Authoring for Adaptive Web-Based Learning Systems: A Case Study

doi:10.3991/ijet.v4s2.920

H. Obeidat, M. Meccawy and P. Blanchfield
University of Nottingham, Nottingham, UK

***Abstract*—Adaptive Educational Hypermedia aims to provide tailored and personalised learning experiences to different students. However, one of its limitations which have resulted in its poor adoption by the e-learning community is the complexity of its authoring process. This paper tries to address this issue in order to push those adaptive systems forward. First, it presents an authoring aid for an existing adaptive system. Lessons learned from this case study are then used to provide a high-level design specification for an authoring tool with an accessible interface to ordinary teachers with limited IT-skills.**

***Index Terms*—Adaptive Educational Hypermedia, Service Oriented Architecture, WHURLE 2.0, Authoring tool**

I. INTRODUCTION

Adaptive Hypermedia (AH) [1] is an alternative to traditional hypermedia systems that aims to provide some help and guidance to users. In an educational context this is achieved through Adaptive Educational Hypermedia (AEH). AEH treats learners as distinct individuals that require personalised assistance and guidance during their online learning process. Those students could vary in terms of their abilities, learning styles, experiences, learning goals, backgrounds and many other aspects. Adaptive Educational Hypermedia Systems (AEHS) aim to overcome the lack of interpersonal interaction in online and distance learning by providing tailored learning experiences. The challenge is to make the system know and understand the learner's capabilities or goals and thus make the adaptation decisions instantly. An AEHS is able to present tailored content and navigation paths by adapting to the user model, which provides a profile that stores the distinct features of an individual user. A learning environment can be considered adaptive [2] if it is capable of achieving three tasks: 1) monitoring the learners' activities and interpreting those activities based on specific models, 2) inferring a learner's requirements and preferences from those models, and 3) acting upon this available knowledge about the learner and the subject matter to dynamically facilitate the learning process.

In AEH, the domain of a learning application can be represented as a hierarchy of concepts. The user model stores values for each concept for each user in the profile. In some cases it assumes that when a student logs into a page and reads it or passes an online test, he/she has learned the topic and the user profile is updated accordingly. Those applications also use a set of prerequisite relationships between these concepts, which together with the concept hierarchy and the concept values can be used by the system to decide if the learner is ready for the new concept. The links that are presented on each page by the system can let a user know whether it leads to a page containing knowledge that is already known, new knowledge or advanced knowledge that is not ready to be taken by this user at his/her current level [3].

Many empirical studies [4-6] have shown the added value that adaptation brings to the Web, especially in the context of e-learning. More precisely, adaptive navigational support could accelerate the user's navigation and learning while adaptive presentation can improve understanding of the content [5].

Despite the benefits and great potential of AEH, its implemented systems suffer from some disadvantages that have limited their usage. Its transitive nature according to the changes in the user model results in the complexity of the design, authoring and implementation of those systems in comparison with traditional Web-based educational systems. It is the price for getting the adaptation effects [7]. As stated by De Bra in [3], AEH it is not the magic wand that solves all the Web-based e-learning problems. Hence, good practice in learning content development is still a key element for the success of AEHS. Conlan for example in [4] stresses the importance of reusability of adaptive content given that the development of new adaptive learning content is expensive and time consuming. This is due to authoring of adaptive content being more complex and requiring higher level of IT skills than would be expected of the majority of non-IT teachers. In some cases this authoring process requires creating multiple versions of the same lesson as in the Felder-Solomon [8, 9] (visual-verbal) learning style adaptation. Hence, the authoring problem [10] is one of the key issues that is preventing AEHS from being part of main stream education. Therefore, researchers have tried to address this by either introducing new authoring tools such as NetCoach [11] or alternatively by allowing the reusability of adaptive content. This will not remove the initial authoring costs but will limit this expensive and complex content development process to a single attempt for any given course, which is known as the 'create once and use many' approach [10].

This paper follows the first approach which tries to produce a rich and easy to use authoring tool for adaptive hypermedia. It presents a case study where an authoring aid has been developed to work with an existing AEHS known as WHURLE 2.0 [12, 13] (section 2). This authoring aid (section 3) is the first step towards producing a general authoring tool (section 4) for adaptive hypermedia as this paper will proceed to demonstrate.

