Athena\textsuperscript{Q1}: A Hybrid Management System for Multi-Device Educational Content

RAQUEL VÁSQUEZ-RAMÍREZ,\textsuperscript{1} GINER ALOR-HERNÁNDEZ,\textsuperscript{1} ALEJANDRO RODRÍGUEZ-GONZÁLEZ\textsuperscript{2}

\textsuperscript{1}Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Av. Oriente 9, 852. Col. Emiliano Zapata, C.P. 94320, Orizaba, Mexico

\textsuperscript{2}Computer Science Department, Universidad Carlos III de Madrid, Av. Universidad 30, Leganés 28911, Madrid, Spain

Received 15 March 2012; accepted 28 May 2012

ABSTRACT: Over the past few years, a number of software applications have been proposed to tackle the administration, documentation, tracking, and reporting of training programs and contents such as CMS (Content Management System) and LMS (Learning Management System). However, CMS focuses on the administration and development of websites, while LMS is mainly focused on the management of training through the use of learning objects. CMS and LMS have features that can be used in conjunction in order to create educational environments which enhance the teaching–learning process in classrooms. This article presents a hybrid management system called Athena for multi-device educational content providing a set of Web-based interfaces for managing, presenting, and generating PHP-based applications with educational content, reducing both the time and effort invested during the development process of this type of software. Athena borrows the main features of both CMS and LMS in order to manage and reuse educational content. Athena allows the user to develop platform-independent multi-device educational software using existing educational content. Athena aims to provide new forms of accessibility to educational content, envisioning a better future for online education. To demonstrate Athena’s capabilities, we have presented a case study for generating a multi-device physics course to emphasize our contribution. © 2012 Wiley Periodicals, Inc. Comput Appl Eng Educ 9999:1–14, 2012; View this article online at wileyonlinelibrary.com/journal/cae; DOI 10.1002/cae.21567

Keywords: CMS; LMS; hybrid management system; educational software; educational environments

INTRODUCTION

Nowadays, CMS (Content Management System) and LMS (Learning Management System) are responsible for this management through the use of a toolkit set which reduces any given organization’s working time. While content management may seem at first glance a way to create great websites, it is a fact that it is quite a complex process that involves collecting, managing, and publishing content \cite{1}. Although LMS and CMS share the function of information management, both are focused on different fields. For instance, LMS is mainly focused on the management of training education activities \cite{2}, while CMS is focused on digital content management in webpages \cite{3}. From this point of view, LMS is a specialized CMS used to manage learning-objects-based content, which include courses, assessments, and support materials for teaching, while CMS can be customized to manage websites in a number of different fields. The use of CMS and LMS arises from a set of completely different needs for an organization. It is noteworthy that CMS and LMS do not actually create content \cite{4}. The CMS and LMS tools are integrated with authoring tools and provide friendly interfaces for managing content when a user avails of these kinds of software applications. Both allow structure definition through a content schema or content model to manage content as data \cite{2,3} and to render content into documents following a specified layout through the use of templates \cite{4}. However, both have some disadvantages, given

---

\textsuperscript{Correspondence to G. Alor-Hernández (galor@itorizaba.edu.mx).}

© 2012 Wiley Periodicals, Inc.
the difficulties which might manifest themselves during the content management process, it is sometimes necessary to make large-scale changes in order to get the expected result. The facilities that CMS and LMS provide for website content storage, in many cases, would be extremely useful for said content to be made available on mobile devices, rather than desktop or portable computers, therein allowing users to access the software on a device that they can carry anywhere.

In recent years, mobile devices have successfully supported educational processes by allowing the software to be used at almost any given time, even when the user is standing rather than sitting. Mobile devices offer greater independence, in terms of location and time; when compared to web-based education accessed via computers [5]. One of the benefits of displaying educational content on mobile devices is usability; however, some constraints, such as processing power, storage space, and communication capabilities, must also be taken into account. In this sense, multi-device interfaces can improve interaction for different users and in different usage contexts, increasing performance, stability, robustness, expressive power, and efficiency. Multi-device interaction is a characteristic of everyday human activities and communications, in which we speak, listen, look, make gestures, write, draw, touch, and point, alternatively or at the same time in order to achieve an objective. Taking this into account, there is a need to develop a hybrid content management system which borrows the best features of LMS and CMS, in addition to adding new functionality with the aim of enabling the generation of the end-user application including content to be displayed on multi-device interfaces. A hybrid content management system solves the problem of managing and reusing educational content on different formats, styles, and designs. This article presents the development of a hybrid content management system called Athena to manage the educational content to be presented as educational applications providing multi-device interfaces.

RELATED WORKS

In recent years, several works have been proposed with the aim of improving the teaching–learning process. Some of these works emphasize the development of LMS and CMS by adding new features. Huang et al. [6] discuss that formative assessment, summative assessment, and self-assessment need to be used on different devices in order to support students in the teaching–learning process. Ghiani et al. [7] developed the Cicero designer, a tool to allow people without programming experience to create and modify guides accessible through both mobile and large screen stationary devices. Simeone and Ardito [8] describe the prototype of a framework for designing interactive applications for cultural heritage sites by following an end-user development approach. The project is focused on cultural heritage experts and visitors to cultural sites. Van der Schaaf et al. [9] present a set of educational tools to support students when learning how to design models based on a 5-step modeling approach. This article describes the design and use of two tools used in a case study to get student and teacher feedback. The first tool is the ModelComposer, this supports the student when composing a mathematical model and second is the ModelRunner, that lets the user execute experiments with the model.

Stock et al. [10] developed a suite of interactive and user-adaptive technologies for museum visitors. This project is based on: (1) animated agents to motivate visitors and focus their attention; (2) the generation of adaptive video documentaries on mobile devices; and (3) the generation of post-visit summaries that reflect the individual interests of visitors as determined by their behavior and the choices made during their visit. Roselli et al. [11] developed an online Free Body Diagram (FBD) Assistant that allows students to construct 2D FBDs and to receive constructive feedback for a wide range of practice problems. The system’s architecture allows the interoperability between learning management systems and interactive student simulations designed to improve both learning and assessment. Ciluglug and Inceoglu [12] discuss the personalization issues in e-learning systems and the technologies that are used for designing and developing adaptive distributed learning environments. They present a number of technologies that can be used to achieve personalization in adaptive distributed learning environments, such as: (1) web services; (2) Semantic Web; and (3) AI techniques. Rodriguez et al. [13] describe the environment used in the Computer Architecture Department of the Technical University of Madrid (UPM) for managing small laboratory work projects and a specific application for an Assembly Language Programming Laboratory. Breuer et al. [14] present different software prototypes to support collaborative learning in the classroom through interactive technologies. They enhance interaction with whiteboards and allow students to work in groups and access whiteboard spaces through pen-tablets and PDAs. Gerosa et al. [15] presented an action research that illustrates how developers may improve coordination support in educational forums, providing summary information in reports and notifications through PDAs and mobile devices. Action research takes place in a real environment such as ITAE courses and the AulaNet LMS. Virvou and Alepis [5] described a mobile authoring tool named Mobile Author. This tool retains the high quality of the educational application in terms of interactivity, adaptability, and personalization, providing a good solution for non-computer experts to create cost-effective, high quality tutoring applications. Licea et al. [16] describe a platform that supports the development of small and middle size mobile and wireless information systems for handheld devices called MADEE (Mobile Application Development and Execution Environment). Sung et al. [17] developed a system using a distributed mobile device architecture for rapid prototyping of wireless mobile multi-user applications for use in classroom settings called Mobile-IT Education (MIT.EDU). MIT.EDU is a stable and accessible system that combines inexpensive, commodity hardware, a flexible sensor/peripheral interconnection bus and a powerful, light-weight distributed sensing, classification and inter-process communications software architecture to facilitate the distribution of distributed real-time multi-modal and context-aware applications.

