

Creating and Administering Interactive Online Tutorials and Performance Evaluation Tests through a Novel Web Platform

Carlos R. Jaimez-González, Christian Sánchez-Sánchez, J. Sergio Zepeda-Hernández
*Departamento de Tecnologías de la Información,
Universidad Autónoma Metropolitana - Unidad Cuajimalpa, México*

Abstract

Distance learning applications or computer-supported collaborative learning (CSCL) are forms of education that use several resources and self-learning mechanisms. There are pedagogical techniques, such as tutorials, which can help the process of learning if they are implemented with the appropriate technology. Currently, there are many tutorials accessible through the Web. However, the Web sites that provide the tutorials lack of a platform for their creation and administration. Also, the content of those tutorials is limited in most of the cases to only text and images, which avoids interactivity with the users. Moreover, these tutorials do not provide any means to evaluate the performance or progress achieved by the students.

In this paper we present a novel Web platform that allows the creation and administration of interactive online tutorials and performance evaluation tests. This platform is easy to use and guides the tutorial creator to produce tutorials, which contain not only text and images, but also other types of multimedia elements, interactive examples, and a set of different tests to evaluate the performance of the students.

1. Introduction

Information technologies give the possibility to offer products and services to millions of people. The Internet has different services such as the Web, which provides online access to applications and programs. Examples of these applications can be found in many universities around the world, which offer distance learning applications, or computer-supported collaborative learning (CSCL) [1]. This form of education uses several resources and self-learning mechanisms.

There are pedagogical techniques, such as tutorials, which can help the process of learning if they are implemented with the appropriate technology. A tutorial is “a series of interconnected instructions, a step-by-step guide that aims to teach one and only one task” [2].

Currently, there are many tutorials accessible through the Web [3], [4], [5]. However, the Web sites that provide the tutorials lack of a platform for

their creation and administration. Also, the content of those tutorials is limited in most of the cases to only text and images, which avoids having interactivity with the user. Moreover, they do not provide any means to evaluate the performance of the tutorial user, such as tests, which would be useful to determine the progress achieved.

In the Information Technologies and Systems undergraduate program of our university, we teach courses on programming languages and Web technologies, such as HTML, JavaScript, XML, XSL, Java, C++, among others. These programming languages and Web technologies require interactivity (human-computer) in order to master them, because it is necessary to write programs, compile/interpret them, execute them, and test them. Due to the nature of the courses we teach, we are developing a novel Web platform to be used by lecturers/teachers (of our undergraduate program initially) that allows the creation and administration of interactive online tutorials. This platform is easy to use and guides the tutorial creator to produce tutorials which contain not only text and images, but also multimedia elements, such as video, animations, sound files, examples, etc. Our Web platform also supports the validation and execution of code fragments (when it is required), through interactive elements, where the student can practice at the moment of reading or studying the tutorial. Additionally, it supports the creation and administration of performance evaluation tests, which are intended to evaluate the performance of a student for particular topics of a tutorial, or for an entire tutorial.

This paper is organized as follows. The following section presents some related work about tutorials. Section 3 describes the Web platform for the creation and administration of interactive online tutorials and performance evaluation tests, its architecture, the storage mechanism, and the execution of programs. In section 4 we present the process of creating a tutorial; where we introduce the types of multimedia and interactive elements allowed in a tutorial, the structure and sequence of topics in a tutorial, and the visualization of a tutorial once it has been created. Section 5 is dedicated to describe the different types of performance evaluation tests that our platform supports. A following section discusses the usability principles we consider for our Web platform.

Conclusions are presented in section 7. Finally, section 8 presents our future work.

2. Related work

This section presents some related work. There are very few papers focused on tutorials, and we have not found any related work or development of a platform for the creation and administration of interactive online tutorials, nor about performance evaluation tests in this field.

There are Web sites to generate simple tutorials, such as *LiveJournal* [6], which offers “how to” tutorials. However, the tutorials that can be generated are very simple in its content (HTML); the Web site does not guide the user to develop the tutorial; it is not easy to be used by any user, because it requires the use of some HTML tags; and additionally, once the tutorial has been created, it cannot be administered. *LiveJournal* is closer to be an HTML generator rather than a tutorial creator.

