This document is published in:


DOI: http://dx.doi.org/10.1002/cae.20240

© 2009 Wiley
A software player for providing hints in problem-based learning according to a new specification

Pedro J. Muñoz Merino*, Carlos Delgado Kloos

Department of Telematic Engineering, Carlos III University of Madrid
Av. Universidad, 30. E-28911 Leganés (Madrid) Spain
pedmume@it.uc3m.es, Tel: (+34) 91-624-8801, Fax: (+34) 91-624-8749

Abstract

The provision of hints during problem solving has been a successful strategy in the learning process. There exist several computer systems that provide hints to students during problem solving, covering some specific issues of hinting. This paper presents a novel software player module for providing hints in problem-based learning. We have implemented it into the XTutor Intelligent Tutoring System using its XDOC extension mechanism and the Python programming language. This player includes some of the functionalities that are present in different state of the art systems, and also other new relevant functionalities based on our own ideas and teaching experience. The paper explains each feature for providing hints and it also gives a pedagogical justification or explanation. We have created an XML binding, so any combination of the model hints functionalities can be expressed as an XML instance, enabling interoperability and reusability. The implemented player tool together with the XTutor server-side XDOC processor can interpret and run XML files according to this new defined hints specification. Finally, the paper presents several running examples of use of the tool, the subjects where it is in use, and results that lead to the conclusion of the positive impact of this hints tool in the learning process based on quantitative and qualitative analysis.

Keywords: assessments, learning systems; hints; teaching strategies; software tool

Acknowledgements

Work partially funded by Programa Nacional de Tecnologías de la Sociedad de la Información, MEC-CICYT project MOSAIC-LEARNING TSI2005-08225-C07-01 and 02

* Corresponding author
1. Introduction

Nowadays, learning through computers is performing a standardization process and properly tools that are compliant with such standards are need for the successful adoption. An overview about distance learning interoperability and standards is shown for example in [1]. There exist multiple specifications generated by different organizations. Some examples are: SCORM [2] for content packaging and sequencing, IMS-QTIv2 [3] for test and assessments or IMS-LD [4] for describing pedagogical models.

Several experiments have shown that the provision of hints during problem solving has been a successful strategy in the learning process (for example a recent one in [5]). It follows the philosophy of student learning during the process, not giving directly the solution to the student, but immersing the student in a process of discovery [6]. The IMS QTI v2 specification [3] allows providing hints through its adaptive items. Nevertheless, it only explains the most basic form of hint, delivering a string after a candidate request. However, it does not set a taxonomy of the different ways of providing hints, specific scoring methods taken into account hints, etc. In addition, since there are no specific XML tags in IMS QTI v2 for hints, then hint material representation may lead to big XML files, using condition tags over variables within the “responseProcessing” tag. We think it can be a very interesting issue to create a new hints specification that explains the different cases for providing hints and uses specific XML tags for hints reducing the size of hints XML material.

In this work, a reach theory model of hints for problem-based learning is proposed and a software player tool that implements all of the model features is presented. The player module has been implemented into the XTutor (http://xtutor.org/) Intelligent Tutoring System using its XDOC extension mechanism and it has been programmed using Python. This player together with the XTutor server-side XDOC processor can take generated XML files according to the proposed hints specification, and it can interpret and run
them according to its semantics and XML binding data model. Moreover, an authoring tool in Apache Tomcat using Java has been developed in a Master Thesis [7]. This tool is capable of generating XML code according to the defined hints specification. Therefore, students can take assessments with hints in the XTutor system, which have been previously generated by teachers with the authoring tool.

The novelty of this hints player software relies on two aspects: 1) It includes a wide spectrum of possibilities for providing hints: It includes the union of different state of the art hint system features, and also new functionalities based on our own ideas and teaching experience. 2) It is the first hint system that can interpret and run XML files that describe hint material according to this new specification about hints that distinguishes between different hinting possibilities and enables specific hint tags.

