. Hypertext a psychological perspective. Ellis Horwood, New York. 1993.

[Jipsen, 2007] Jipsen, Peter . Translating ASCII math notation to MathML and graphics. Available in web [http://www1.chapman.edu/~jipsen/mathml/asciimath.html], 2007

[Macias, 2007] Macias Ferrer, D.: "Las nuevas tecnologías y el aprendizaje de las matemáticas". Revista Iberoamericana de Educación, Número 42/4, 2007

[Nielsen, 2000] Nielsen, Jacob "Usabilidad : diseño de sitios Web" Prentice-Hall, 2000

[NTCM, 2000] National Council of Teachers of Mathematics. (2000). Principles and Standards for School Mathematics. Reston Va.: National Council of Teachers of Mathematics.

[Preece, 1993] Preece, J.: "A guide to usability: human factors in computing". Addison-Wesley, 1993

[Schoenfeld, 1985] Schoenfeld, A.: "A. Mathematical problem solving", Academic Press, New York, 1985

[Schoenfeld, 1985b] Schoenfeld, A.: "Metacognitive and epistemological issues in mathematical understanding", in Teaching and learning mathematical problem solving: Multiple research perspectives (pp. 361-380). Hillsdale, New Jersey, 1985.

[Schoenfeld, 1992] Schoenfeld, A.H. Learning to think mathematically: Problem Solving, Metacognition and Sense-Making in Mathematics. In Grouws, D. (Ed.) Handbook for Research on Mathematics Teaching and Learning. New York: Macmillan. 1992

[Schoenfeld, 1998] Schoenfeld, A., H. Reflections On a Course In Mathematical Problem Solving. Research in Collegiate Mathematics Education III., pp. 81-113, 1998.