Aguilar et al. [7] carried out a study of an intelligent tutorial for teaching maths (sums and subtractions), but it was focused on the verification and validation of the system. The aim of the system is to personalize the teaching process based on the features or skills of every student. The system uses artificial intelligence (fuzzy logic and multi-agent systems), in order to generate plans of activities for the students, monitor them, and plan them again if necessary. The activities of a tutorial are presented with multimedia elements in order to be easy to learn. The tutorial also allows storing the student evaluations, and sending them to the administrator, so the progress of the student can be measured.

The verification of the system is carried out through a series of tests: verifying that all its parts work independently (unit testing); putting all the parts together and verifying that they still work (integration testing); and finally, the system is verified as a whole (system testing). The validation of the system is carried out in two ways: qualitative (evaluating performance and usability), by recording the users of the system and keeping track of the mouse movements while they were using the tutorial; and quantitative through a series of statistics, where they compared the answers of the system against to the answers of experts. However, this study is only focused on one tutorial for pre-school and primary school children.

The Open Universities Australia (OUA) company worked on a project called Tutorial Support Management (TSM), which was intended to provide the same quality in teaching and learning, but through the use of online facilities. The disciplines included in this project were accounting, computer studies, criminology, business, internet studies, arts and primary education. TSM brought advantages, such as the improvement of student retention,

performance and satisfaction. The seven key elements of the TSM model were reported in [8] by Lynch and Paasuke, and they are the following: 1) Lowering the student: tutor ratio in high enrolment units; 2) Determining the required number of tutors, the means for identifying and contracting the tutors; 3) Organizing and delivering paid tutor training in online student-centered discussion (OSCD); 4) Preparation of Tutor guides by Course Coordinators to be provided to tutors; 5) Tutor facilitation of weekly, structured asynchronous content-related online student centered discussions, as well as other means of support such as email; 6) Tutors to maintain a log of activities; 7) The employment of a TSM Project Manager and Online Tutor Supervisor to offer coordination and support to Providers implementing the TSM model and to review the results of these activities.

The paper by Lynch and Paasuke [8] demonstrates that there are effective methods for distance education. It also gives evidence of the need of a system/platform to make easy the process of learning in the key elements. Such system or platform does not exist.

3. The Web platform

Our platform allows the creation and administration of interactive online tutorials, and performance evaluation tests. It is accessible through the Web, and allows storing different types of resources (elements) as part of a tutorial. Additionally, it allows including interactive elements, which can be executed online. In the following subsections we will give more details.

3.1. The architecture

The architecture of our platform has three main components, which are described below and shown in Figure 1.

Web interface for the administrator and the tutorial creator. Through this Web interface, the user (administrator or tutorial creator) can have access to the main functionality of the platform (according to the user rights and permissions). An *administrator* is the most powerful user, who has all the rights and permissions in the platform. A *tutorial creator* is a user that is able to create tutorials and performance evaluation tests, and administer them.

Administration module. This module is intended for the administration of tutorials. It allows four important actions. a) *User accounts administration*, which controls the access to the platform and specifies the rights and permissions for every user: *administrators*, *tutorial creators (lecturers)*, and *tutorial users (students)*. b) *Storage controller*, which handles the disk space where the tutorial resources will be stored. c) *Execution controller*,

which controls where the different components (interactive elements) will be executed. d) The actual *interface for creating tutorials*.

Creation module. The purpose of this module is the creation of tutorials. Through this module, the tutorial creator user can actually create a tutorial by assembling the different multimedia and interactive elements allowed by the platform. The creation of the tutorial is guided and validated by the creation process, which is explained in section 4. The creation module contains two libraries: one for the multimedia elements allowed; and another one for the interactive elements allowed.