The remainder of this paper is organized as follows. In Section 2, there is an overview of related work about hints systems and the relation of our work with respect to state of the art hints systems is explained. Section 3 explains the features of our software player module, which are the same that the requirements of our new proposed hints model for problem-based learning. In addition, there is a respective justification for each aspect based on pedagogical ideas or other reasons. In Section 4, there is an outline of the general system framework including the implemented tools. Section 5 shows some examples running with the Xtutor hints player that illustrates the different aspects of the defined model. Section 6 presents the courses where the tool is used, while Section 7 presents some results of use of the tool in classroom and positive conclusions are obtained from quantitative and qualitative analysis. Finally, Section 8 is devoted to our conclusions.

2. Related Work about hints systems

There are several educational hints systems. The common idea under these systems is to present some type of help to a student related to some topic. The hints functionalities that can support these systems vary. M.
Hough and T. Marlin [8] presented a Web-based interactive machine where students can request for hints and they can receive feedback. J. Zhou et al. [9] showed a hint system inside the CIRCSIM-Tutor Intelligent Tutoring System that provides a hint or the solution but only when a student submits an incorrect answer. A. Gertner, C. Conati, and K. VanLehn [10] and K. VanLehn, et al. [11] used the Andes tutoring system that provides hints as a response to a student request for help, being able to select several of them. Hume, G., et al. [12] generated 315 hints in their tutors that were triggered due to student errors, one modality of hints consisted in delivering a sequence of step by step logical related hints. M. Y. Feng, N. T. Heffernan and K.R. Koedinger [13] presented a study over the ASSISTment tutoring system trying to predict the score of a specific test based on different parameters of ASSISTment. That system delivers different test items to the students and let students ask for help several times when they answered wrong, obtaining different hints for each request. That system also allows several attempts for an item. M. A. Ringenberg and K. VanLehn [14] also used two different versions of Andes (one with graded hints and another with worked-out examples). The presented hints system has feedback for correct and incorrect answers and different hints hierarchy levels. The different hints hierarchy levels available for student requesting can mean a problem of hint abuse [15]. Finally, the SIETTE assessment Web based system [16] is also a system that let tests with hints when there is an incorrect answer. Our proposed framework allows most of these possible types of functionalities available in these state of the art hints systems, but also provides new different functionalities.

A different issue is the strategy to elaborate the hints content with a set of given hints system functionalities. This depends on the hints creators design criterions. The same hints system can support different hints contents. For example, J. Zhou et al. [9] determined useful hinting strategies from human tutors such as giving an intermediate casual link, refer to an anatomy object, pointing out the laws of physics involved, giving evoking terms and synonyms or linguistic hints. Other example is Hume et al. [12] that exposed how hints may directly transmit information or point to previous student knowledge. In
our proposal, our model provides a framework where different hints content creation strategies can be achieved (such as for example all the ones in the papers cited here).

Several of the hint systems provide a personalization of hints depending on the student’s needs and profiles. J. Zhou et al. [9] improved the CIRCSIM-Tutor allowing hints to a problem depending on the incorrect answer to that problem. A. Gertner et al. [10] showed how the Andes system gives hints depending on the user profile based on a Bayesian Network Student model. S. Suebnukarn and P. Haddawy [17] also used Bayesian Networks to model students in order to generate personalized hints for problem-based learning, but they used a different system called COMET where collaborative group interaction is allowed. Moreover, in that work hints given by the system were evaluated by comparing them with those of experienced human tutors. K. N. Martin and I. Arroyo [18] presented AgentX, a tutor that divides students in clusters depending on their level and provides specific hints to students depending on the cluster they belong to. Finally, the CMU Algebra tutor was used to provide personalized hints [19].

Our proposal provides a framework to allow different users receive different hints according to their profile. Nevertheless, our proposal does not model the different path contents depending on conditions, for such thing another specification such as IMS LD [4] may be considered. For achieving personalization, the proposed hints specification could be combined with other different specifications or techniques.

3. Software player features according to hints model requirements

Next, there is a list of the software player features that are according to the hints model requirements. The combination of all the features is allowed. The correspondent justification for each feature is provided:

- **Problems can be presented to students as part of assessments.** This is according to a problem-based learning methodology. This philosophy is presented in multiple papers, such as in [20] or [21]. The importance of assessments is argued in multiple studies, such as in [22] or [23].
• **Problems can be of the following types:** Fill In the Blank, Multiple Choice, Multiple Response and Short Answer. These types are included in IMS QTIv2 [3]. Other types of items that are part of IMS QTIv2 could be included. The reason why these other items have not been included is because our purpose is to illustrate the different hints features and the different types of question items is an orthogonal issue. Therefore, we have selected what we feel are the more important ones for simplicity. Nevertheless, the tool could be extended easily for including other types of items.