Idrus and Ismail [18] discuss the roles of institutions of higher learning in employing their expertise in the teaching and learning processes, modular instruction, learning styles, and content. The results presented in this article highlight the importance of mobile learning technologies in order to clearly support the transmission and delivery of rich multi-media content. Ghiani et al. [19] proposed a multi-device, location-aware guide supporting museum visits, which also provides the possibility of enriching the museum visits through an individual or collaborative game. The project used a previously existing application named Cicero for mobile devices. The new approach for mobile guides exploits RFID technology to enable
location-awareness features and games, together with large shared screens for supporting social interaction and learning. Monahan et al. [20] developed a desktop system and web-based multi-user environment named CLEV-R. The system uses virtual reality to mimic a real university and provides an interface to a general e-learning system.

Chieu and Zeng [21] describe a web-based system for creating, maintaining, and publishing a database comprising news in the form of text and graphical images. The proposed system is designed to enable non-technical users on computer programming to add new material to a website or to modify the existing content. Staccini et al. [22] developed the j@lon platform, an authoring tool based on the Zope framework. j@lon implements the native and custom features of the Plone content management system. The main goal of j@lon is to involve to teachers in developing educational content. Wang and Trigano [23] present a psycho-pedagogical method for the generation of online courses using netUniversité. Furthermore, an interactive user questionnaire, integrating the results of lecture and quantitative research, is presented as a part of the user model. Chen and Huang [24] present the framework of a mobile knowledge management learning system that encourages learners to acquire, store, share, apply, and create knowledge. This framework was tested on a cell phone, a PDA, and a laptop. The result was that the use and quality of the system is better on devices with larger screens. Huang et al. [25] implemented an Interactive Service Module to assist teachers by receiving feedback from students via a mobile device. The proposed system enhances the feedback mechanism and implements an enhanced model for achieving mobile interaction in a synchronous learning environment. The model is called Interactive Service Module, which enables interactions among teachers and students via short message delivery.

Romero et al. [26] discuss the importance of Mobile learning and testing is emerging as a potential educational environment. This article presents a case study for using Mobile and Web-Based Computerized Tests to Evaluate University Students. Authors describe an authoring tool to develop adaptable and adaptive computerized tests that can be executed on different platforms such as personal computers, personal digital assistants, and mobile phones. Shee and Wang [27] proposed a multi-criterion methodology based on the learner satisfaction perspective in order to support those evaluation-based activities taking place at the pre- and post-adoption phases of the web-based e-learning system (WELS) life cycle. Meo et al. [28] presented X-Learn, an XML-based multi-agent system for supporting e-learning activities. X-Learn operates with three types of agents: (1) a User-Device Agent, that handles an e-learning session carried out by a user; (2) a Skill Manager Agent, that helps a user determine the skills and the subjects to study; and (3) a Learning Program Agent, that generates personalized learning programs for a specific user.

These initiatives suffer from several drawbacks, such as: (a) they require prior knowledge on how to behave in each situation; (b) they do not use advanced capabilities of rich Internet application technologies; and (c) they lack effective multi-device interfaces for displaying educational content. These deficiencies can be improved by: (a) using features of LMS and CMS, such as version management, assessment tools and content management tools, among others; (b) employing friendly and easy-to-use multi-device interfaces which are necessary in order to improve the user’s experience.

### Athena: Architecture and Functionality

Our hybrid content management system called Athena has a layered design in order to organize its components. This layered design allows scalability and easy maintenance because its tasks and responsibilities are distributed.

#### Architecture Description

The general architecture is shown in Figure 1. Each component has a function explained as follows:

- **Data layer**: This layer stores educational content; additionally it contains all the configuration tables allowing the operation of the modules and services offered by Athena. There is also information about each template, including pages, sections, and the distribution of content in each section of a page.
- **Data Access Layer**: A database management system is located in this layer which maintains the data persistence by executing insert, update, delete, and query operations within Athena’s architecture. These operations are encapsulated in this layer in order to provide security to the upper layers and avoid shortcuts on the database management system.
- **Data Management layer**: This layer communicates with the Data Access layer and it contains the representation of flat objects of each relationship of the database management system. Additionally, it provides a number of entities with read/write operations for each relationship which is executed using the Data Access layer.
- **Transformation Layer**: This layer is responsible for transforming user content, and it builds a portable application of two different types: (1) a Web-based application; and (2) an application for mobile devices. The Transformation Layer performs seven main tasks which are described below: (1) receiving all user requests and identifying whether this request is for the generation of a portable application. If the request service is different, it sends the request directly to the service layer; (2) analyzing the request for content transformation; (3) building the configuration file to export the generated application, including information that the user has provided; (4) choosing the type of technology used for building the application based on the type of device chosen by the user; (5) searching for the possible incompatibilities between the content and the technology selected by Athena; (6) analyzing conflicts and replacing any inconsistencies found with the best solution provided in Athena’s repository. This task is also responsible for notifying users of possible changes in the content; (7) making the necessary changes in order for the content to be converted to a native application for use on a mobile device or Web-based application.
- **Service Layer**: This layer provides a set of services (modules) offered by Athena. These services represent different functions such as (1) registry; (2) elimination; (3) search; and (4) meta-information. This layer has access to the Template Engine Layer in order to obtain the configuration information for the active service in order to display it on the template. This layer communicates with the Data Management Layer to obtain the content at any given time, but it is also responsible for setting the template via

#### Multi-Q2-Device Educational Content

This layer stores educational content; additionally it contains all the configuration tables allowing the operation of the modules and services offered by Athena. There is also information about each template, including pages, sections, and the distribution of content in each section of a page.

#### Data Access Layer

A database management system is located in this layer which maintains the data persistence by executing insert, update, delete, and query operations within Athena’s architecture. These operations are encapsulated in this layer in order to provide security to the upper layers and avoid shortcuts on the database management system.

#### Data Management Layer

This layer communicates with the Data Access layer and it contains the representation of flat objects of each relationship of the database management system. Additionally, it provides a number of entities with read/write operations for each relationship which is executed using the Data Access layer.

#### Transformation Layer

This layer is responsible for transforming user content, and it builds a portable application of two different types: (1) a Web-based application; and (2) an application for mobile devices. The Transformation Layer performs seven main tasks which are described below: (1) receiving all user requests and identifying whether this request is for the generation of a portable application. If the request service is different, it sends the request directly to the service layer; (2) analyzing the request for content transformation; (3) building the configuration file to export the generated application, including information that the user has provided; (4) choosing the type of technology used for building the application based on the type of device chosen by the user; (5) searching for the possible incompatibilities between the content and the technology selected by Athena; (6) analyzing conflicts and replacing any inconsistencies found with the best solution provided in Athena’s repository. This task is also responsible for notifying users of possible changes in the content; (7) making the necessary changes in order for the content to be converted to a native application for use on a mobile device or Web-based application.
Figure 1. Athena™️ Architecture. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]
the management service when the user or the same service requires it.

- **Template Engine Layer:** This layer allows changes to the presentation mode of an educational application, as well as its distribution and structure. Furthermore, it manipulates new templates designed and provided by third parties, following the structure of templates that Athena uses. In order to achieve this, Athena provides a wizard to facilitate the task.

- **Integration Layer [API]:** It allows the creation of new educational applications through a series of public interfaces which provide easy access to a set of services provided by Athena. Services compositions are presented and defined in this layer.

- **Presentation Layer:** In this layer, Athena determines the best way to display the educational content by using XHMTL when HTML5 is not supported. The Presentation Layer does not know what events are taking place inside Athena and how the services are provided, it only uses them to show the end-user interface.