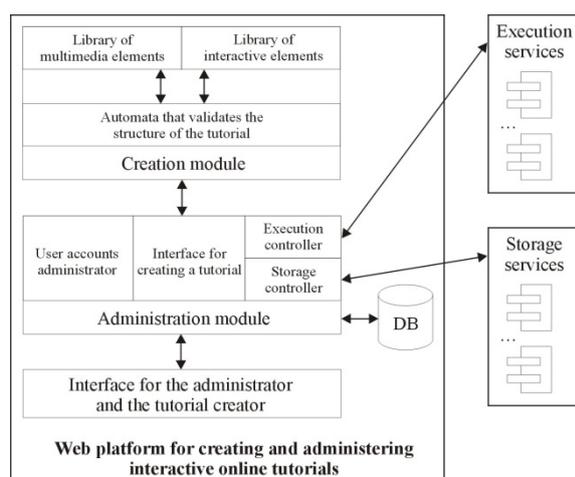


Figure 1. The architecture of the platform

3.2. The storage mechanism

The resources (multimedia and interactive elements) used as part of the tutorials in our platform, are not located in the same server as the platform. This is mainly for two reasons: disk space and security.

Because the multimedia elements (images, videos, animations, sound files, etc.) used in a tutorial can occupy several MB of disk space, we leave the tutorial creator to decide where to store them. This is achieved by representing such elements as hyperlinks when the tutorial is being created, so that the elements are actually stored in a different server. Regarding the security issue, we decided to restrict the storage of elements in our platform server because of possible virus in the files to be uploaded, and due to the execution of programs or components, which are supported by our platform, as it will be explained in the following section.

When the tutorial is being read or studied by a tutorial user (normally a student), our platform makes transparent the fact that the resources (multimedia and interactive elements) are located at different computers. This is achieved by using the storage and execution services (shown in Figure 1), which are implemented by using Web services. A

Web service is a piece of software or procedure that can be accessible to other programs or applications through the Internet.

3.3. Execution of programs

One of the novel ideas that we introduce in our Web platform is the execution of interactive elements, which can be code fragments in any programming language, or even complete programs to be executed as part of the topics of a tutorial.

The execution of these programs is carried out on the client side (Web browser) or on the server side. Because of security issues with the execution of programs, we decided that these interactive elements (the ones that are to be executed on the server side) must be executed in a different computer from the one that holds our Web platform. This computer has to be specified by the tutorial creator when the tutorial is being created. Since we are leaving the execution of code to a different computer, the Web platform is protected from any malicious code that could potentially be executed on it.

When the tutorial is being read by a tutorial user, the execution of these programs or code fragments is transparent, since the code is executed in the server side. This process is also carried out by the storage and execution services, shown in Figure 1, which are implemented by using Web services.

4. The process of creating a tutorial

In order to determine a process for the creation of tutorials, we analyzed some papers about design and methodologies of tutorials [9], [10]. We also reviewed some other papers about creation of tutorials [2], [11]. Additionally we had a look at some existing tutorials and systems in different fields [3], [4], [5], [6], [12], [13]. Our analysis was carried out with the purpose of identifying the elements that are part of a tutorial, its structure and sequence. The results are described in subsections 4.1, 4.2 and 4.3. Section 4.4 shows how a tutorial is visualized once it has been created.

4.1. Elements

This subsection describes the possible elements of a tutorial, according to our analysis.

Name of the tutorial (e_1). The name gives an idea of the contents of the tutorial.

Purpose of the tutorial (e_2). It helps the tutorial user to determine whether the contents of the lessons will be of interest.

Table of contents (e_3). This table provides the topics that the tutorial covers. This helps to understand the organization of the tutorial.

Conventions (e_4). The purpose of the conventions is to help to understand the contents of the tutorial.

Copyright (e₅). This element describes the owners of the contents of the tutorial.

Notes (e₆). These notes specify the requirements and recommendations for the users of the tutorial.

Name of the topic (e₇). It gives an idea of the topic or lesson to be taught. This applies to topics and subtopics. Every topic has a name.

Explanation of the topic (e₈). It provides a short introduction or explanation of the topic or lesson to be taught.

Multimedia elements (e₉). These elements help to better understand the topics; and they can be images, videos, sound files, and animations.

One-step explanation (e₁₀). This explanation describes what the tutorial user should do in order to reach a specific objective, or to learn a specific functionality.

One-step example (e₁₁). This element provides an example of one step, in such a way that the tutorial user can better understand.

Interactive example (e₁₂). This type of element provides an example, and gives the user the opportunity to interact with it, by following it, or modifying it. With this type of element the user is allowed to visualize the results of the execution of the example. There are many types of examples considered, but at the moment we only have a small set of programming language examples, such as CSS+HTML, JavaScript+HTML, XML+XSL, etc.