• **Each problem can have an indefinite number of hints related to it.** These hints can be requested by students pressing a button. Enabling a student to request for hints is a technique for engaging student active learning [24] where the student decides when it needs help.

• **The hints related to a problem can be available at the beginning or they can become available only as a result of an incorrect answer to the problem.** Also, a combination of both methodologies can be done. This is to let the most flexibility for the hint creators covering all the possible combinations regarded to this issue. Hints as a result of an incorrect answer to a problem let the system guide the students and not to let hint abuse [15].

• **The atomic elements of hints can be text or other problems.** By this way, there are two kinds of problems: a) Global. If it is not part of a hint b) Hint. If it is part of a hint. Due to the philosophy of problem-based learning [20], [21], one or several problems can compound a hint. Therefore, the own hint is composed of hint problems and students can follow the same problem-based learning methodology when he/she is provided with hints.

• **An indefinite level of hints hierarchy can be achieved.** The hinting content can be designed in such a way that a student can request or receive a first level set of hints for a problem, but each hint can be a combination of hint problems (that all together conform the hint), for which the student can request or receive another set of hints of second level for which each hint can be another different combination of hint problems, for which the student can request or receive another set of hints of third level and so on. Furthermore, the proposed model must support that in whichever hierarchy level, hints can be available at the beginning or they can be made available only if student’s answer is incorrect. This
philosophy of hints hierarchy is in order to enable students learn step by step through a nested hints provision, giving help to the students not only for the initial problem but also for whichever hint problem that compose a hint. This is to provide a framework to let a student learn the previous concepts that are required for learning some another concept and allowing it in an indefinite level nesting according to the meaningful learning theory [25].

- **Text Feedback is allowed in whichever problem.** Delivering of feedback to students is allowed as a response to different situations: correct answer, incorrect answer or syntax error. This is according to several studies such as [26] that argue about the benefits of feedback.

- **Hint Sequencing is allowed.** A sequence hint can be defined as a hint that is composed of a set of ordered related hint problems associated to the same hint topic. This is to allow flexibility.

- **Hint Grouping is allowed.** Hints can be grouped into different categories in a hierarchal way (but note that this concept is different from the previous hints hierarchy feature), and the student can see the grouping. This is in order to be able to classify and categorize the different hints.

- **Students can select a maximum number of hints from a list based on meta-information.** The model let the system provide a number of N independent hints meta-information for which the student can select to visualize a maximum number of K hints, being K<=N. The model let the student select the hints based on some previous informative text that gives meta-information about the hint (for example the type of hint problems, the score penalization for viewing the hint, a cue about the hint, etc.). This is according to active learning, in which the student is in the centre of the teaching/learning process and he/she can take decisions about what hints to visualize based on some previous information. This is also to enable students to develop the meta-cognitive skill of decision-making based on some information. He/She will not be provided with all the information but only with some cues about the hints, so the student cannot take all the information and he/she should make a decision. Finally, this way can be considered as a way to prevent student hint abuse.

- **Definition of different hints depending on the student aim is allowed.** The model defines a framework to allow delivering different hints depending on the student aim. As an example, students
could be divided according to the Bloom taxonomy [27] and different hints would be provided depending on the category a student belongs to. This is according to a methodology of delivering students with specific personalized materials adapted to their particular needs.

- **Hints for assessments and sections can be provided.** A set of global problems can form a section. All sections must form an assessment. The model allows providing a set of hints related to a section and also to an assessment. The section and assessment hints can contain the same functionalities considered for problems hints. This is to have more potentiality in the model and to allow hints that are common to a set of problems that compounds a section or common to the global assessment.