II. WHURLE 2.0

The advances of Web 2.0 technologies, has prompted a rethink of the implementation design of AEHS. WHURLE 2.0, which was developed in the University of Nottingham, is one of those new designs where the system's components (i.e. user modelling component, adaptation filter, lesson plan and the learning resources) become a set of modular, autonomous, independently-developed services that can communicate with each other by adhering to a specific set of protocols that have been previously agreed upon; Web services protocols [14], namely SOAP and WSDL. All the services share values of independence, interoperability and flexibility. In WHURLE 2.0, the learner is presented with a workplace rather than a single document and hence the system's Portal or delivery service provides the single login point to this learning space or environment. When a user logs into this portal, it acts as a client and communicates in the background with other system components (services) to perform the required task.

WHURLE 2.0 is composed of six main components that are explained in detail in the following sections. These are five independent Web services that collaborate with each other to tailor a unique view of the learning content for a given learner, and a delivery service where the learner views this adaptive content. Those services are the: Aggregation Service (AGS), User Modelling Service (UMS), Lesson Plan Service (LPS), Adaptation Filter Service (AFS) and the Chunk Management Service (CMS). While the delivery service for the new learning environment, was chosen to be a Learning Management System or an LMS (for example Moodle[15], Blackboard[16], ATutor[17], SAKAI[18]). The learning content is saved in conceptually discrete units called chunks which are XML files that contain text or references to other media types which are located in the CMS. Fig. 1 provides a conceptual description of WHURLE 2.0's architecture.

The AGS is the heart of WHURLE 2.0 and is both a client and a service. Invoked by Moodle as a service, AGS will then act as a client that invokes four main services before returning the results to the LMS (Moodle in this case). The UMS has an XML database which stores the users' profiles and that is queried using XSLT to match users with their level of knowledge – this is returned to the AGS. The LPS has an XML database, where all the lesson plans' (LP) names and locations are stored. The AGS passes the lesson's name to the LPS, which returns the lesson plan for that lesson. Once the AGS has those two significant arguments (user's level and lesson's plan) it passes them to a third service, the AFS. The AFS filters the lesson plan according to its rules and produces a list of the required chunks. This XML list of required chunks is then passed to the CMS. The CMS consists of an XML database which contains all the requisite information about the chunks (name, location, type, author, etc) including the meta-data provided for semantic exchange and interoperability. Some of the chunks which are classified as "internal" are stored in the CMS, which also serves as a repository for them. The list produced by the AFS is used by the CMS to select those chunks and pass them to the AGS, which will return them to Moodle. A client in Moodle is responsible for populating its tables of a given learning activity such as the Lesson activity with this learning content.

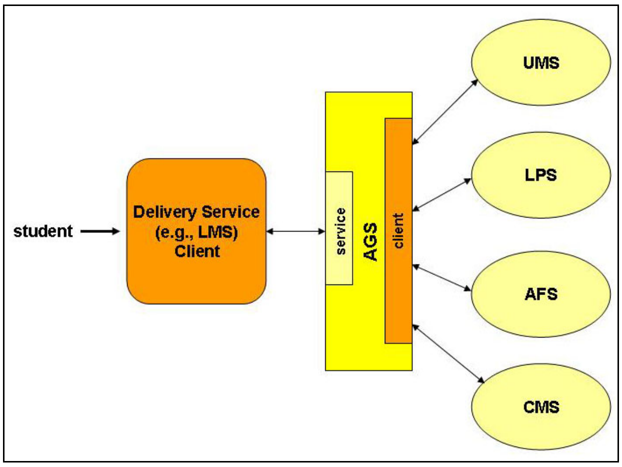


Figure 1. WHURLE 2.0 Conceptual Design

The flexibility and open nature of WHURLE 2.0 allows any new service to be added to the architecture without affecting the other services. This is feasible because of the decision to have a dedicated service for integration and traffic control which is the Aggregation Service (AGS) that acts both as a client and a service. Therefore, the newly developed authoring aid service creating chunks and lesson plans, which has been developed independently can be plugged into the system as will be explained in the following sections.

III. WHURLE 2.0AUTHOR – AN AUTHORING AID

The development of the design of WHURLE2.0Author followed the principle of service oriented architectures (SOA) which enables the integration of WHURLE 2.0Author as an authoring aid into WHURLE 2.0 as an AEHS. Web services technologies have been used as the means of communication to integrate WHURLE2.0Author as part of WHURLE 2.0. A major focus of Web services is to make functional building blocks accessible over standard Internet protocols that are independent from platforms and programming languages. Therefore, Developing WHURLE2.0Author as a Web service can also make it interoperable with other AEHS that want to produce adaptive content in the form of XML chunks.