According to the emphasis on automation, Athena’s architecture can be accessed in two modes of interaction: by using a basic or advanced wizard. The basic wizard is aimed at non-experienced computer users. This wizard guides users through a series of steps to develop an educational application. On the other hand, the advanced wizard is aimed at experienced computer users.

**Component Description**

In Athena’s architecture, each component has a function which is explained as follows:

- **GUI component.** This component represents the user interface. It is responsible for managing the interaction between the user and the Athena manager. The GUI component is responsible for capturing user events and controlling functions for the validation of input data. Furthermore, this component is responsible for processing and managing user requests via HTTP-based protocol. This component includes the modules for the basic and advanced wizards mentioned in the previous subsection. It also includes the H-CMS module and the mobile wizard, both modules to guide the user in content management.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>API Libraries Used for Athena in Order to Develop Multi-Device Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>API Library</strong></td>
<td>Description</td>
</tr>
<tr>
<td>PhoneGap</td>
<td>PhoneGap is an open source development tool developed by Nitobi to act as a bridge between Web applications and mobile devices. PhoneGap allows users to ignore the details of each SDK and develop applications in a consistent fashion [29,30].</td>
</tr>
<tr>
<td>Rhodes</td>
<td>Rhodes is an open source Ruby-based framework to rapidly build native apps for all major smartphone operating systems, such as iPhone, Android, RIM, Windows Mobile, and Windows Phone 7. These are true native device applications which work with synchronized local data and take advantage of device capabilities such as GPS, PIM contacts and calendar, camera, native mapping, push, barcode, signature capture, Bluetooth, and Near Field Communications (NFC) [31].</td>
</tr>
<tr>
<td>Appcelerator Titanium</td>
<td>Appcelerator Titanium is a platform for developing mobile applications, tablet and desktop applications using existing Web skills such as JavaScript, HTML, CSS, Python, Ruby, and PHP [32]. Titanium applications are divided into four main parts: (1) the html/css/javascript code that makes up the core application logic and UI; (2) the APIs that access native device/desk functionality, analytics or other modular functionality; (3) the language-OS bridge that compiles code into native application code; and (4) the run-time shell that packages the application for cross-platform distribution.</td>
</tr>
<tr>
<td>QuickConnectFamily</td>
<td>QuickConnectFamily is a mobile application development framework that facilitates the development of cross-platform applications using HTML, CSS, and JavaScript. The QuickConnectFamily templates allow complete access to behavior normally restricted to native applications; for instance, full database access across all the supported platforms. Each library contains specific functions that simplify the addition of a set of features for the mobile application development [33].</td>
</tr>
</tbody>
</table>

The first module allows the creation of chat rooms to keep in contact with the users while accessing the content through Athena, in addition to questionnaires that assess the knowledge gained after using the educational application. The second module allows the users to create a portable educational application for the mobile device or operating system that the user selects.

- **Request Analyzer Component.** This component is responsible for identifying all user requests and sending them to the Service Selector Component.

- **Service Selector Component.** This component includes all services provided by Athena for content management (e.g., the service for exporting content via the Athena manager) and has direct access to the API libraries in order to execute the request made for the Analyzer Request Component.

- **Builder Response Component.** This component receives the responses coming from the Transformation Layer. This component processes two kinds of responses: (1) direct information from the database; or (2) a packaged file containing the application generated for a mobile device or web-based modality. In both cases, the component is responsible for processing information and sending it to the user.
analyzing and verifying that the source code follows the syntax rules. Subsequently, the source code generated in high-level programming language is translated into the device code that each technology requires.

**Application Packager.** This component is responsible for receiving the source code and compiling and packaging it into a file that contains the application source code, the API libraries per technology, and all the application resources such as images, audio, video, text, and flash files. The packaged file extension depends on the type of operating system for which the application was generated. For instance, if an Android-based application is developed, the result is a file with an .apk extension or .ipa extension in the case of an iOS-based application.

**Source code generator for web-based application.** This component is responsible for managing the set of content delivered via the Athena manager and generating a web-based application. This component has the following modules, as explained below:

1. **Compiler Packager.** This module creates a copy of the template and pages selected by the user through the basic or advanced wizard. Subsequently, it is responsible for placing the corresponding tags on each page to permanently integrate them into the content.

2. **Application Packager.** This component receives the location of the new template and generates a zip file. The result of unzipping this file is an HTML-based portable application that can be easily distributed. The content is distributed in a set of folders and sub-folders.

**Service Set.** This component includes a set of services for content management via Athena. These services are described below:

- **Drag&Drop Content Manager:** The content management is executed using the drag and drop functionality through a user interface in which the user determines the position of each multimedia element in the selected template of the educational application. This functionality uses the DropZone component of the WARP Framework that uses PHP, JavaScript, and AJAX for handling both the drag and drop events and the effects on the database. WARP is a PHP-based framework for agile Web development. WARP incorporates a JavaScript-based framework called WARP JS, which is used for developing Rich Internet Applications.

- **HTML5-based Portable Application:** This service is responsible for generating an HTML-based portable application, so it only requires a browser and it does not depend on a database connection. In order to do this, the component transforms the content files stored in the repository to HTML via the Template Manager in order to identify the available sections in the template. By means of a read/write process, this component embeds the content to a temporary file. For this activity, the **TextoToFile** module is used and the **ZIPFile** module packages the generated application.

- **Template Manager:** This service is responsible for editing the available templates provided by the Athena manager. This component handles the following tasks: (1) displaying the active template automatically when the user makes a request without an ID to Athena; (2) deploying the Athena management panel; (3) displaying a list of available templates on the content manager; (4) setting a template as an asset; (5) importing a template; (6) exporting a template; (7) removing a template from the content manager; (8) building the structure for creating a template (folder structure and files ready for editing and creating a new template); (9) displaying the selected template in the content manager for use with a basic or advanced wizard; (10) creating a backup of the selected template in order to generate an HTML-based portable application; and (11) creating a backup of the selected template to generate a multi-device application.

**Service Configuration.** This module provides the configuration of the service required by the Service Set component. The configuration indicates the path to the resources and tasks to be performed.

**Template Engine.** This component stores the PHP-based source code and HTML-based content separately. The template engine is responsible for analyzing a template and determining the specific areas in which to dynamically place the content.

**Plain Objects Component.** This module is responsible for representation through the data objects used by Athena in order to operate the content management. A plain object is an object that has no application logic and it is used only to represent data as a POJO (Plain Old Java Object).

**Entities Component.** This component includes the entities from the Athena database, which separate the CRUD operations from its representation at the object level. CRUD stands for Create Read Update Delete (Create, Read, Update, Delete) used to refer to the basic functions in the database or persistence layer of a software system.

**PDO—ODBC component.** This component includes a PDO_ODBC driver. This driver implements the PHP Data Objects (PDO) interface to enable access from PHP to databases through ODBC drivers or through the IBM DB2 Call Level Interface (DB2 CLI) library [34].

**Workflow Description**

The workflow describes the functionality of Athena’s architecture, as described below:

1. The user makes a request to Athena using a Web browser by indicating a URL.
2. The Presentation Layer identifies the user’s request and redirects the HTTP-based request to the Integration Layer where the HTTP-based request is analyzed to determine what type of service is being requested.
3. The Integration Layer uses the Service Selector Component to determine whether the HTTP request is valid, if the service is available or a request for a service has not been requested. If a request for a service has been requested, Athena returns a list of available services in order for the user to select one.
4. If the service is valid and available, a query is carried out to recover the service identifier in the set of services provided by Athena. In the event that the service is not available or is invalid, Athena returns an error message with the failure reasons and a list of available services.
(5) If the requested service is the generation of a mobile application, the request is redirected to the Transformation Layer where it builds a file which describes the application to be generated. This file depends on the type of mobile device; it may be an APK file in the case of Android or an IPA file in the case of iOS. This file is sent to be processed by a number of frameworks and API libraries within Athena that depend on the selected development platform.