- **Let a score system that takes into account the hints.** The proposed model allows marking the assessments, sections and problems. It let students know their knowledge and progress in some topic. In addition, the score for each element can be given or not to the students. The model allows flexibility in order to let teachers and course designers select their desired scoring method. The assessment score will be a weighted score of the related sections. A section score will be a weighted score of the global problems that compound that section. Finally, the score model allows for each problem (being global or part of a hint) the following: (some important problem score configuration parameters are presented with a special notation for a posterior problem score calculation).
  
  - **Score normalization.** MS (Maximum Score) represents the maximum possible absolute score for a problem, while NS (Normalization Score) is the maximum normalized score.
  
  - **Reward correct answer to the problem.** CS (Correct Score) represents the points to add to the score for solving the problem correctly.

  - **Penalize for viewing hints.** The penalization can be different depending on the type of hint. This is to avoid hint abuse, to be fair in the evaluation of the student knowledge (if a student answers correct without seeing a hint then a better score must be provided) and to make a student takes decisions. VP (View Penalty) is the penalization in points to the immediately higher hierarchy problem score for viewing this hint problem. This has no sense in global problems and is only applicable for hint problems that have a parent problem with score.
- **Penalize for incorrect attempt for a problem.** The model provides the possibility of penalization for each incorrect attempt for a problem until a maximum of attempt tries. AP (Attempt Penalty) is the penalization in points to the score, for a failure attempt to solve the problem. MAP (Maximum Attempt Penalty) is the maximum penalization in points to the score for attempting a problem.

- **Penalize incorrect hint resolution and reward correct hint resolution.** The scored problem has the possibility of score penalization if the student obtains a bad scoring solving the hints related to the problem or score reward if the student obtains a good scoring solving their related hints. Moreover, the penalization or bonus can be different depending on the specific hint. As an indefinite hierarchy level is possible, then a bad or good scoring in a hint problem of a level i, can have effect in the scoring of the global problem. The model allows setting a maximum penalization (HP, Hint Penalty, represents this maximum) or bonus (HB, Hint Bonus, represents this maximum) for a problem related to incorrect or correct hint resolution. These HC and HB parameters have no sense for problems that have no hints related to them. On the other hand, UC (Up Coefficient) is the weight or effect of a hint problem score with respect to the immediately higher level hierarchy problem score. This has no sense in global problems and is only applicable for hint problems that have a parent problem with score.

This model let designers set a fair score system that can be adapted to the particular materials and criterions. The hint visualization penalty can prevent from hint abuse, while the attempt penalization can prevent from attempt abuse. In addition, it promotes active learning where students should make decisions.

Next, the score algorithm calculation for a generic problem P is presented. Firstly, for each global problem, a tree of problems is formed. In this graph or tree, each problem has a parent (for which the descendant problem is a hint) with the exception of the global problem. This can be viewed in Figure 1.
Figure 1: Tree of problems for calculating the score

The formulas for calculating the score of any problem of the tree are the following:

- For level n:

  $$SCORE_n = \frac{(CS_n - AP_n \cdot \text{attemptsnumber}_n)}{MS_n} \cdot NS_n$$

- For level n-1 (the n summand represents the addition of all the problems of level n in the hierarchy. Depending on if the score of a problem is positive or negative, this contributes to add or subtract to the problem score respectively):

  $$SCORE_{n-1} = \frac{(CS_{n-1} - AP_{n-1} \cdot \text{attemptsnumber}_{n-1}) - \sum_{\gamma_n}^{VP_n}(\sum_{\gamma_n}^{\text{SCORE}^{\text{positive}}_{n-1} \cdot \text{HC}_{n-1} \cdot \text{UC}_{n}) - (\sum_{\gamma_n}^{\text{SCORE}^{\text{negative}}_{n-1} \cdot \text{HC}_{n-1} \cdot \text{UC}_{n})}{MS_{n-1}} \cdot NS_{n-1}$$
For an intermediate level $i$ (the $i+1$ summand represents the addition of all the problems of level $i+1$ in the hierarchy. Depending on if the score of a problem is positive or negative, this contributes to add or subtract to the problem score respectively.)

$$
SCORE_i = \frac{(CS_i - AP_i \cdot attempt\text{-}number_i) - \sum_{y=1}^{VP_{i+1}} + (\sum_{y=1}^{V\text{Epositive}_{i+1}} \cdot HB_{y} \cdot UC_{i+1}) - (\sum_{y=1}^{SCORE\text{negative}_{i+1}} \cdot HP_{y} \cdot UC_{i+1})}{MS_{y}} \cdot NS_{i}
$$

From these equations, it can be inferred that all the problem score requisites of the model are fulfilled. Note that a change in the score of a hint problem of level $i$ can have an influence in the score of a problem of level 0 through the different HB and HP coefficients along the hierarchy.

