

Implementing WAI Authoring Tool Accessibility Guidelines in Developing Adaptive Elearning

Said Talhi

Computer Science LASTIC Laboratory, University of Batna, Algeria
Email: saidtalhi@gmail.com

Fairouz Khadraoui

Computer Science LASTIC Laboratory, University of Batna, Algeria
Email: khadraoui_f@yahoo.fr

Mahieddine Djoudi

Laboratoire SIC & Equipe IRMA, University of Poitiers, France
Email: mahieddine.djoudi@univ-poitiers.fr

Abstract — Adaptive learning technology allows for the development of more personalized online learning experiences with materials that adapt to student performance and skill level. The term “adaptive” is also used to describe Assistive Technologies that allow the usability of online based courses for learners with disabilities and special needs. Authoring tools can enable, encourage, and assist authors in the creation of elearning content. Because most of the content of the Web based adaptive learning is created using authoring tools, they may be accessible to authors regardless of disability and they may support and encourage the authors in creating accessible elearning content. This paper presents an authoring tool designed for developing accessible adaptive elearning. The authoring tool, dedicated to Algerian universities, is designed to satisfy the W3C/WAI Authoring Tool Accessibility Guidelines (ATAG), and to allow collaboration functionalities among teachers where building elearning courses. After presenting the W3C/WAI accessibility guidelines, the collaborative authoring tool is outlined.

Index Terms — Accessibility guidelines, Elearning, Adaptive Learning Systems, Authoring Tools, Engineering Education

I. INTRODUCTION

Adaptive elearning is an educational method which uses computers to adapt the presentation of educational material according to students' weaknesses, as indicated by their responses to questions. Adaptive learning has also been known as adaptive educational hypermedia, adaptive instruction, and intelligent tutoring.

Adaptive Learning Systems (ALS) tries to transform the learner from passive receptor of information to collaborator in the educational process. They have been designed as both desktop computer applications and web applications [1].

The ALS development remains a difficult task to undertake, despite all the efforts carried out during the last few years. This interdisciplinary task requires often the collaboration between experts from different fields such as education, psychology and computer science engineering.

In order to improve the productivity in this field and allow a wider community to be involved, authoring tools are generally used and allow the users to develop some elearning material, sometimes, without writing a single line of code.

Authoring tools can enable, encourage, and assist authors in the creation of elearning content through prompts, alerts, checking and repair functions, help files and automated tools. Thus, the task is reduced in a way that the domain authors are needed only to introduce contents into a generic framework predetermined in the authoring system. Some of educational authoring systems are discussed by Tom Murray in [2]. However, most of them were designed without taking into account accessibility features for disabled people.

The authoring tools may be accessible to authors regardless of disability and they may support and encourage the authors in creating accessible elearning content [3]. Because most of the content of the Web based ALS is created using authoring tools, they play a critical role in ensuring the accessibility of the Web. Since the Web is both a means of receiving information and communicating information, it is important that both the Web content produced and the authoring tool itself be accessible [3,4].

In Algeria, the number of disabled students entering higher education has increased over the last few years. The Algerian legislation was introduced to ensure that disabled students have the same opportunities as non-disabled students and it is expected that the educational community should do as much as possible to ensure that this happens, and so, ensuring that inclusive learning is

available to all students without discrimination. As well as fulfilling legal and ethical obligations, this will also improve the learning environment for all students, regardless of their support requirements.

However, while Algerian's legislation includes laws protecting the rights of disabled persons, and the country is a signatory to related international conventions, the daily reality is otherwise. For students and teachers with disabilities, going to the university involves a considerable effort that goes beyond keeping up with studies. Often lacking of lodging in the campuses, their problem begins with public transport vehicles where the most of them have no accommodations for wheelchairs. Given the difficulties of getting around, some of them decide to abandon their studies and return to their home villages and towns simply because they can't continue in this way, especially when the scholarship is relatively low.