(6) API Libraries Module makes a request to the Service Layer to build an HTML5-based portable application. Once the request has been fulfilled and the application is ready, the API Libraries Module processes the generated application and transforms it into a mobile application using the appropriate API based on the platform that has been selected by the user.

(7) The service requested by the API Libraries Module accesses the Service Configuration Module for a list of resources, paths, and tasks to be performed in order to carry out the request received.

(8) Once the service configuration is obtained, Athena sends the template ID to the bottom layer (Template Engine Layer) in order to retrieve the file structures and application design.

(9) The template is sent to the Service Layer for its temporary processing; while a period of waiting is instigated in order to obtain the content to be presented in the template.

(10) The service selected via the Service Layer sends the necessary parameters to identify the content. This is carried out within the Data Management Layer in order to access the entities.

(11) The entities contain the data manipulation logic and therefore they require a request for the data stored in the corresponding flat object.

(12) By using an access method, flat objects retrieve the data and send them back to the entity to be handled.

(13) To achieve this data manipulation, the entity must execute a query through the Data Access Layer in order for the data obtained from the query execution to change the value of each flat object required.

(14) The connection to the Data Layer is achieved via the PDO with native support by PHP for accessing the database management or with ODBC, and a query provided by the entity is executed.

(15) The database management returns a RecordSet Object with data obtained from the Data Access Layer.

(16) The Data Access Layer transforms the RecordSet Object into an array of flat objects. This array is returned to the corresponding entities of the Data Management Layer.

(17) Entities are sent as objects (class instances) to the Service Layer for their integration with the corresponding template. At this point, a portable Web-based application with HTML5 and CSS3 has been generated.

(18) The portable application is returned to the Transformation Layer in order to be processed using the corresponding API library.

(19) Once the transformation has been completed, the Application Compiler Module generates the native application.

(20) Once the native application has been generated, the application is sent to the Application Packager Module to be packaged with the format and file extension corresponding to the mobile device required.

(21) The application packaged in a zip file is sent to the Response Builder Component which is responsible for processing the information and sending the zip file to the user.

(22) The user downloads the zip file in order to unpack it. The result of unpacking this zip file is a native application ready to be installed on the mobile device.

To illustrate the functionality of Athena, we describe a case study for generating a multi-device physics course that has already been implemented.

**CASE STUDY: GENERATING A MULTI-DEVICE PHYSICS COURSE**

Athena is a multi-device educational content management system that facilitates the generation of educational software by taking advantage of the ease of managing content via a CMS and incorporating the learning and assessment forms of an LMS. The main idea of Athena is to reuse content; it being only necessary to select the set of educational content to be embedded in the new application to develop and follow a series of steps provided by the Athena wizards. To explain Athena’s functionality, the following case study for generating a multi-device physics course is presented.

Let us suppose that a high school teacher has a number of content concerning the subject of matter and its properties and wants to develop an educational application that serves as a support tool in a physics course. In this context, there is a constraint: the teacher is a non-experienced computer user and therefore does not have the required skills to develop the educational applications. As a solution to this issue, Athena provides a set of graphic interfaces that allow any user to develop an educational application by using Athena’s contents and following the steps described below:

(1) The user accesses Athena via an authentication mechanism by entering a login name and password.

(2) Once the user has been authenticated, the user selects the option to create a new educational application and selects the basic wizard modality. This process is depicted in Figure 2. Here, the first step in the application design is provided when the user selects one of the structures provided by Athena. We have developed a set of 13 structures handling a combination of five sections which are: Column Header, Left Column, Center Column, Right Footer Column, and Column. Each one of these sections is a workspace where content can be added.

(3) The next step is the selection of the application’s layout. For each section, we have added a number of default layouts that are ready to be used by the user. Additionally, each layout has different styles.

(4) Once the user selects a layout, the next step is the insertion and distribution of content. This step loads the selected layout in edit mode. In this step, the user can
add contents such as text, audio, video, flash files, and images by clicking on the corresponding icons in the Athena toolbar and moving the content to any section of the Web page. This is depicted in Figure 3.

(5) Once the user has finished adding content to the current page, he or she can add new pages and more contents by clicking on the “Add Page” icon in the Athena toolbar and selecting one of the new pages available for the current design. Below this is the new selected page in edit mode and the user can proceed to insert the new content in accordance with step 4. This process can be repeated as many times as the user chooses. In this step, the user can add an item topic or subtopics by clicking on the “Add topic” button in the toolbar. Athena then shows the wizard for creating an agenda. In this wizard, the user can add a title for each item and specify the webpage to which the user will be redirected when they click on this item. In this way, the user has control over the navigation of the application being generated. The user can choose the color of the head and body of the topic or leave the current color as it is. When the topic is completed, the user clicks on the Add button, closes the wizard, and returns to the content insertion and distribution. The previously created agenda appears on all the pages of the educational application for navigation control purposes. In order to complement the content,
Athena provides a module for incorporating questionnaires as a learning evaluation tool. To add a questionnaire it is necessary to click the “add questionnaire” button on the toolbar. This section presents the wizard to create, update, and delete questions for each questionnaire. The user can choose between adding open-ended questions and closed-ended questions.

(6) Once the user has finished the content distribution process, they can generate the educational application by clicking the Zip Generate option in the Athena toolbar. The generated educational application is available in a zip file. The user unpacks the zip file and opens the index.html file to display the application. The generated application is an HTML5-based portable application.

Due to constant changes in educational content, the user needs to add new content to the generated application, and this application can be displayed on a tablet running the Android OS. In order to do this, the user accesses the Athena manager and clicks on the “import my template” option. In this section, the user clicks the browse button, chooses the location of the zip file and clicks the “next” button. The Athena manager then inspects the XML-based configuration file. If errors are not found, the content is loaded in edit mode. In this step the user can add new content. To export the generated content for an Android-based application, the user clicks on the “Build a multi-device application” icon in the toolbar and chooses the type of operating system on which the educational application will be installed. In this case, the user chooses Android OS and clicks on the “Build” button. The application is then made available in a zip file. The user unpacks the zip file and the result is an .apk file ready to be installed on a tablet running on the Android OS. This result is shown in Figure 4.

In order to validate the proper functioning of Athena, we have implemented other case studies to generate different courses for areas such as math, chemistry, and Mexican history in order to complement the skills being taught to high school students. These tests have generated portable mobile applications on the Motorola XOOM tablet, the iPad 2, and the Blackberry.

**DESIGN EVALUATION**

Literature has produced several methods aimed at assessing the quality of CMS [35] and LMS [27,33,34,36,37]; however, these methods do not offer in depth insight into the suitability of the design for the domain that Athena represents. Boeykens and Neuckermans [35] present a qualitative assessment of some CMS and LMS. This assessment is based on generic requirements (know how license, database language development), and can be applied to most schools, especially when the design and creation of graphical content are the most important. Shee and Wang [27] propose a multi-criteria methodology from the perspective of learner satisfaction, to support those evaluation-based activities taking place at the pre- and post-adoption phases of the WELS (Web-based E-Learning System) lifecycle. Ozkan et al. [36] propose a conceptual e-learning assessment model named HELAM (Hexagonal e-Learning Assessment Model). The aim of this study is to find ways of defining, evaluating, and promoting e-learning. The proposed model identifies the critical factors that influence user satisfaction in six different categories with 45 criteria to evaluate the effectiveness of e-learning. Wang [37] proposes the development of an instrument to measure the satisfaction level of learning systems. This proposal allows the user to take corrective actions on the points that have great disadvantages in order to improve student understanding. Kljun et al. [38] present an evaluation of LMS based on common features and new features that have arisen due to user needs. The goal of the analysis is to identify whether there is a pattern that influences the appearance of new characteristics of learning systems over time.