4. Outline of the implemented architecture

Figure 2 shows a general overview of the implemented system architecture. The XML files interchanged between both tools are according to an XML Schema for the hints specification that has been created. This XML Schema sets the possible valid XML documents for describing XML instances of the hints specification and describes how the needed information from the hints model requirements can be represented as a set of XML elements. The XML binding has been derived from the previous hints model requirements and conforms to the W3C XML 1.0 Specification [28].

![Figure 2: General Overview of the implemented hints system architecture](image)

The different elements of the implemented architecture are the following:
- **Hints authoring tool.** It has been implemented as part of a Master Thesis [7] as a context inside the Tomcat Apache Server using the following technologies: JAVA, JSPs, Struts and JDOM. This is a Web-based authoring tool that let designers create XML files that are according to the hints XML binding specification presented in this paper. The tool allows setting whichever possibility of the specification in an easy graphical way through forms rather than writing the XML code by hand.

- **Hints player.** We have implemented the player as a new software module inside the XTutor server. XTutor is an Intelligent Tutoring System developed at MIT (Massachusetts Institute of Technology) as part of the iCampus project (http://icampus.mit.edu/XTutor/). The hints player has been implemented as a new tag handler called hints.tags.py using the Python programming language and it uses different services provided by XTutor such as the database backend, state maintenance or the XMLTag methods. The implemented hints player module together with the XTutor server-side XDOC processor allows loading whichever file that is according to the hints specification that is presented in this paper and run it accordingly to the presented semantics.

XML files according to the hints specification generated by the implemented hints Authoring Tool can be interpreted and executed inside the hint player implemented in XTutor. Another completely different technology could be chosen for implementing the authoring tool or the player. In general, any Learning Management System (LMS) or Intelligent Tutoring System (ITS) could implement a player that can load XML files according to the hints specification and run them according to the presented semantics.

The presented architectural schema for the hints specification is analogous to any e-learning specifications that have an XML-binding. For example, the IMS-LD [4] specification has authoring tools such as RELOAD (http://www.reload.ac.uk/) that can create XML files that can be loaded and run into IMS-LD players, such as the one inside the .LRN Learning Management System (www.dotlrn.org).
5. Examples running with the Hints XTutor Player

In this section, some examples of material created according to the hints specification are shown running with the help of the hints XTutor Player.

Example 1: Sum exercise

Figure 3 shows the example in a specific moment of the student interaction.

Initially, the student can only see a global Fill In the Blank problem that is the sum $17 + 32 + 121 + 23$. The student can request for a hint pressing the ‘plus’ button. In this moment, a hint sequence composed by three hint Fill In the Blank problems is opened. The three hint problems of level 1 in the hierarchy are $121 + 23$, $17 + 32$ and $(17 + 32) + (121 + 23)$. If the student had pressed the ‘plus’ button related to $17 + 32$ then a hint associated to this problem would have appeared on the screen. The student can request for a hint related to $121 + 23$ then at this moment a hint sequence composed by three Fill In the Blank problems
is opened. The three hint problems of level 2 in the hierarchy are \(3 + 1, 20 + 20,\) and \((100) + (20 + 20) + (3 + 1)\). In the same way the \(20 + 20\) problem has a hint composed by three hint Fill In the Blank problems, which are hint problems of level 3 in the hierarchy. Note, that at the leaf of the branches, there are problems without hints, such as \(3 + 1\) for which there is no possibility to request a hint for. When a student clicks the “Check” button, then there will appear a feedback on the right of each problem (global or part of a hint) with a specific feedback depending on if the answer is correct, incorrect or a syntax error.