Fig. 2 illustrates the conceptual design of WHURLE 2.0 with the addition of WHURLE2.0Author using a Web service oriented approach. It can be noted that the design of WHURLE2.0Author does not interfere with any of WHURLE 2.0's components. Rather, it co-exists with WHURLE 2.0 components by communicating with the AGS, which deals with different requests and responses and manages WHURLE 2.0 communications between its different components. Being the authoring service for WHURLE 2.0 this service communicates with the AGS either to provide authored-edited chunk and lesson plans that will be then passed to their dedicated services, or to accept chunks and lesson plans from AGS to be edited when required. Fig. 3 represents a snapshot of the developed authoring aid.

IV. WHURLE 2.0AUTHORING SERVICE: AN AUTHORING TOOL

The design of the authoring tool has focused on expanding the author role without the need for technical XML knowledge that may be unrealistic to expect from authors who are not IT-experts. Thus, with WHURLE2.0

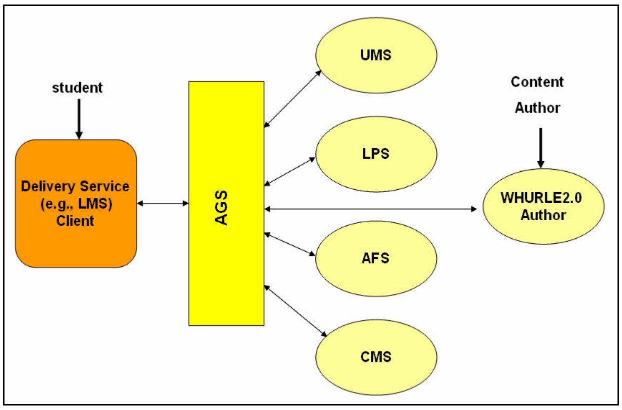
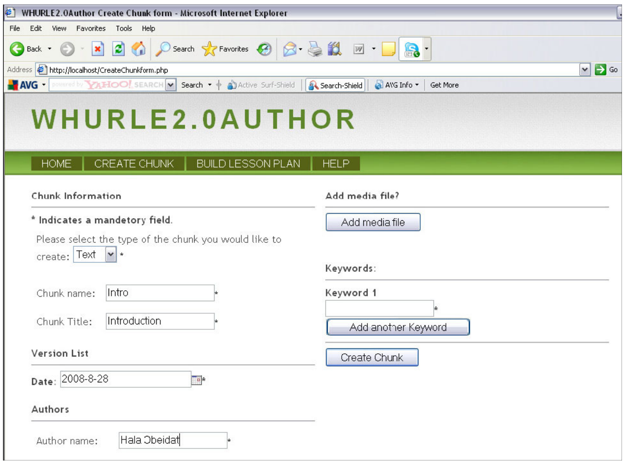


Figure 2. WHURLE 2.0 conceptual Model with WHURLE2.0Author



http://localhost/CreateChunkform.php

Chunk Information

* Indicates a mandetory field.

Please select the type of the chunk you would like to create: Text *

Chunk name: Intro *

Chunk Title: Introduction *

Version List

Date: 2008-8-28 *

Authors

Author name: Hala Obeidat *

Add media file?

Keywords:

Keyword 1 *

Figure 3. WHURLE2.0Author Authoring Aid Interface

Authoring Service (W2AS) comes the availability of a tool that can act as an author interface, and can enable authors to easily add their content of chunks to the system, thereby creating narrative (structure or story line) and building lesson plans to serve the different needs of their students. Fig. 4 demonstrates the adopted architecture of W2AS as a web service that enhances WHURLE 2.0.

The design of this service depended on a number of inputs that would graphically assist the user in creating adaptive lessons; these are user created chunks of knowledge (text, audio and video based chunks). The design of the service is also such that it provides a number of graphical processing functionalities, such as:

- A chunk editor (fig. 5): this is a basic HTML editor (basic html toolbox) for defining and editing chunks.
- A lesson plan editor (fig. 6): this is a more advanced drag and drop authoring tool with basic HTML editing tools for creating WHURLE 2.0 lesson plans and managing the chunks' positioning within lessons.

Through the chunk editor, the basic metadata required to build a chunk are captured from the author via a user-friendly form. The entry of chunk contents can be facilitated through a carefully designed chunk editor, as displayed in fig. 4. It can be seen that the chunk editor utilizes four panels: The working area panel, the chunk list panel, the resources panel and the toolbox panel.