![Figure 4](https://example.com/figure4.png)  
**Figure 4** Screenshot of the Android-based educational application for a physics course. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]
Albarrak et al. [39] present an evaluation that is based on a review of the functions and tools of open source LMS and commercial LMS. This evaluation is based on the data collection of technical reports, the official LMS website, and LMS evaluation.

As can be inferred from the literature, one of the problems of CMS and e-learning system design is the difficulty of measuring their quality. It is quite difficult to evaluate in a quantitative way the accuracy of the solution provided. In this approach, the authors have decided to use a qualitative evaluation in order to measure the main aspects of the Athena design. Thus, we present a different evaluation approach from the quantitative method, but one aimed at providing a sound research style. With this aim, an evaluation approach based on a weighted matrix is presented. This assessment method has a long tradition in the software engineering and information systems literature (e.g., Refs. 40–41).

For this weighted matrix, we have proposed the evaluation of four aspects in order to assess the legitimacy of the Athena design. The aspects have been selected from the educational, pedagogical, and usability design by a team composed of a physician, a computer scientist with a pedagogical background, and a software engineer proficient in engineering design. Each aspect has an associated score with respect to a given scale defined. In order to provide a comparison, this evaluation was performed with two well-known CMS and e-learning systems: Joomla and Moodle, respectively. These management systems have been selected because they are two of the most popular and most cited platforms. The overall evaluation is the sum of the scores obtained by the three systems in the four aspects assessed. The authors wish to note that this evaluation has been made in order to provide a way in which to quantify the design proposed. The comparison with other CMS and e-learning systems is only to illustrate the benefits of Athena.

Aspect 1: Interface Design of the System

This aspect reflects whether the system is easy to use for the user. The design of the user interface is one of the most important parts of a software system, and it refers to whether the interface provided by Athena improves the usability of the application, thereby helping the user to perform tasks more easily [42]. An appropriate user interface increases the percentage of usability; on the other hand, a system with a poor interface represents a difficult to use application for an inexperienced user; it decreases both usability and user acceptance. This aspect allows the user to know the system, taking into account two factors:

Factor 1: Have a set of interface operation guidelines been provided?
Factor 2: Is the software user friendly?

Aspect 2: Features Promoting Use

This aspect refers to the presence of features that promote the use of Athena such as modularity, in addition to whether or not Athena was developed with a well-defined purpose. The factor of modularity reflects whether the software contains modules where each one represents a concrete set of concepts which can be reused in other environments. Modularity increases exponentially the number of possible combinations of a system, making it much more flexible [43]. Well-defined purposes guarantee the quality of the activities carried out through them. This offers users the tools needed for a specific activity and avoids unnecessary functions that will not be used, or missing features that hinder the use of Athena, or that reduce the acceptance and use of Athena. The questions for promoting the use of Athena are:

Factor 1: Has Athena been designed using modularity principles?
Factor 2: Have Athena’s functions a well-defined purpose?

Aspect 3: Perceived Effectiveness

There are several systems that offer great potential in the management and tracking of users, but many do not meet the necessary features for the creation of learning environments [35]. For this reason, it is necessary to integrate with any other system that provides the features required. However, the system does not necessarily mean that the information will be useful to students given the fact that technological advances have influenced the educational process. Students themselves are looking for new ways to interact and learn using mobile devices. This aspect reflects whether the software is useful in the teaching–learning process. To evaluate this aspect, we have considered two factors:

Factor 1: Does the system enhance the teaching–learning process?
Factor 2: Does the system offer new features with which to visualize educational content?

Aspect 4: Accessibility

This aspect reflects whether the software implements the accessibility standards for the design of a graphical user interface. This is important as a large segment of the global population has a disability, and if the software does not provide the appropriate mechanisms for facilitating the use of the application, this means that an important user sector will be omitted [44]. This aspect reflects whether or not the software provides an easy-to-use environment for the user. To evaluate this aspect we considered two factors:

Factor 1: Is Athena easy to operate?
Factor 2: Does the graphical user interface take into consideration people with disabilities?

Once all aspects have been defined, the scales used for grading the aforementioned aspects are as follows:

- Yes: +2 points.
- Partial: +1 point.
- No: +0 points.

RESULTS

Table 2 shows the results of the aspects provided to each management system (Joomla/Moodle/Athena) and the score of these evaluations for each aspect. Below, the argumentation of the results for each aspect is discussed.
Joomla, Moodle, and Athena provide a set of guidelines for the operation of the user interface, either through user manuals, contextual help, tutorials, or videos, so the Aspect 1 Factor 1 is satisfied for all analyzed content management systems (+2 points). Additionally, the appearance of Aspect 1 Factor 2 indicates whether the managers evaluated provide a friendly environment for the user, and which, together with the operational guidelines of the user interface, provide a usable and user friendly interface. Joomla partially meets this goal (+1 point), given that its management mode is based on forms, and the installment of components can only be undertaken by users with computing skills and knowledge of Joomla management. Like Moodle, Joomla is very similar to that effect, as it requires the user to know the platform, and even though it is much easier than Joomla, it presents difficulties for the user, so for Aspect 1 Factor 2, it also receives a partial score (+1 point). Finally, Athena has been designed to provide a visual environment that is both comprehensible and similar to that of a desktop environment, in order to ensure that the user feels comfortable with the options offered, and therefore is easy to use, so it is fully compliant with Aspect 1 Factor 2 (+2 points).

Joomla is a CMS designed to manage webpage content; however, given its capacity for customization through plug-ins, it can be used for educational purposes, in addition to being implemented in a number of different domains. Therefore, it, in principle, has a definite purpose, which is web content management, users can adapt and use it to meet their needs, meaning that this CMS has lost its original purpose to that of personalization, partially meeting (+1 point) Aspect 2 Factor 2. Meanwhile, the LMS Moodle is developed for the management of educational content, such as courses, tasks, activities, among many other options, so it fully complies with Aspect 2 Factor 2 (+2 points) by providing tools for this purpose. Moreover, the Hybrid Manager System (HMS), Athena, proposed in this article is designed and developed in order to manage and create educational software, which is why it offers a number of well-defined and delimited tools for the educational sphere, in full compliance with Aspect 2 Factor 2 (+2 points).

Aspect 2 Factor 1 aims to determine whether the software being evaluated follows principles of modularity in order to discover how scalable and robust the software is. If it follows the principles of modularity, new and improved features can be easily added to the software through the insertion of modules; furthermore, software maintenance is simpler if and when this principle is fulfilled. This is an important element given the fact that this ensures that the software continues to evolve and offer an improved user experience. This is true of Joomla, Moodle, and Athena as they follow the principles of modularity and offering module support through plug-ins (+2 points).

Just as Aspects 1 and 2 seek to evaluate the software from a design perspective, Aspect 3 evaluates it in terms of effectiveness during the teaching–learning process. In terms of Aspect 3 Factor 1, both Athena and Moodle are of use during the teaching–learning process, as they are both designed for this purpose given the fact that they offer the tools necessary to facilitate learning (+2 points). For Joomla, it is possible to personalize it for the teaching environment, but it is not designed for the purpose and, as such, only partially fulfills this factor (+1 point).