**Example 2: Equation exercise**

Figure 4 shows an equation exercise in a specific moment of the student interaction for three respective aims. The material delivered by the XTutor hints player depends on the student aim. If the student has an aim called ‘sum’ (case a) then he/she will be delivered with a set of hints related to the sum. If the student has an aim called ‘remember’ (case b) then he/she will be delivered with a set of hints related to remember the equation for calculating the roots. Finally, if the student’s aim is ‘solveequation’ (case c) then he/she will be delivered with a set of hints related to solve other easier equations.
In this presented example, the global problem is a Multiple Choice. If the student’s aim is ‘sum’ then he/she will be able to request for successive hints. At the first level, there is a Fill In the Blank problem as hint. At the second level there is a Multiple Choice problem as hint. Finally, at the third level there is a sequence hint composed by one Multiple Choice and two Fill In the Blank problems. If the student’s aim is ‘remember’ then the student will be able to request for a hint that is a text, but not a problem. Finally, if the student’s aim is ‘solveequation’ then the student will be able to request for successive hints. There are two hints levels that are composed by a Fill In the Blank problem as hint.
**Example 3: Rivers exercise**

In this exercise, the overall question is a Short Answer problem. Figure 5 shows a specific moment of the student interaction with this exercise. The first level hint that can request the student is a Fill In the Blank problem, where the student is asked for 5 rivers. The student loses points by incorrect attempts. Next, a hint Group is presented to the student that is composed by three hints for which the student can select a maximum of two. To make the decision of what hints to take, the student has some help that gives him/her some meta-information about each hint (score penalizations or/and bonus, hint topic, etc.). The student loses points for viewing the hints and he/she receives a message (“You cannot select more hints of this group. You have exceeded the maximum number to select”) when he/she tries to see more hints than allowed for this hint Group.
Both hints are about rivers
Two hints are shown to you. You can select a maximum of two of them.

This is a MC first and a MR second. You will lose 40 points if you select this hint. For correct/incorrect you can win a maximum of 20 and You can lose a maximum of 8.

One of the top 10 rivers flows into Guadalquivir and outlet in Palma del Rio. Which is this River?

**SCORE: 0.0/100.0**

---

Both hints are about rivers
Two hints are shown to you. You can select a maximum of one of them.


Write the name of the river that is not from Spain. 

**SCORE: -10.0/100.0**

Select only the rivers in the top 10 longest rivers of Spain. Please type an answer.

- Lena
- Turia
- Tajo
- Guadaira
- Huna

**SCORE: -6.0/100.0**

One of the top 10 rivers flows into Guadalquivir and outlet in Palma del Rio. Which is this River?

**SCORE: -90.0/100.0**

---

This is a Multiple Choice for all aims. You don't lose anything to see this hint.

You cannot select more hints of this group. You have exceeded the maximum number to select.

---

Write 5 rivers between the top 10 longest rivers of Spain.

- Duero
- Tajo
- Ebro
- Jucar
- Segura

**SCORE: 47.45/100.0**

---

Figure 5: Rivers Exercise

Note from Figure 5 that the first hint of second hierarchy level that the student decided to see is a sequence of two problems: a Multiple Choice and a Multiple Response. While the Multiple Choice problem has no more hints related to it, the Multiple Response has another hint Group where the student can select a maximum of one hint between two possibilities. In this case the student selected a third level hierarchy hint that is a Fill In the Blank. Note that an error attempt of the student in this third level hierarchy hint has an effect on the score of the first level hierarchy Fill In the Blank problem. This is because the error causes a decrease of points in the own problem and this decrease is transmitted through the different hierarchy levels due to the UC and HP parameters according to the presented scoring
equations. In the same way, a correct answer of the third level hierarchy hint problem has a bonus effect on the first level hierarchy problem. Furthermore, note that the visualization of the third level hierarchy hint problem has a directly effect over the score of the second level hierarchy Multiple Response problem due to the VP coefficient, but also and indirect effect over the score of the first hierarchy level hint problem, due to the UC and HP parameters according to the scoring equations.