- The working area panel was designed to contain the active chunks the author is actively creating (or editing). The three tabs (panes) that appear in the working area panel (design, code and split panes) were designed to accommodate the preferences and abilities of different authors.
- The chunk list panel was designed so that an author can easily drag a chunk from the chunk list into the main working area for editing purposes. At the same time, it was designed giving the author the flexibility of searching for his desired chunk which automatically loads into the main working area whenever the Go! button is pressed.
- The resources panel contains a tree structure of available resources of images and media that are available in the active chunk (the chunk currently available in the working area panel).
- The toolbox panel contains simple html tags that were designed to be dragged and dropped into the working area panel.

The lesson plan editor has similar panels as the chunk editor, with some additions to the working area panel. It was also designed to have an additional "levels panel" which will be discussed later.

The additions that have been made to the working area panel of the lesson plan editor can be summarised as follows: for each of the target user groups a lesson narrative is built. The progress of the lesson can be tracked by the different levels of the lesson that are represented via the levels tab. For example, the lesson plan that is illustrated in the screen in fig. 5 has two levels: Level 1 and Level 2. Each level must contain at least one page which in return should contain the chunks that construct the lesson. Pages within levels were also designed in tabs. Within a page, each chunk is displayed as an expander control (a collapsible window that displays content) with the chunk

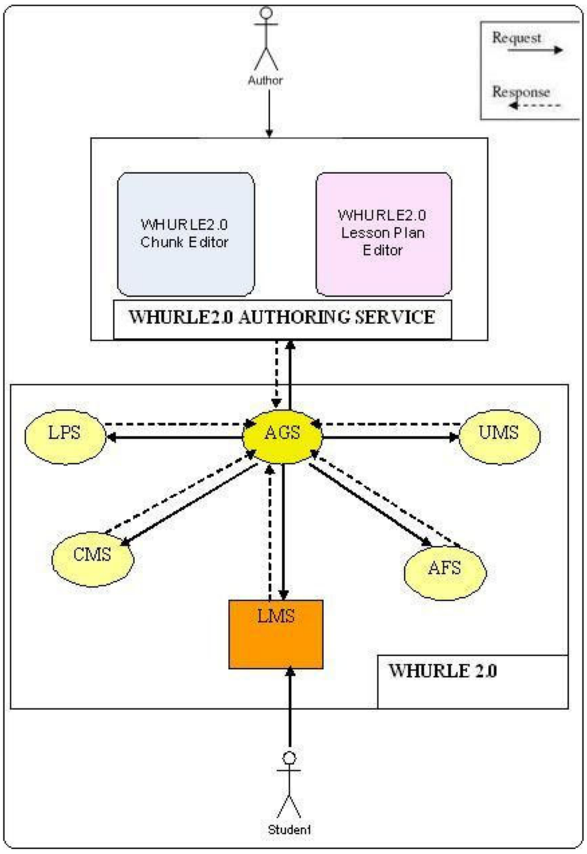
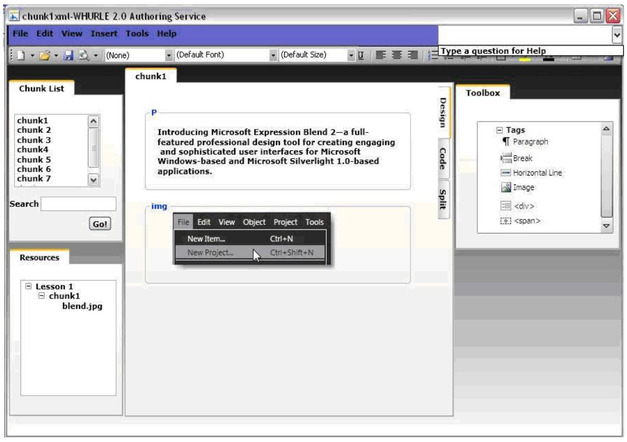
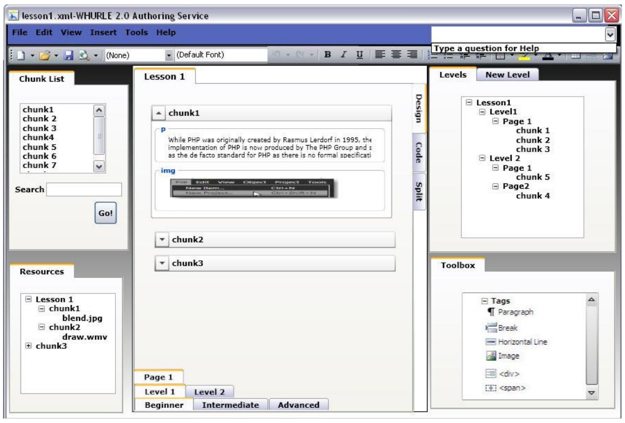


Figure 4. The adopted architecture of W2AS



Introducing Microsoft Expression Blend 2—a full-featured professional design tool for creating engaging and sophisticated user interfaces for Microsoft Windows-based and Microsoft Silverlight 1.0-based applications.