Like Aspect 3 Factor 1, Factor 2 evaluates how the teaching–learning process is benefited, through new educational content visualization features, such as, for example, the use of tablets, mobile devices, such as smartphones, or digital televisions. Both Joomla and Moodle offer content customization through PCs, and there are some mobile adaptations of Moodle [38]; however, the content is not optimized for mobile devices and constitute specific developments which are not “regularly” available on the platform. This is why they do not fulfill this factor (0 points). Athena, on the other hand, incorporates the generation of multi-device software, including tablets, mobile devices, such as smartphones, and digital television, which is why it fulfills this factor (+2 points).

Finally, an evaluation of whether the software complements accessibility to its user interface was undertaken. Generally speaking, it was evaluated whether the graphic user interface is apt for all types of users, no matter whether they are disabled or not. Aspect 4 Factor 1 evaluates the ease of use for the end user. This is true of Athena, which incorporates the best usability features from Joomla, Moodle, and other CMS and LMS (+2 points), offer intuitive navigation via a keyboard and other devices, such as a mouse. The use of Joomla is influenced by the fact that it is not designed for the teaching–learning proving and its navigation is not the best; however, it does offer an acceptable level of usability (+1 point). Finally, Moodle offers adequate navigation, but for users who are not
familiar with the platform, it is complicated to interact intuitively, which is why the program only partially fulfills this factor (+1 point). Furthermore, Aspect 4 Factor 2 takes into consideration the inclusion of mechanisms which provide an accessible means of interaction for users, such as, for example, the use of subtitles on videos, which offer users with hearing problems the opportunity to understand the video being played. Both Joomla and Moodle leave this task to the content creator, who must embed the video subtitles (0 points), while Athena offers a wizard to help incorporate separate subtitle files or create them without the creator of the educational content having to use conversion tools or video authoring in order to create audiovisual content with subtitles; the same is also true for audio tracks or animations (+2 points). Furthermore, additional aspects, such as color metrics, contextual help, and keyboard shortcuts are all included in Athena, unlike Joomla or Moodle which do not offer these services and in which it is necessary for these features to be developed individually in each new version of this educational software.

**DISCUSSION**

In order to analyze the closest works and compare them with our proposal, we have defined a comparative table (Table 3) which summarizes relevant aspects covered by Athena. This table also presents the similarities of the more related works in comparison with Athena.

These initiatives have deficiencies with respect to our proposal such as: (a) some proposals do not include the content management, (b) some proposals are not focused on the development of mobile applications or they are only focused on developing applications for mobile phones. In this case, the use of other devices like tablets is missed. Athena tries solving the aforementioned deficiencies providing support for different frameworks in order to develop multi-device educational content.

Although in this article Athena focuses on the education field, Athena can be used for the generation of commercial or business content. For example, the creation and management of training courses for medicine, industrial services, to mention but a few. In this sense, Athena allows to rapidly build a course according to some needs contributing in decreasing the costs involved in the instructional design and the teaching process. Additionally, Athena has the feature of generating Web-based applications; this feature is not considered in comparison with the discussed work. This generation process is similar to the process of multi-device application development, but the result is a Web-based application that can be independently used with the possibility of including new content when it is required. In this way, the user can choose between a multi-device and Web-based application development.

**FUTURE DIRECTIONS**

In terms of future directions, we are considering adding new features, such as the generation of applications for digital interactive television (T-learning) through the generation of educational software for television by using the Google platform called “Google TV” which integrates Google’s Android operating system and the Linux version of Google Chrome browser to create an interactive television overlay on top of existing internet television and WebTV sites to add a user interface. Likewise, we have considered the use of JavaFX platform. JavaFX is an expressive rich client platform for creating and delivering rich Internet experiences to all screens. JavaFX applications can run in several execution environments such as the web, desktop, mobile, and TV. JavaFX is integrated with a Java platform which takes advantage of its performance and ubiquity to implement a productive and collaborative developer-designer workflow. Although most students have a mobile device, almost all of them have a television in their homes. As a result of this, we are developing a set of JavaFX components offered by Athena for use through the TV in order to convert it into a learning environment, allowing the student to interact directly and thus stop being a spectator.

**CONCLUSIONS**

The proposed architecture provides the basis for the development of a hybrid management system called Athena. This proposal provides the foundations for generating new and better content management systems in order to provide new features for facilitating the generation and development of educational software to end users. Athena allows reusing the content and reducing development time of new educational applications. The functionality to export the contents through Athena, it allows to view educational content on multiple heterogeneous devices.
end-user devices. Athena is based on a device-independent application model to increase the availability of new types of interaction platforms for non-experienced computer users. Furthermore, Athena borrows the main features of both CMS and LMS in order to manage and reuse educational content. Finally, we believe Athena tries solving the management and reuse issues of educational content on different formats, styles, and designs.

ACKNOWLEDGMENTS

This work was supported by the General Council of Superior Technological Education of Mexico (DGEST). Additionally, this work was sponsored by the National Council of Science and Technology (CONACYT) and the Public Education Secretary (SEP) through PROMEP. Additionally, this work was supported by the Spanish Ministry of Industry, Tourism, and Commerce under the projects TRAZAMED (IPT-090000-2010-007), ENERFICIENCY (TSI-020400-2011-56), GECALLIA (TSI-020100-2011-244), and SEMOSA (TSI-020400-2011-51). It was also supported by the Spanish Ministry of Science and Innovation under the project FLORA (TIN2011-27405).

REFERENCES


Dear Author,

During the copyediting of your paper, the following queries arose. Please respond to these by annotating your proofs with the necessary changes/additions.

- If you intend to annotate your proof electronically, please refer to the E-annotation guidelines.
- If you intend to annotate your proof by means of hard-copy mark-up, please refer to the proof mark-up symbols guidelines. If manually writing corrections on your proof and returning it as a scanned pdf via email, do not write too close to the edge of the paper. Please remember that illegible mark-ups may delay publication.

Whether you opt for hard-copy or electronic annotation of your proofs, we recommend that you provide additional clarification of answers to queries by entering your answers on the query sheet, in addition to the text mark-up.

<table>
<thead>
<tr>
<th>Query No.</th>
<th>Query</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Please provide authors’ biographies and photographs.</td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>A running head short title was not supplied; please check if this one is suitable and, if not, please supply a short title of up to 45 characters that can be used instead.</td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>Please check the change made.</td>
<td></td>
</tr>
<tr>
<td>Q4</td>
<td>Author names do not match with the reference number “18”, check.</td>
<td></td>
</tr>
<tr>
<td>Q5</td>
<td>Color figures will appear in B&amp;W in print and in color in online. If they are willing to pay for color processing in print, they should contact <a href="mailto:lespinet@wiley.com">lespinet@wiley.com</a> for a price quote.</td>
<td></td>
</tr>
<tr>
<td>Q6</td>
<td>Author name does not match with the reference number “36”, check.</td>
<td></td>
</tr>
<tr>
<td>Q7</td>
<td>Please check the change made.</td>
<td></td>
</tr>
<tr>
<td>Q8</td>
<td>Please provide the book title for Reference [7].</td>
<td></td>
</tr>
<tr>
<td>Q9</td>
<td>Please provide the book title for Reference [8].</td>
<td></td>
</tr>
<tr>
<td>Q10</td>
<td>Please provide the book title for Reference [12].</td>
<td></td>
</tr>
<tr>
<td>Q11</td>
<td>Please provide the book title for Reference [14].</td>
<td></td>
</tr>
<tr>
<td>Q12</td>
<td>Please provide the Editors for Reference 31.</td>
<td></td>
</tr>
<tr>
<td>Q13</td>
<td>Please provide the publisher name for Reference [33].</td>
<td></td>
</tr>
</tbody>
</table>
USING E-ANNOTATION TOOLS FOR ELECTRONIC PROOF CORRECTION

Required Software

Adobe Acrobat Professional or Acrobat Reader (version 7.0 or above) is required to e-annotate PDFs. Acrobat 8 Reader is a free download: http://www.adobe.com/products/acrobat/readstep2.html. For help with system requirements, go to: http://www.adobe.com/support/.