**Example 4: Assessment exercise**

Figure 6 illustrates an example where there is an assessment with three sections. The second section and the overall assessment have a hint related to them. In addition to it, a problem of a section has a hint that is delivered to the student only in case that the student answers in an incorrect way to the problem, so it is not presented at the beginning. Moreover, this hint for the problem is dependent on the student ‘aim’.
This is the second section: About mountains

Which is the highest mountain of Spain? Please type an answer.

This hint is a section hint.

This is the third section: About sums

$12 + 34 = \phantom{000}$

$17 + 32 + 121 + 23 = \phantom{000}$

This hint is an assessment hint.

Figure 6: Assessment Exercise

6. Courses where the tool is used

We are using this tool in classroom in three different courses: Computer Architecture Laboratory, Communications Software and Information Servers. Table 1 shows an overview of each of these courses, providing information about them. All are given at the Carlos III University of Madrid, offered by the Department of Telematic Engineering.
For these three courses, we created different hint material including problems with different types of hints, ways of scoring, etc., trying to cover the different features of the commented new hint specification. For Information Servers, the problems with hints are provided only in Spanish language, while for the other two subjects are provided in Spanish and also English language, since there is an English group and other Spanish groups. In the three courses, the problems with hints are available for open access for students anytime anywhere during several days, until the exam day. This activity is voluntary and it has not a direct reward to students, but only as a preparation for the final exam.

7. Evaluation of the hints tool

We present results of use of the tool based on quantitative and qualitative analysis. For the quantitative analysis we take into account only the Computer Architecture Laboratory course because this experience
ended (while the experiences in the other two courses are still active in this moment) and because we performed in this course different experiences in previous years 2005 and 2006 using the QTI v1 module of the .LRN platform, so we can compare this tool with the hints player.

### 7.1 Quantitative analysis

The QTIv1 module of .LRN allows assessments based on the QTI v1 specification, but it does not include hints. In years 2005 and 2006 students were able to use it with different problems without hints. The problems were of type Fill In the Blank, Multiple Choice and Multiple Response. Table 2 provides information for each year course edition about the total number of registered students, the percentage of students that did not take the exam, the percentage of students who passed the exam (over the total of registered students) and the average mark of the students (over the presented students to the exam).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of registered students</td>
<td>191</td>
<td>206</td>
<td>141</td>
</tr>
<tr>
<td>Percentage of students that did not take the exam (%)</td>
<td>44.5</td>
<td>19.9</td>
<td>11.3</td>
</tr>
<tr>
<td>Percentage of students that passed the exam (%)</td>
<td>23.1</td>
<td>67.0</td>
<td>78.7</td>
</tr>
<tr>
<td>Average mark of students in the final exam (within the range [0,10])</td>
<td>4.5</td>
<td>6.5</td>
<td>6.5</td>
</tr>
</tbody>
</table>

**TABLE 2: Comparison between different course editions of Computer Architecture Laboratory**

If we merged the results of the two years in which the QTI v1 .LRN module was used (2005 and 2006) and next we compare these merged results with respect to year 2007 in which the hints XTutor player was used, then a big improvement would be observed in all the considered parameters. But the following question arises: Is this improvement because of the use of the hints player instead of the QTI v1 one? From the data we can observe that there is a huge difference between years 2005 and 2006, but the same
assessment method was used in both years. Analyzing the data, we have concluded that among all the factors there was one that had the biggest impact in the huge difference between these two years. In year 2005 class attendance and practical assignment submissions were completely optional, so very few people attended to classes and also very few submitted their practical assignments. On the other hand, in year 2006 attendance to a minimum of classes was required and submission of practical assignments was mandatory. Despite of the fact that this was not the main objective of our present research, we can conclude from this data that attendance to class and submission of practical assignments had a strong positive effect on student learning. Therefore, data from year 2005 should not be taken into account since there was a very important external factor influence.