Figure 5. W2AS Chunk Editor



While PHP was originally created by Rasmus Lerdorf in 1995, the implementation of PHP is now produced by The PHP Group and s as the de facto standard for PHP as there is no formal specificati

Figure 6. W2AS Lesson Plan Editor

name being the header of the expander. The content within the expander contains an editable version of chunk content represented in group boxes as described before in the chunk editor screen. It is worth mentioning here that these chunks can be saved, or saved in different names from within the lesson plan editor. The use of expanders was suggested to allow the user to keep track of all the chunks contained within a page, and simultaneously, be able to expand one of the chunks and edit it. The design, code and split tabs were designed to increase the system usability by accommodating to the different preferences, knowledge and experiences of various authors in a way similar to that described in the chunk editor.

The final addition to the lesson plan editor was the levels panel. This panel contains a tree structure of the active lesson that is currently being edited within the working area panel. It classifies a lesson into levels that contain pages, which in return, hold the chunks that construct the lesson. This panel was provided to help the author easily track the levels, pages, and chunks while creating the lesson.

The expected outputs of the design should be XML files that contain chunks that make up adaptive lessons which are technically compatible with the WHURLE 2.0 environment.

The main purpose of using a simple html editor is that the basic component which will ultimately appear for the student (the set of chunks that appear to the student in the virtual lesson) will be much easy and less time consuming to develop since the majority of teachers have limited IT-skills. The presented drag and drop editor used for the creation of Lesson Plan is very simple to use and supports the author by enabling them to work on multiple views simultaneously.

V. DISCUSSION AND CONCLUSIONS

Adaptive Educational Hypermedia is a well established area of online learning in today's e-learning community which holds great potential for the future. It is believed that this type of rich adaptive or interactive content would have great impact for improving web-based learning [19]. It is a move from the traditional simple 'one-size-fits-all' way of learning with its simple predefined content and page- turning approach. However, this type of learning suffers from a number of limitations that resulted in its current absence. One of those limitations is the complexity of its authoring that falls beyond the capabilities of most teachers or tutors who will be using those systems to create content. Creating manual XML chunk and lesson plans files as in the case of WHURLE 2.0 is seen as a serious cumbersome especially when compared with simple HTML editors provided by LMS for examples. Therefore creating an authoring aid has helped the teacher to automatically create those XML files and is seen as the first step in hiding the complexity of the authoring process. The authoring aid was designed to be wrapped as a Web service which made it easy to integrate with the WHURE 2.0 framework which itself is a SOA-based. This indicates also the flexibility provided by those architectures hence they allow the integration of independently developed tools and services that will enrich the learning environment. Moreover, developing this authoring service provided an excellent case study where lessons were learned in order to design a generic, visual and feature-rich authoring tool which was designed to present users with an accessible tool for authoring adaptive content as well as structuring those adaptive lessons. The designed tool will continue to be developed and tested with multiple teachers to evaluate its usability and accessibility. It is the authors hope that providing such tools would help in pushing this type of e-learning system forward and gaining it wider acceptance and adoption by teachers and educators in the e-learning community.

ACKNOWLEDGMENT

The second author is a PhD student who is sponsored by the Ministry of Higher Education of The Kingdom of Saudi Arabia. The WHURLE concept was first developed by Dr Adam Moore of Nottingham University. The authors would also like to thank Dr Tim Brailsford for his helpful advice...

REFERENCES

[1] Brusilovsky, P., Adaptive Hypermedia. User Modeling and User Adapted Interaction.2001. (11): p. 87-110. (doi:10.1023/A:1011143116306)

[2] Paramythis, A., and Loidl-Reisinger, S., Adaptive Learning Environments and e-Learning Standards. Electronic Journal of e-Learning (EJEL).2004. 2 (2).

[3] De Bra, P., Adaptive Educational Hypermedia on the Web. Communication of the ACM.2002. 45 (5): p. 60-61. (doi:10.1145/506218.506247)

[4] Conlan, O., and Wade, V. Evaluation of APeLS - An Adaptive eLearning Service based on the Multi-model, Metadata-driven Approach. In: Proc.Third International Conference on Adaptive

Hypermedia and Adaptive Web-Based Systems (AH2004). Eindhoven, The Netherlands: Springer August 2004. 2004.