Once you have Acrobat Reader on your PC and open the proof, you will see the Commenting Toolbar (if it does not appear automatically go to Tools>Commenting>Commenting Toolbar). If these options are not available in your Adobe Reader menus then it is possible that your Adobe version is lower than 7 or the PDF has not been prepared properly.

PDF Annotations (Adobe Reader version 7 or 8) – Commenting Toolbars look like this:

(PC, Adobe version 7)

(PC, Adobe version 8, right-click on title bar (Comment & Markup) to show additional icons)

(Mac)

PDF Annotations (Adobe Reader version 9)

If you experience problems annotating files in Adobe Acrobat Reader 9 then you may need to change a preference setting in order to edit.

The default for the Commenting toolbar is set to ‘off’ in version 9. To change this setting select ‘Edit | Preferences’, then ‘Documents’ (at left under ‘Categories’), then select the option ‘Never’ for ‘PDF/A View Mode’. (the Commenting toolbar is the same as in version 8).
TO INDICATE INSERT, REPLACE, OR REMOVE TEXT

- **Insert text**

Click the ‘Text Edits’ button on the Commenting toolbar. Click to set the cursor location in the text and simply start typing. The text will appear in a commenting box. You may also cut-and-paste text from another file into the commenting box. Close the box by clicking on ‘x’ in the top right-hand corner. It can be deleted by right clicking (for the PC, ctrl-click on the Mac) on it and selecting ‘Delete’.

- **Replace text**

Click the ‘Text Edits’ button on the Commenting toolbar. To highlight the text to be replaced, click and drag the cursor over the text. Then simply type in the replacement text. The replacement text will appear in a commenting box. You may also cut-and-paste text from another file into this box. To replace formatted text (an equation for example) please Attach a file (see below).

- **Remove text**

Click the ‘Text Edits’ button on the Commenting toolbar. Click and drag over the text to be deleted. Then press the delete button on your keyboard. The text to be deleted will then be struck through.

**HIGHLIGHT TEXT/MAKE A COMMENT**

Click on the ‘Highlight’ button on the commenting toolbar. Click and drag over the text. To make a comment, double click on the highlighted text and simply start typing.

**ATTACH A FILE**

Click on the ‘Attach a file’ button on the commenting toolbar. Click on the figure, table or formatted text to be replaced. A window will automatically open allowing you to attach a file. To make a comment, go to ‘General’ and then ‘Description’ in the ‘Properties’ window. A graphic will appear indicating the insertion of a file.

**LEAVE A NOTE/COMMENT**

Click on the ‘Note Tool’ button on the commenting toolbar. Click to set the location of the note on the document and simply start typing. **Do not use this feature to make text edits.**

**REVIEW**

To review your changes, click on the ‘Show’ button on the commenting toolbar. Choose ‘Show Comments List’. Navigate by clicking on a correction in the list. Alternatively, double click on any mark-up to open the commenting box.

**UNDO/DELETE CHANGE**

To undo any changes made, use the right click button on your mouse (for PCs, Ctrl-Click for Mac). Alternatively click on the ‘Edit’ in the main Adobe menu and then ‘Undo’. You can also delete edits using the right click (Ctrl-Click on the Mac) and selecting ‘Delete’.

**SEND YOUR ANNOTATED PDF FILE BACK TO WILEY VIA lespinet@wiley.com**

Save the annotations to your file and return as an e-mail. Before returning, please ensure you have answered any questions raised on the Query form that you have inserted all the corrections: later inclusion of any subsequent corrections cannot be guaranteed.

**Note:** Comprehensive instructions are provided within your PDF file: to access these instructions please click on the Comments and Markup menu in the main tool bar, or click on Help.
Electronic proof checklist, Computer Applications in Engineering Education

***IMMEDIATE RESPONSE REQUIRED***
Please follow these instructions to avoid delay of publication.

☐ READ PROOFS CAREFULLY
• This will be your only chance to review these proofs.
• Please note that the volume and page numbers shown on the proofs are for position only.

☐ ANSWER ALL QUERIES ON PROOFS (Queries for you to answer are attached as the last page of your proof.)
• Mark all corrections directly on the proofs. Note that excessive author alterations may ultimately result in delay of publication and extra costs may be charged to you.

☐ CHECK FIGURES AND TABLES CAREFULLY (Color figures will be sent under separate cover.)
• Check size, numbering, and orientation of figures.
• All images in the PDF are downsampled (reduced to lower resolution and file size) to facilitate Internet delivery. These images will appear at higher resolution and sharpness in the printed article.
• Review figure legends to ensure that they are complete.
• Check all tables. Review layout, title, and footnotes.

RETURN ☐ PROOFS
☐ CTA (If you have not already signed one)

RETURN WITHIN 48 HOURS OF RECEIPT VIA E-MAIL

QUESTIONS?
Laura Espinet, Associate Production Manager
E-mail: lespinet@wiley.com
Refer to journal acronym and article production number (i.e., CAE 00-001 for Computer Applications in Engineering Education ms 00-001).
COPYRIGHT TRANSFER AGREEMENT

Date: __________________ Contributor name: __________________

Contributor address: _____________________________________________

Manuscript number (Editorial office only): ____________________________

Re: Manuscript entitled ____________________________________________

____________________________________________________________________

____________________________________________________________________

____________________________________________________________________

for publication in ___________________________________________________

____________________________________________________________________

____________________________________________________________________

Dear Contributor(s):

Thank you for submitting your Contribution for publication. In order to expedite the editing and publishing process and enable Wiley-Blackwell to disseminate your Contribution to the fullest extent, we need to have this Copyright Transfer Agreement signed and returned as directed in the Journal's instructions for authors as soon as possible. If the Contribution is not accepted for publication, or if the Contribution is subsequently rejected, this Agreement shall be null and void. Publication cannot proceed without a signed copy of this Agreement.

A. COPYRIGHT

1. The Contributor assigns to Wiley-Blackwell, during the full term of copyright and any extensions or renewals, all copyright in and to the Contribution, and all rights therein, including but not limited to the right to publish, republish, transmit, sell, distribute and otherwise use the Contribution in whole or in part in electronic and print editions of the Journal and in derivative works throughout the world, in all languages and in all media of expression now known or later developed, and to license or permit others to do so.

2. Reproduction, posting, transmission or other distribution or use of the final Contribution in whole or in part in any medium by the Contributor as permitted by this Agreement requires a citation to the Journal and an appropriate credit to Wiley-Blackwell as Publisher, and/or the Society if applicable, suitable in form and content as follows: (Title of Article, Author, Journal Title and Volume/Issue, Copyright © [year], copyright owner as specified in the Journal). Links to the final article on Wiley-Blackwell's website are encouraged where appropriate.

B. RETAINED RIGHTS

Notwithstanding the above, the Contributor or, if applicable, the Contributor's Employer, retains all proprietary rights other than copyright, such as patent rights, in any process, procedure or article of manufacture described in the Contribution.

C. PERMITTED USES BY CONTRIBUTOR

1. Submitted Version. Wiley-Blackwell licenses back to the Contributor the following rights with respect to the final published version of the Contribution:

a. After publication of the final article, the right to self-archive on the Contributor's personal website or in the Contributor's institution/employer's institutional repository or archive. This right extends to both intransits and the Internet. The Contributor may not update the submission version or replace it with the published Contribution. The version posted must contain a legend as follows: This is the pre-peer reviewed version of the following article: FULL CITE, which has been published in final form at [Link to final article].

b. The right to transmit, print and share copies with colleagues.