If we compare data from years 2006 and 2007 we can see an important improvement because the average mark of students in the final exam is the same but there is an important increase in the percentage of students that passed the exam and a decrease in the percentage of students that did not take the exam. Therefore, this data lead to conclusion that the XTutor hints module produced an improvement with respect to the .LRN QTI v1 module. But, a similar natural question arises again: Was there any important external factors between years 2006 and 2007 experiences apart from the fact of the use of the hints module instead of the QTI v1 module? Since courses perform necessary modifications and are in an unstoppable evolution between year editions and some factors are not possible to control, it is difficult to replicate exactly the same conditions between year editions. We have analysed the possible differences between these two year course editions and there were some differences that may be important such as the number of students repeating the course, different initial level of students, modifications on the topics covered, difficulty of the final exam, correlation of the final exam with respect to the QTI or hints questions, number of statistical data, etc. Doing a balance between years 2007 and 2006 factors, we consider that factors that may benefit student’s results in year 2006 are bigger than in 2007. Therefore, as despite that, better results are obtained in year 2007, then we conclude this is because of the positive effect of the used hints module according to the new specification.
Finally, we provide some extra information about the use of the tool in Computer Architecture Laboratory in year 2007 until the exam day. Let consider that a student requested a problem when he/she saw it at least, and let consider that a student requested hints related to the problem when he/she saw one hint of the problem at least. In this way, the maximum number of student problem requests is equal to the maximum of student hints requests and it is the number of problems with hints (16) multiplied by the number of total students registered in the subject (141), which gives a result of 2252. From the data on the system, the number of problem requests was 1760 (which gives a 78.2 % of the maximum of utilization in this aspect) while the number of hints requests was 1012 (which gives a 57.5% of use of hints for problems). The 78.2% means that students were interested in using the tool. The 57.5% of use of hints denotes that students used hints in several problems but they did not do hint abuse which is an undesirable aspect, so the features of the model in global prevented from hint abuse. Doing a more detailed analysis, 19.1% of students did not use the system (we consider it when they took less than 5 problems with hints), 6.4% of students took the problems but they did not use hints (we consider it when they took less than 15% of hints with respect to problems taken), and 12.8% of students did hint abuse systematically (we consider it when they took more than 85% of hints with respect to problems taken). This data reveals that the system was widely used despite of the fact that the activity was voluntary, and that students used the system in a proper way, viewing hints when they need them, but without hint abuse. All of them are positive results.

7.2 Qualitative analysis

From the qualitative analysis of the experiences based on comments between students, informal meetings with students, student’s attitudes, etc. we have obtained positive income about the tool and we have observed strong interest of students for the hints tool. Specifically, some results we have observed are:

- Students like doing different activities with respect to what they usually do. They perceive the tool as something new that stimulates leaving from monotony. And when a student is motivated and interested for discovering the tool, then it is easier to learn.
• Students perceive the tool as useful. They think the tool is useful and helps to pass the subject.

Another important factor observed is the function of the problems with hints as formative assessment and not only as summative assessment. When students encounter problems in solving some exercise they require help and the hints system guide them through the process.

8. Conclusions

The provision of hints is an interesting and relevant aspect in learning that had not been modelled, although there already exist several systems that implement specific features related to it. In this work, a novel hints software player implemented into XTutor using the Python programming language is presented. This tool allows performing different state of the art hinting functionalities, combine them, and also other new relevant functionalities based on our own ideas and teaching experience. The proposed features for the software tool are the requirements of the hints specification and it takes into account pedagogical issues (such as student centered learning, active learning, meaningful learning or student personalization) and other best practices that justify each requirement.

An XML binding of the full hinting data model has been provided. Different authoring tools can generate XML files that are according to the hinting model, while different players can load and interpret these XML files according to the semantics with independence from the authoring tool that created it. In this way, interoperability between different hinting systems and reusability of different hinting descriptions can be achieved. We have implemented the first player that is full compliant to this new hints specification and this software module can interpret and run these XML files with the help of the XTutor system.

At present, we are using the XTutor hints player in classroom in order to let students interact with created hinting material. From quantitative and qualitative data analysis from the experiences, we can infer that the tool is very useful in the learning process and positive results have been obtained.
References


**Figure legends**

Figure 1: Tree of problems for calculating the score

Figure 2: General Overview of the implemented hints system architecture

Figure 3: Sum exercise

Figure 4: Equation Exercise with aims ‘sum’, ‘remember’ and ‘solveequation’

Figure 5: Rivers Exercise

Figure 6: Assessment Exercise