Interactive exercise (e₁₃). This type of element is used to request the tutorial user to do an exercise. It is necessary to describe the aim of the exercise, so the tutorial user knows the result to be achieved. This interactive exercise is supposed to rely on the previous lessons of the tutorial. The interactive exercise is normally used for the performance evaluation tests (see Section 5).

Performance evaluation tests (e₁₄). These elements are tests that are intended to evaluate the performance of a student for particular sections of a tutorial, or for an entire tutorial. Section 5 is dedicated to describe the different types of performance evaluation tests that our platform offers.

4.2. Structure

We define the structure of a tutorial through a regular expression [14], which helps us identify the automata for validation. The structure of a tutorial is defined as follows.

$$e_1e_2e_3e_4?e_5?e_6? [e_7e_8e_9*[e_{10}(e_9|e_{11}|e_{12}|e_{13})*e_{14}*]]^+ \quad (1)$$

Where e_i ($i=\{1,...,14\}$) represents the elements described in section 3.1. The “?” symbol indicates zero or one occurrences of the element; the “*” symbol indicates zero or more occurrences of the element; and the “+” symbol means that the element can appear one or more times. The square brackets “[

” and the parentheses “()” are only used to group elements; and the “|” symbol is used as the logic operator “OR”.

According to our analysis, the creation of a tutorial follows the structure described in (1). For example, when a tutorial creator starts creating a tutorial, the first step will be to provide the name and purpose of the tutorial; next it will be needed to specify the table of contents; after that, the conventions; then the copyright and notes; next the topics of the tutorial, and so on.

Figure 2 illustrates the Web interface that allows the creation of a tutorial, with some of the elements described in subsection 4.1. The tutorial creator is responsible for the input of every element of the tutorial: name and purpose, table of contents, conventions, copyright, notes, and topics. For every topic, the tutorial creator will input one of the actual topics of the tutorial, which can contain text, images, videos, animations, sound files, interactive examples, and interactive exercises.

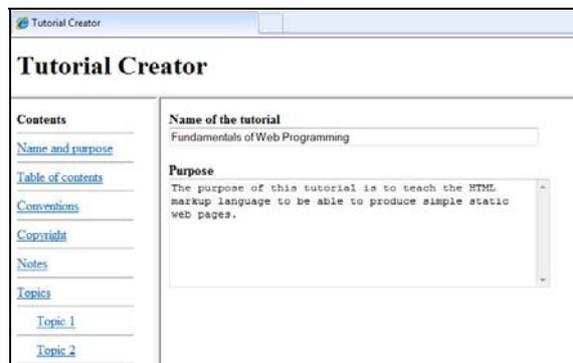


Figure 2. Web interface for creating a tutorial

Figure 3 illustrates the Web interface to input one of the different interactive exercises supported in the platform (CSS+HTML), which allows the tutorial creator to specify fragments of code in CSS and HTML.

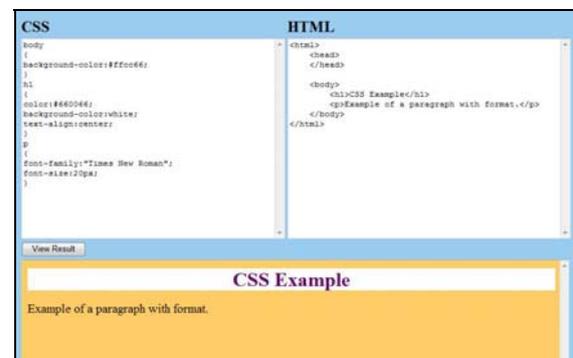


Figure 3. Creating a topic with an interactive exercise (CSS+HTML)

The inclusion of interactive elements in a tutorial will help the tutorial user to better understand the lessons, since the programs (code fragments) are executed online, without the need of installing any program or environment on the computer of the tutorial user.

4.3. Sequence

A sequence of topics and/or steps should be established once the tutorial has been created. The sequence will define the order in which the topics of the tutorial should be covered by the tutorial user. Because there are different types of tutorials, the definition of a sequence is optional.