In order to help disabled students at the University of Batna (Algeria) to cope with the various problems they encounter during their schooling at the university, a Research Project (NRP 12/u05/5212) is initiated in 2010 with as objective the integration of the ICT (Information and Communication Technologies) in the learning process of the disabled students. A blended learning solution that combines distance learning and face to face learning sessions has been adopted by the Research Group. The development of a web based ALS that takes into consideration accessibility features for disabled people is thus proposed.

This is the purpose of this paper. We investigate this idea through a web based ALS dedicated to Algerian Universities called EASY (Elearning Accessible System).

We organize the rest of this paper as follows. Section 2 and section 3 summarize briefly the concept of accessibility and the W3C accessibility guidelines both WCAG and ATAG. Sections 4 until section 8 present the details of the ALS design, and finally, section 9 briefly concludes this paper.

II. THE W3C ACCESSIBILITY GUIDELINES

A. What is Web Accessibility

Web accessibility means that people with disabilities (deaf, blind, dyslexic, etc.) can use the Web. More specifically, it means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits people without disabilities in certain situations, such as people using a slow Internet connection, people with "temporary disabilities" such as a broken arm, and people with changing abilities due to aging (visual, hearing, etc.) [3].

B. Why Web accessibility is important

The Web is fundamentally designed to work for all people, whatever their hardware, software, language, culture, location, or physical or mental ability. Also, the

Web becomes an increasingly important resource in many aspects of life: education, employment, government, commerce, health care, and more.

Thus, it is essential that the Web be accessible in order to provide equal access and equal opportunity to people with disabilities. An accessible Web can also help people with disabilities more actively participate in society. When websites, web technologies, or web tools are badly designed, they can create barriers that exclude some people from using the Web [3].

C. The W3C/WAI Accessibility Guidelines

The Web Content Accessibility Guidelines (WCAG) provides an international set of guidelines. They are developed by the Worldwide Web Consortium (W3C), the governing body of the web. These guidelines are the basis of most web accessibility law in the world.

One of the roles of the Web Accessibility Initiative (WAI), that is a W3C project, is to develop guidelines and techniques that describe accessibility solutions for "Web software" and Web developers. These WAI guidelines are considered the international standard for Web accessibility.

In 1999, the WAI published the first version of the Web Content Accessibility Guidelines (WCAG 1.0). On 11 December 2008, the second version (WCAG 2.0) is released. WCAG 2.0 aims to be up to date and more technology neutral. In recent years, these have been widely accepted as the definitive guidelines on how to create accessible websites.

While Version 1.0 of the Accessibility Guidelines focused heavily on the techniques for accomplishing accessibility, especially as related to HTML, Version 2.0 takes a different approach: it focuses more heavily on the principles of accessibility, and presents some techniques in separate documents. By focusing more on principles rather than techniques, version 2.0 of the guidelines is more flexible, and encourages developers to think through the process conceptually. Version 2.0 of these guidelines, are based on four principles, where the first letters of them that spell the word POUR may help us to remember them [4]:

- **Perceivable:** Available to the senses (vision and hearing primarily) either through the browser or through assistive technologies (e.g. screen readers, screen enlargers, etc.)
- **Operable:** Users can interact with all controls and interactive elements using the mouse, keyboard, or an assistive device.
- **Understandable:** Content is clear and limits confusion and ambiguity.
- **Robust:** A wide range of technologies (including old and new user agents and assistive technologies) can access the content.

On 27 July 2012, a first Draft of a document untitled “Applying WCAG 2.0 to Non-Web ICT” is published and it describes how the WCAG 2.0 and its principles, guidelines, success criteria and conformance model can be applied to non-Web Information and Communications Technologies (ICT), specifically non-web documents and software aspects of products [3].

D. The major categories of disability types

The major categories of disability types are [5]:

- *Visual*: Blindness, low vision, color-blindness.
- *Hearing*: Deafness.
- *Motor*: Inability to use a mouse, slow response time, limited fine motor control.
- *Cognitive*: Learning disabilities, distractibility, and inability to remember or focus on large amounts of information.