[5] Boyle, C., and Encarnacion, A.O., Metadoc: An adaptive hypertext reading system. User Modeling and User-Adapted Interaction.1994. 4 (1): p. 1-19. (doi:10.1007/BF01142355)

[6] Brusilovsky, P., and Pesin, L., Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. Computing and Information Technology 1998 6(1): p. 27-38.

[7] Brusilovsky, P. Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In: Murray T, Blessing, S., and Ainsworth, S., ed. Authoring Tools for Advanced Technology Learning Environment, Kluwer Academic Publishers: Dordrecht. p. 377-409 2003.

[8] Felder, R.M., and Silverman, L.K., Learning and Teaching Styles in Engineering Education. Engineering Education.1988. 78 (7): p. 674--681.

[9] Felder, R.M., and Soloman, B.A. Index of Learning Styles Available from: http://www.ncsu.edu/felder-public/ILSpage.html [cited 2008 1/8/2008].

[10] Stewart, C., Cristea, A., Brailsford, T., and Ashman, H. Authoring Once, Delivering Many: Creating Reusable Adaptive Courseware, in IASTED International Conference on Web-based Education (WBE '05): Grindelwald, Switzerland. February 21-23 2005.

[11] Weber, G., Kuhl, C-H, and Weibelzahl, S. Developing Adaptive Internet Based Courses with the Authoring System NetCoach. In: Proc.Revised Papers from the International Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity: Springer-Verlag. 2002.

[12] Meccawy, M., Blanchfield, P., Ashman, H., Brailsford, T., and Moore, A. WHURLE 2.0: Adaptive Learning Meets Web 2.0. In: Proc.EC-TEL 08: 3rd European Conference on Technology-Enhanced Learning Maastricht, The Netherlands: Springer Verlag, September 16-19. 2008.

[13] Meccawy, M. WHURLE 2.0: An Adaptive Web Service e-Learning Environment for the Web. In: Proc.Saudi Innovation Conference (SIC 2007). Newcastle upon Tyne, UK May 11-12. 2007.

[14] WSG. Web Services Glossary. [Online Document]2004 (30th March 2006). Available from: http://www.w3.org/TR/ws-gloss/ [cited 2008 May 6].

[15] Moodle. Available from: http://moodle.org/ [cited 2008 January 28].

[16] Blackboard. Available from: http://www.blackboard.com/us/index [cited 2008 January 28].

[17] ATutor. Available from: http://www.atutor.ca/ [cited 2008 January 28].

[18] SAKAI. Available from: http://sakaiproject.org/portal [cited 2008 September 8].

[19] Rey López, M., Brusilovsky, P., Meccawy, M., Díaz Redondo, R., Fernández Vilas, A., and Ashman, H, Resolving the Problem of Intelligent Learning Content in Learning Management Systems. International Journal on E-Learning.2007. 7 (3).

AUTHORS

Hala Obeidat has finished her M.Sc. degree in Management of Information Technology from the University of Nottingham (2008) and has interest in the field of Adaptive Educational Hypermedia. She has gained her B.Sc. in Computer Information Systems from Jordan University of Science and Technology in Jordan (2004) (e-mail: hxo07m@cs.nott.ac.uk).

Maram Meccawy is a Ph.D. student at the University of Nottingham, UK. She obtained a M.Sc. in Distributed Systems and Networks from the University of Hertfordshire (2003) and holds a BSc in Computer Science from King Abdul-Aziz University in Saudi Arabia (2002). She is interested in using of Adaptive Hypermedia, Learning Management systems and Web services to enhance online learning. Her interests also include applying Web 2.0 and E-learning 2.0 technologies to improve Web-based education (e-mail: mzm@cs.nott.ac.uk).

Peter Blanchfield is the Director of the Informatics Institute of Information Technology, which is part of the School of Computer Science at the University of Nottingham. His research is now developing in the use of computer games in education. Another aspect of this has been the use of computer games in the treatment of mental illness in children. He also has interest in many aspects of Web adaptation and has some remaining interest in image processing problems (e-mail: pxb@cs.nott.ac.uk).

This article was modified from a presentation at the International Conference on Interactive Mobile and Computer Aided Learning IMCL2009, April 2009, in Amman, Jordan. Manuscript received 3 May 2009. Published as submitted by the authors.