If we think about a tutorial to learn how to cook, and each recipe is a topic in the tutorial, it is not important what recipe is read first by the tutorial user, because the recipes are independent. In this case it is not necessary to establish a sequence for the topics, as can be seen in Figure 4.

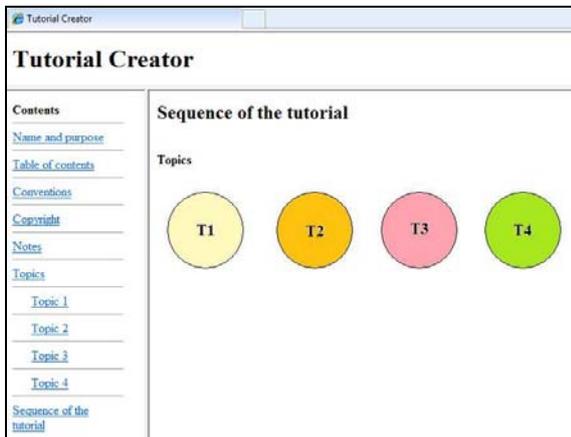


Figure 4. A tutorial with no sequence of topics

However, if we think about a tutorial to learn how to program in the C programming language [9], we cannot skip topic 1 (which is about *data types and pointers*), and jump to topic 4 (which is about *data structures*), because the tutorial user would not be able to understand topic 4. In this case, the tutorial creator must establish a sequence in the tutorial, where the topic 1 would be the first topic to be read by the tutorial user, next topic 2, then topic 3, and so on. This is illustrated in Figure 5.

The sequence of topics has more sense when the tutorial user has also defined a performance evaluation test for each topic of the tutorial. The tests can be used to determine whether the student can continue with the following section of the tutorial, based on the score obtained. Please see Section 5 for a description of the different types of performance evaluation tests that the platform offers. These tests can be defined for every topic of a tutorial.

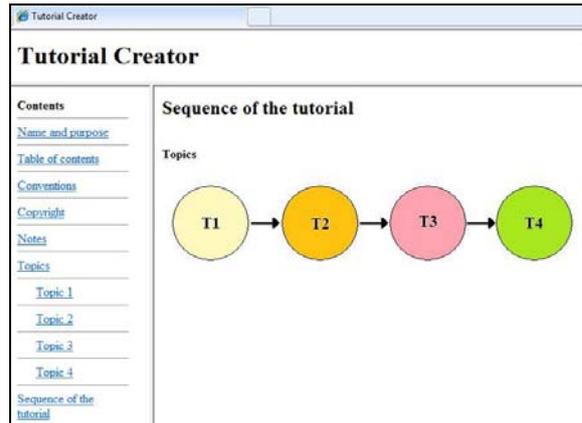


Figure 5. A tutorial with a sequence of topics

4.4. Visualization

Once the tutorial has been created, and a sequence of topics has been established, it is ready to be published and made accessible to potential tutorial users (students). We are considering having different styles of visualization, which would be determined by the tutorial creator.

Figure 6 shows an example of the visualization of a tutorial that is ready to be studied by tutorial users. It is divided in three panels: the top panel shows the name of the tutorial; the left panel contains the table of contents, with topics and subtopics; and the right panel shows the contents of a specific topic of the tutorial (observe that a topic already includes text and multimedia elements, such as images, videos, and audio files). Section 6 explores some usability principles that we are considering to make tutorials more attractive and effective to users.



Figure 6. Visualization of a tutorial

5. Performance evaluation tests

In order to measure the performance of a student while studying a tutorial, our platform allows defining *performance evaluation tests*, which are

intended to evaluate the performance of a student for particular topics of a tutorial, or for an entire tutorial. In this section we describe the different types of performance evaluation tests that our platform offers. We also explain the aspects taken into consideration for the evaluation of a test.

5.1. Types of tests

In this subsection we provide a description of the different types of performance evaluation tests supported by our platform. The types of tests are determined by the types of questions that they allow. We have classified them in six types of tests: multiple-choice questions, true/false questions, free-text questions, interactive-exercise questions, mixed questions, and other tests.

When a tutorial is being created, the tutorial creator can also create a performance evaluation test for every topic of the tutorial, so the performance of the student can be measured on a topic basis. A test can also be defined for the entire tutorial.