Each of the major categories of disabilities requires certain types of adaptations in the design of the web content. Most of the time, these adaptations benefit nearly everyone, not just people with disabilities. For example, while captions are a necessity for Deaf users, they can be helpful to others, including anyone who views a video without audio.

E. Principles of Accessible Design

Below is a list of some key principles of accessible design. This list does not present all accessibility issues, but by addressing these basic principles, we will ensure greater accessibility of web content to everyone [5].

- Provide appropriate alternative text to non-text content in web pages, in order to help people who are blind and rely on a screen reader to have the content of the website read to them.
- Provide headings for data tables to enable for screen reader users to navigate and understand the data table.
- Ensure users can complete and submit all forms, and make sure the user can submit the form and recover from any errors, such as the failure to fill in all required fields.
- Ensure links make sense out of context because screen reader users may choose to read only the links on a web page. Certain phrases like "click here" and "more" must be avoided.
- Caption and/or provide transcripts for Videos and live audio. With archived audio, a transcription may be sufficient.
- Ensure accessibility of non-HTML content, including PDF files, Microsoft Word documents, PowerPoint presentations and Adobe Flash content in a way to help people using a screen reader.

- Allow users to skip repetitive elements on the Web page by providing a method that allows users to skip navigation or other elements that repeat on every page.
- Do not rely on color alone to convey meaning because that information may not be available to a person who is colorblind and will be unavailable to screen reader users.
- Make sure content is clearly written and easy to read by using clear fonts, headings and lists appropriately.
- Make JavaScript accessible by ensuring that its event handlers are device independent (e.g., they do not require the use of a mouse) and make sure that Web pages does not rely on JavaScript to function.
- Design to standards by using valid HTML and CSS in order to separate content from presentation. These provide better search engine optimization.

F. Essential Components of Web Accessibility

It is clear that Web accessibility depends on several components working together, in order for the Web to be accessible to disable people. These components are [6]:

- *Content*: the information in a Web page or Web application, including (natural information such as text, images, and sounds; code or markup that defines structure, presentation, etc.
- *Web browsers, media players, and other User Agents*.
- *Authoring tools*: software used to create Web sites.
- *Evaluation tools*: Web accessibility evaluation tools, HTML validators, CSS validators, etc.
- *Assistive technologies*: in some cases (screen readers, alternative keyboards, switches, etc.).
- *Users' knowledge and experiences*: and in some cases, adaptive strategies using the Web.
- *Developers*: designers, coders, authors, etc., including developers with disabilities.

The interaction between these components is as follows (Fig. 1) [3]:

- Web developers usually use authoring tools and evaluation tools to create and validate Web content.
- Users use Web browsers, media players, assistive technologies, or other User Agents to get and interact with the content.

Web accessibility guidelines for all these components are developed by W3C (WAI):

- Authoring Tool Accessibility Guidelines (ATAG) addresses authoring tools.

- Web Content Accessibility Guidelines (WCAG) addresses Web content, and is used by developers, authoring tools, and accessibility evaluation tools.
- User Agent Accessibility Guidelines (UAAG) addresses Web browsers and media players, including some aspects of assistive technologies.

W3C/WAI guidelines are based on the fundamental technical specifications of the Web, and are developed in coordination with W3C technical specifications (HTML, XML, CSS, SVG, SMIL, etc.).

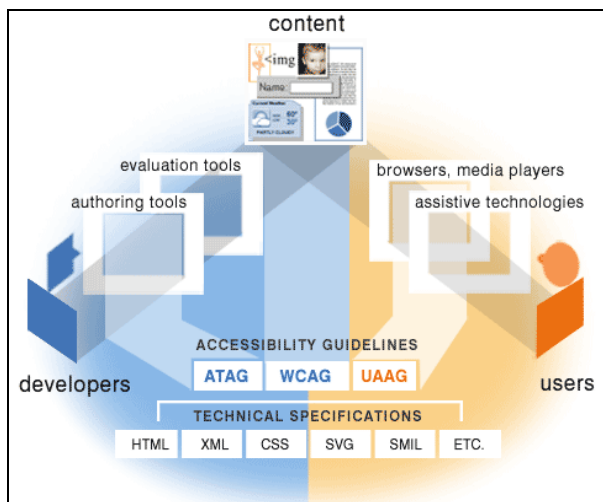


Figure 1. Interaction between the different components

III. THE ACCESSIBILITY AND AUTHORING TOOLS

As we have seen in the previous section, the authoring tools constitute the components of the Web that are addressed by the W3C accessibility guidelines (ATAG). Authoring tools are software and services that people use to produce Web pages and Web content. Types of authoring tools are listed below [7]:

- Editing tools specifically designed to produce Web content, for example, WYSIWYG HTML and XML editors.
- Tools that offer the option of saving content in a Web format, for example, word processors or desktop publishing packages.
- Tools that transform documents into Web formats, for example filters to transform desktop publishing formats to HTML.
- Tools that produce multimedia, especially where it is intended for use on the Web, for example, video production and editing suites, SMIL authoring packages.
- Tools for site management or site publication, including content managements systems (CMS), tools that automatically generate Web sites dynamically from a database, on-the-fly conversion tools, and Web site publishing tools.

- Tools for management of layout, for example, CSS formatting tools
- Web sites that let users add content, such as blogs, wikis, photo sharing sites, and social networking sites

The goal of the Authoring Tool Accessibility Guidelines (ATAG) is to:

1. Define how authoring tools should help Web developers produce Web content that is accessible and conforms to the Web Content Accessibility Guidelines (WCAG).
2. Explain how to make authoring tools themselves accessible so that authors with disabilities can use the tools [8].

ATAG is primarily addressed for developers of authoring tools but also intended to meet the needs of many different audiences, including policy makers, managers, and others. For example:

- People who want to choose authoring tools that are accessible and that produce accessible content can use ATAG to evaluate authoring tools.
- People who want to encourage their existing authoring tool developer to improve accessibility in future versions can refer the authoring tool vendor to ATAG.

There are two versions of ATAG: ATAG 1.0 which was made a recommendation on 3 February 2000, and ATAG 2.0 last released on 10 April 2012, which is currently in the last Working Draft status. The Guidelines are accompanied by a “techniques document” that suggests how the guidelines can be implemented using various tools. The ATAG 2.0 guidelines are broken down into two parts (Part A and Part B), each consisting of several principles.

Part A focuses on ensuring that people (authors) with disabilities are able to effectively use an authoring tool, while Part B focuses on ensuring that content or applications created by an authoring tool meet accessibility standards and incorporate accessibility features. Here below are the principles of the ATAG 2.0 Guidelines [8]:

Part A. Make the authoring tool user interface accessible

A.1. Authoring tool user interfaces must follow applicable accessibility guidelines

A.1.1. (For the authoring tool user interface) Ensure that web-based functionality is accessible

A.1.2. (For the authoring tool user interface) Ensure that non-web-based functionality is accessible

A.2. Editing-views must be perceivable

A.2.1. (For the authoring tool user interface) Make alternative content available to authors

A.2.2. (For the authoring tool user interface) Editing-view presentation can be programmatically determined

A.3. Editing-views must be operable

A.3.1. (For the authoring tool user interface) Provide keyboard access to authoring features

A.3.2. (For the authoring tool user interface) Provide authors with enough time

A.3.3. (For the authoring tool user interface) Help authors avoid flashing that could cause seizures

A.3.4. (For the authoring tool user interface) Enhance navigation and editing via content structure

A.3.5. (For the authoring tool user interface) Provide text search of the content

A.3.6. (For the authoring tool user interface) Manage preference settings

A.3.7. (For the authoring tool user interface) Ensure that previews are at least as accessible as in-market user agents

A.4. Editing-views must be understandable

A.4.1. (For the authoring tool user interface) Help authors avoid and correct mistakes

A.4.2. (For the authoring tool user interface) Document the user interface including all accessibility features.