2. Accepted Version. Re-use of the accepted and peer-reviewed (but not final) version of the Contribution shall be by separate agreement with Wiley-Blackwell. Wiley-Blackwell has agreements with certain funding agencies governing reuse of this version. The details of those relationships, and other offerings allowing open web use, are set forth at the following website: http://www.wiley.com/go/funderstatement. NIH grantees should check the box at the bottom of this document.

3. Final Published Version. Wiley-Blackwell hereby licenses back to the Contributor the following rights with respect to the final published version of the Contribution:

a. Copies for colleagues. The personal right of the Contributor only to send or transmit individual copies of the final published version in any format to colleagues upon their specific request provided no fee is charged, and further-provided that there is no systematic distribution of the Contribution, e.g. posting on a list serve, website or automated delivery.

b. Re-use in other publications. The right to re-use the final Contribution or parts thereof for any publication authored or edited by the Contributor (excluding journal articles) where such re-used material constitutes less than half of the total material in such publication. In such case, any modifications should be accurately noted.

c. Teaching duties. The right to include the Contribution in teaching or training duties at the Contributor's institution/place of employment including in course packs, e-reserves, presentation at professional conferences, in-house training, or distance learning. The Contribution may not be used in seminars outside of normal teaching obligations (e.g. commercial seminars). Electronic posting of the final published version in connection with teaching/training at the Contributor's institution/place of employment is permitted subject to the implementation of reasonable access control mechanisms, such as user name and password. Posting the final published version on the open Internet is not permitted.

d. Oral presentations. The right to make oral presentations based on the Contribution.

4. Article Abstracts, Figures, Tables, Data Sets, Artwork and Selected Text (up to 250 words).

a. Contributors may re-use unmodified abstracts for any non-commercial purpose. For on-line uses of the abstracts, Wiley-Blackwell encourages but does not require linking back to the final published versions.

b. Contributors may re-use figures, tables, data sets, artwork, and selected text up to 250 words from their Contributions, provided the following conditions are met:

(i) Full and accurate credit must be given to the Contribution.
(ii) Modifications to the figures, tables and data must be noted. Otherwise, no changes may be made.
(iii) The reuse may not be made for direct commercial purposes, or for financial consideration to the Contributor.
(iv) Nothing herein shall permit dual publication in violation of journal ethical practices.
D. CONTRIBUTIONS OWNED BY EMPLOYER

1. If the Contribution was written by the Contributor in the course of the Contributor’s employment (as a “work-made-for-hire” in the course of employment), the Contribution is owned by the company/employer which must sign this Agreement (in addition to the Contributor’s signature) in the space provided below. In such case, the company/employer hereby assigns to Wiley-Blackwell, during the full term of copyright, all copyright in and to the Contribution for the full term of copyright throughout the world as specified in paragraph A above.

2. In addition to the rights specified as retained in paragraph B above and the rights granted back to the Contributor pursuant to paragraph C above, Wiley-Blackwell hereby grants back, without charge, to such company/employer, its subsidiaries and divisions, the right to make copies of and distribute the final published Contribution internally in print format or electronically on the Company’s internal network. Copies so used may not be resold or distributed externally. However the company/employer may include information and text from the Contribution as part of an information package included with software or other products offered for sale or license or included in patent applications. Posting of the final published Contribution by the institution on a public access website may only be done with Wiley-Blackwell’s written permission, and payment of any applicable fee(s). Also, upon payment of Wiley-Blackwell’s reprint fee, the institution may distribute print copies of the published Contribution externally.

E. GOVERNMENT CONTRACTS

In the case of a Contribution prepared under U.S. Government contract or grant, the U.S. Government may reproduce, without charge, all or portions of the Contribution and may authorize others to do so, for official U.S. Government purposes only, if the U.S. Government contract or grant so requires. (U.S. Government, U.K. Government, and other government employees: see notes at end)

F. COPYRIGHT NOTICE

The Contributor and the company/employer agree that any and all copies of the final published version of the Contribution or any part thereof distributed or posted by them in print or electronic format as permitted herein will include the notice of copyright as stipulated in the Journal and a full citation to the Journal as published by Wiley-Blackwell.

G. CONTRIBUTOR’S REPRESENTATIONS

The Contributor represents that the Contribution is the Contributor’s original work, all individuals identified as Contributors actually contributed to the Contribution, and all individuals who contributed are included. If the Contribution was prepared jointly, the Contributor agrees to inform the co-Contributors of the terms of this Agreement and to obtain their signature to this Agreement or their written permission to sign on their behalf. The Contribution is submitted only to this Journal and has not been published before. (If excerpts from copyrighted works owned by third parties are included, the Contributor will obtain written permission from the copyright owners for all uses as set forth in Wiley-Blackwell’s permissions form or in the Journal’s Instructions for Contributors, and show credit to the sources in the Contribution.) The Contributor also warrants that the Contribution contains no libelous or unlawful statements, does not infringe upon the rights (including without limitation the copyright, patent or trademark rights) or the privacy of others, or contain material or instructions that might cause harm or injury.

CHECK ONE BOX:

☐ Contributor-owned work

ATTACH ADDITIONAL SIGNATURE PAGES AS NECESSARY

Contributor’s signature Date

Type or print name and title

Co-contributor’s signature Date

Type or print name and title

☐ Company/Institution-owned work

(made-for-hire in the course of employment)

Company or Institution (Employer-for-Hire) Date

Authorized signature of Employer Date

☐ U.S. Government work

Note to U.S. Government Employees

A contribution prepared by a U.S. federal government employee as part of the employee’s official duties, or which is an official U.S. Government publication, is called a “U.S. Government work,” and is in the public domain in the United States. In such case, the employee may cross out Paragraph A.1 but must sign (in the Contributor’s signature line) and return this Agreement. If the Contribution was not prepared as part of the employee’s duties or is not an official U.S. Government publication, it is not a U.S. Government work.

☐ U.K. Government work (Crown Copyright)

Note to U.K. Government Employees

The rights in a Contribution prepared by an employee of a U.K. government department, agency or other Crown body as part of his/her official duties, or which is an official government publication, belong to the Crown. U.K. government authors should submit a signed declaration form together with this Agreement. The form can be obtained via http://www.opsi.gov.uk/advice/crown-copyright/copyright-guidance/publication-of-articles-written-by-ministers-and-civil-servants.htm

☐ Other Government work

Note to Non-U.S., Non-U.K. Government Employees

If your status as a government employee legally prevents you from signing this Agreement, please contact the editorial office.

☐ NIH Grantees

Note to NIH Grantees

Pursuant to NIH mandate, Wiley-Blackwell will post the accepted version of Contributions authored by NIH grant-holders to PubMed Central upon acceptance. This accepted version will be made publicly available 12 months after publication. For further information, see www.wiley.com/go/nihmandate.
Additional reprint purchases

Should you wish to purchase additional copies of your article, please click on the link and follow the instructions provided: https://caesar.sheridan.com/reprints/redir.php?pub=10089&acro=CAE

Corresponding authors are invited to inform their co-authors of the reprint options available.

Please note that regardless of the form in which they are acquired, reprints should not be resold, nor further disseminated in electronic form, nor deployed in part or in whole in any marketing, promotional or educational contexts without authorization from Wiley. Permissions requests should be directed to mailto: permissionsus@wiley.com

For information about ‘Pay-Per-View and Article Select’ click on the following link: http://wileyonlinelibrary.com/ppv