Multiple-choice questions. For this type of test, the platform allows specifying the number of questions, the questions themselves, and their possible answers. In order to define this type of test, the tutorial creator provides the number of questions, and then proceeds to enter the questions (one by one) together with their answers. For each question entered only one answer can be correct, and this must be specified by the tutorial creator once all the answers have been entered.

For the tutorial user (student) this type of test will be presented as a list of questions, where each question will be followed by all its possible answers. The student will be able to select only one of the answers and continue with the rest of the questions. Once the test has been completed by the student, the platform will evaluate every answer, by comparing it with the correct answer given by the tutorial creator; and at the end of the evaluation a score will be presented to the student.

True/false questions. Similarly to the previous type of test, the platform allows defining the number of questions, the questions, and the possible answers for every question. The questions are also entered one by one, but in this type of test only two possible answers are allowed: True/False, Yes/No, 0/1, or any pair of answers that can be defined by the tutorial creator. For every question only one of the two answers can be correct, and this must be specified by the tutorial creator.

This type of test is presented as a list of questions with two possible answers for each question. The student is able to select only one of the two answers for each question. At the end of test the platform will evaluate the answers given by the student, and a score will be provided on screen.

Free-text questions. This type of test only allows specifying the number of questions, and the questions. The tutorial creator cannot create a fixed set of possible answers for every question.

Because of its nature, this type of test is presented to the student as a list of questions with an empty text area for each question, where the student is able to write an answer. Once the test has been completed, all the answers are stored in the platform. The evaluation of the answers will be carried out by the tutorial creator at a later time, and a score will be given to the student. This type of test is more suited for those questions where an answer cannot be restricted to a fixed set of answers.

Interactive-exercise questions. For this type of test, the platform allows specifying the number of questions, the questions, and for each question the type of interactive exercise associated with it. The only interactive exercises allowed in this type of test are those that have already been registered with the platform (for example: HTML+CSS, XML+DTD, XML+XSLT, HTML+JavaScript, etc.). The tutorial creator enters the questions one by one, together with the associated interactive exercises.

In this type of test the student is presented a list of questions with their associated interactive exercises in a text area. The student is allowed to interact with the exercise by modifying it in the text area provided. Once the test has been completed, all the answers (the interactive exercises) are stored in the platform. The evaluation of the answers (which include the interactive exercises) will be carried out by the tutorial creator at a later time, and a score will be given to the student. Part of our further work for this type of tests, is to automate the process of evaluating an interactive exercise, by providing an evaluation module for every type of interactive exercise registered in the platform.

Mixed questions. This type of test allows a mixture of questions from the other types of tests provided by the platform.

The visualization of this type of test is also as a list of questions, where the student will answer each question according to the types described previously. It should be noticed that in this type of test, the platform will only score those answers that can be evaluated automatically; and for those answers that require the tutorial creator to evaluate them (such as the free-text and interactive-exercise questions), the platform will store the answers, and give the score at a later time.

Other tests. We believe that the majority of topics in a tutorial can be evaluated with any of the five different tests supported by the platform. However, we are considering the addition of tests with other different types of questions, such as: word association, where the student associates words from two columns; sentence completion, where the student completes a sentence with specific words; image

association, where the student associates images from two columns; among others.

5.2. Evaluation

In order to measure the performance of a student, the evaluation of every test is required. There are three aspects that control the evaluation of a test in our platform: the numerical value given to individual questions, the time to complete the test, and the passing score.

Value of questions. Every question registered in the platform as part of a test has a numerical value associated, which is entered by the tutorial creator. The numerical value is used to calculate the score of a student, once a test has been completed. The total score that a student can get for a test is calculated as the sum of all the individual values of the questions that are part of the test.

Time to complete a test. The tutorial creator can specify the maximum period of time that a student can take to complete a test. The time is entered as a number of seconds. In a future release of our platform, we are considering the possibility of providing another type of presentation of our tests; not only as a continuous list of questions, but also presenting the questions one by one. In this way, we could also associate a maximum period of time for answering every individual question.

Passing score. The platform allows defining the passing score for every test, so that the performance of the student can be measured. The passing score is also used to determine whether the student can continue with the following topic of the tutorial (if the topics have a sequence specified). If the passing score is not important to continue to the next section, the tutorial creator can specify it as zero.

The passing score cannot be greater than the sum of all the individual values associated to the questions of a test (this is validated by the platform).

6. Usability

Because our Web platform is used for educational purposes, the usability is an important issue to be considered. A system that is usable must be effective, efficient, easy to learn and easy to remember [15], [16]. We are following several design principles in our platform in order to improve its usability. We have divided them in three levels: the development level, the tutorial creation level, and the visualization level. In this section we discuss the principles we use in each of the levels described.

6.1. Development level

The development level refers to the actual implementation of the platform. We consider two

main usability principles for the development of the platform: efficiency and accessibility.

Efficiency. This principle is normally related to the time spent by a user on a particular Web page to complete a task. We achieve efficiency in the development of the platform in several ways: providing light Web pages, which separate the programming code; creating modules according to the functionality; using external services for execution of programs; using external storage mechanisms for multimedia elements; etc. This does not only achieve the efficiency principle, but also allows a scalable and flexible platform.

Accessibility. This principle is related to how accessible our platform is. We achieve this principle by providing access to our platform through the Web, without restricting its use to any special Web browser, or commercial software. We also avoid the use of plug-ins or external libraries to be installed on the user computers. The platform is being developed using standard technologies such as XHTML, CSS, DOM, JavaScript, among others. Additionally we also consider the use of cross-browser compatibility techniques where necessary.

6.2. Tutorial creation level

The tutorial creation level has to do with the Web interface used by the tutorial creator/administrator. The main issues we are considering for the implementation of this Web interface are related to the ease of use, and the manipulation of the basic multimedia elements it allows: images, videos, sound files, interactive examples, and text. In this level we consider four usability principles: consistency, visibility, feedback, and simplicity.

Consistency. This principle is about maintaining the same style throughout the platform. This is achieved, for example, by using the same appearance and position for common elements, such as buttons, menus, forms, links, etc., so that the user of the platform can learn the functionality more easily. In our platform, when the tutorial creator starts creating a new tutorial, a design and page style can be specified, which will be used throughout the tutorial. When the tutorial is visualized by the students it will be shown with the same appearance in all the Web pages, ensuring visual consistency.

Visibility. The visibility principle aims to have all the important parts of an interface “visible” to the user, which means that the platform must be intuitive for a user to find specific functionality by only looking at the buttons of a menu, or the text in hyperlinks, for example. This is normally achieved by grouping elements with similar functionality, using tool tips on buttons, using image buttons, adding descriptive texts to hyperlinks, for example.

Feedback. This refers to provide a *reaction* for the *actions* carried out by the user of the platform, so

that the user knows the next action or set of actions to be done [17]. Our Web platform clearly shows the actions to be taken by a tutorial creator when creating a tutorial through its wizard with a status bar that indicates the tasks/steps already completed. If a task or action cannot be completed by the platform, a notification message is shown to the user with the cause of the problem, so that the user knows what to do next. If feedback is appropriately implemented, it can help to fix problems quicker.

Simplicity. It is one of the most important principles on a Web interface. For Maeda [18], the simplest way to achieve simplicity is through thoughtful reduction; which means that unnecessary icons, buttons, boxes, lines, graphics, shading, and text should be removed to offer a cleaner and easier interface. In order to achieve simplicity in our Web platform we only offer the functionality and features that the user needs to complete a certain task.

6.3. Visualization level

The visualization level refers to the Web interface seen by the students of a tutorial. Currently, we only include principles based on the content and presentation, focusing mainly on the design of the user experience [19], [20]. The aim of this level is to provide a pleasant user experience for the student. We are still working on following some basic usability recommendations on content and presentation, such as those concerned to the use of lowercase and uppercase fonts, background colors, font styles, font sizes, font colors, text alignment, text spacing, etc.

7. Conclusions

In this paper we have presented a novel Web platform for creating and administering interactive online tutorials, and performance evaluation tests. The platform is easy to use and guides the tutorial creator to produce tutorials which contain not only text and images, but also multimedia elements, interactive examples, and different types of tests to measure the student performance.

One of the novel ideas that we introduce in our Web platform is the execution of interactive elements (on the client side or the server side), which can be fragments of code in any programming language, or even complete programs to be executed as part of the topics of a tutorial.

The possibility of creating and administering a variety of performance evaluation tests for every topic of a tutorial is also a very important feature of our Web platform, which makes it ideal to measure the progress of students.

8. Future work

We require further work to complete the development of this Web platform, and to support more types of interactive elements as part of the tutorials, such as programs in different programming languages, or the validation of fragments of code with external execution services.

We also plan to support additional performance evaluation tests, and to provide different types of presentation for them, not only as a continuous list of questions. Concerning the automation of the process to evaluate an interactive exercise test, it also requires further work to implement a separate module that provides evaluation services.

The contents and visualization of tutorials are other important issues to be considered in our platform, in order for tutorials to be effective learning tools for our students. We are working on the implementation of all the usability principles on our platform.

9. References

- [1] Stahl, G., Koschmann, T., & Suthers, D. "Computer-supported collaborative learning: An historical perspective", in R. K. Sawyer (Ed.), *Cambridge Handbook of the Learning Sciences* (pp. 409-426), Cambridge, UK, Cambridge University Press, 2006.
- [2] Max Ferzzola, "Como escribir un tutorial (how to write a tutorial)", <http://www.neoteo.com/como-escribir-un-tutorial-5138.neo> (Accessed in December 2010).
- [3] SUMS, <http://www.gla.ac.uk/sums> (Accessed in December 2010).
- [4] Tutorio, <http://www.tutorio.com> (Accessed in December 2010).
- [5] W3schools, <http://www.w3schools.com> (Accessed in December 2010).
- [6] Howtogenerated (a tool to generate simple tutorials), <http://www.queech.com/janine/howto/howtogenerated.html> (Accessed in December 2010).
- [7] R. M. Aguilar, V. Muñoz, M. Noda, A. Bruno, and L. Moreno, "Verification and Validation of an Intelligent Tutorial System", *Journal of Expert Systems with Applications*, Pergamon Press Inc., Tarrytown, NY, USA, Volume 35, Issue 3, pp. 677-685, October 2008.
- [8] G. Lynch, and P. Paasuke, "Key Elements of the Tutorial Support Management Model", *Australasian Journal of Educational Technology*, Volume 27, No. 1, pp. 28-40, 2011.
- [9] Carol C. W. Hulls, Adam J. Neale, Benjamin N. Komalo, Val Petrov, and David J. Brush, "Interactive Online Tutorial Assistance for a First Programming Course", *IEEE Transactions on Education*, Volume 48, No. 4, pp. 719-728, November 2005.
- [10] Eugenia Fernandez and Elizabeth Rubens, "Developing Online Tutorials: The Methodology and the

Reality”, Proceedings of the 35th Annual Conference on Frontiers in Education, 2005.

[11] Dan Richard, “How-To-Write-Effective-Tutorials”, <http://www.tutorialized.com/view/tutorial/How-To-Write-Effective-Tutorials/12606> (Accessed in December 2010).

[12] Eric Roberts, “An Interactive Tutorial System for Java”, Stanford University, Technical Symposium on Computer Science Education, Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, Houston, Texas, USA, pp. 334-338, 2006.

[13] Thomas G. Cleaver, “Interactive Web-Based Tutorials for Engineering Education”, Proceedings of the IEEE Southeastcon Conference, 1999.

[14] Kleene, Stephen C. “Representation of Events in Nerve Nets and Finite Automata”, in Shannon, Claude E., McCarthy, John, Automata Studies, Princeton University Press, pp. 3-42, 1956.

[15] J. Nielsen, and H. Loranger, *Prioritizing Web Usability*, New Riders, 2006.

[16] J. Nielsen, and M. Tahir, *Homepage Usability: 50 Websites Deconstructed*, New Riders, 2002.

[17] D. Norman, *The design of all things*,

[18] J. Maeda, *The laws of simplicity*, The MIT Press, 2006.

[19] R. Unger, C. Chandler, *A Project Guide to UX Design: For user experience Designers in the Field or in the Making*, New Riders, 2009.

[20] M. Notess, *Usability, User experience and Learned experience*, Indiana University.