Part B. Support the production of accessible content

B.1. Fully automatic processes must produce accessible content

B.1.1. Ensure automatically specified content is accessible

B.1.2. Ensure accessibility information is preserved

B.2. Authors must be supported in producing accessible content

B.2.1. Ensure accessible content production is possible

B.2.2. Guide authors to produce accessible content

B.2.3. Assist authors with managing alternative content for non-text content

B.2.4. Assist authors with accessible templates

B.2.5. Assist authors with accessible pre-authored content

B.3. Authors must be supported in improving the accessibility of existing content

B.3.1. Assist authors in checking for accessibility problems

B.3.2. Assist authors in repairing accessibility problems

B.4. Authoring tools must promote and integrate their accessibility features

B.4.1. Ensure the availability of features that support the production of accessible content

B.4.2. Ensure that documentation promotes the production of accessible content

These principles describe features with which an authoring tool should comply in order to ensure that it is itself accessible and the Web content it produces is also as accessible as possible. However, it is clear that authoring tools are not all compliant to these principles; they do not promote insertion of accessibility features such as alternative text for images for example [9].

Fortunately, since the recommendation of the ATAG 1.0, there has been a flurry of development of accessibility checking and retrofitting tools such as (Bobby, A-Prompt, The Wave, etc.), and some of these are now integrated into popular web content authoring tools such as (Dreamweaver, Flash, Director, etc.). A similar effect is noticeable in authoring tools aimed specifically at the learning technology field, and accessibility of courseware authoring tools such as (Blackboard, WebCT, Captivate, Lectora, Storyline, etc.) is now being addressed.

However, even with an authoring tool specifically designed to create fully accessible content, it is vital for content authors to be aware of accessible design techniques, particularly in light of the current constraints affecting Web development environments. Content developers should be aware of the limitations of authoring tools in creating accessible content and should ensure that all resources created are not only designed with accessibility in mind but are checked for accessibility throughout the design lifecycle of the resource [9].

IV. GENERAL ARCHITECTURE OF THE EASY SYSTEM

As presented in (Fig. 2) there are three main environments in development during the first stage of the project. Each Environment is dedicated to a category of users which are Administrator, Students and Teachers.

A. The Administration Environment

Via this environment, the System Administrator takes full control over the system, manages system resources and settings, creates user accounts and user groups, and assigns user permissions. The system administrator also monitors event logs and maintain the system in secure and stable state.

B. The Adaptive Learning Environment.

Two course delivering modes are accessible to the student in the Adaptive Learning Environment (ALE):

“discovering mode” (free exploration of the course) and “learning mode” (planned learning with self-assessment process).

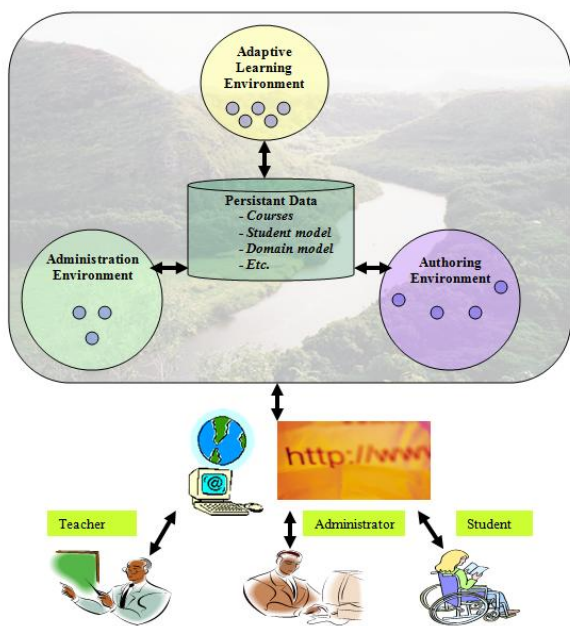


Figure 2. General architecture of the EASY system

The ALE architecture is composed of five (5) modules:

1. *Module 1.* An “Exploration Module” that represents the “discovering mode” of the student.
2. *Module 2.* A “Domain-Expert Module” using a domain model and some rules to search and filter concept-indexed HLU asked by a pedagogical module at a given moment. The domain model contains knowledge about the subject to be taught. Both domain concepts and relationships among them are represented.
3. *Module 3.* A “Pedagogical Module” that allows the negotiation of learning goals with the student and generates in turn, sequencing plan for the adaptive presentation of the lesson. Two sub-modules realize these two tasks: the “negotiator” using some “negotiation rules” and the “planer” using some “planning rules”. This module will provide students with the necessary guidance during the learning process. It will design and compose the tutorial sessions, that is, it will decide in each moment the adequate task to be performed by the student.
4. *Module 4.* A “Diagnosis Module” that allows the student evaluation and the maintenance of an overlay type student model. This module is made up of three sub-modules: an “evaluator” using some “evaluation rules”; a “deduction agent” using some “deduction rules; and a “student model manager” managing its persistent content. The function of this module is to create and update the student model. Once a tutorial component is executed, the student is evaluated and its model is updated.

5. *Module 5.* A “Supervisor Module” that allows on one side, the communication of the system with the student, and on another side, the coordination between the other modules.

C. The Authoring Environment

The authoring environment will be used by teachers to:

- Create and modify accessible courses according to pre-defined models. The teachers supplies and manages learning object meta-data records according to LOM standard, describes the course structure and provides course materials by uploading resources.
- Monitor the evolution of their students. It will show the progress of each student, the statistics about the course use, and average student performances.

The most important shortcoming of a Hypermedia Based ALS is the authoring part. Developing knowledge space in ALS is not simple and it is very time consuming. In addition, courseware is usually non transferable and non reusable [10]. Therefore, some research has been done to address the problems by developing generic authoring models and systems. For example, we find in the literature, My Online Tutor [10,11], LAOS authoring model [12, 13] AHA [14] and WHURLE [10, 15].

Murray [2,16] listed more than twenty references in his state of the art review of the authoring systems dedicated to intelligent tutoring systems and adaptive hypermedia systems. He has classified them in seven categories according to the type of adaptive learning system they produce. These categories are as follows: (1) curriculum sequencing and planning, (2) tutoring strategies, (3) device simulation and equipment training, (4) domain expert system, (5) multiple knowledge types, (6) special purpose and (7) intelligent/adaptive hypermedia.

The EASY system, presented in this paper, generates adaptive courseware that can be typed as first and seventh category of the Murray classification mentioned above. In this category of authoring systems the learning material is generally structured as a network of Learning Units (LUs) where every LU is build to satisfy some learning goals. The LUs are linked together to show some relationships between them.

Although these authoring systems do not use any explicit representation of domain knowledge but hypertext representation, they investigated nevertheless the intelligence at the sequencing process of the LUs, the manipulation of the hypertext links and the adaptation of the course according to a student level of knowledge. The LUs to be presented to the learner are then adapted dynamically based on the learner model, the lesson learning objectives and the relations that exist between the different LUs.

On the other hand, adaptive educational hypermedia systems require massive and complex knowledge spaces in order to provide adaptation and personalization in

learning. Hence, it is difficult for a single person or a small group of people to develop them effectively. Former studies addressed the problem by developing collaborative authoring systems.

The EASY system is web based, though; it allows teachers to collaborate in courseware production by sharing different tasks. However, we should notice that the software does not constitute the only aspect in the success of such cooperative system. Also, we have to take into account the human factors involved due to the group activities because of their importance. Thus, to avoid the inherent conflicts due to the human nature, we propose a group organization that allows an optimal way the construction of the courseware.

This organization facilitates also the manipulation of different components of the courseware during all steps of the project advancement. So, we define three roles through which the teachers can participate during the building process: “main teacher”, “helping teacher” and “reviewing teacher”.

- The role of the “main teacher” is to coordinate the whole work and to verify that the calendar is well respected. He defines the courseware logical structure to be produced by decomposing it in several components (parts, chapters, LU, figures, images, etc.), then he affects the roles to different co-teachers. He has free access to all courseware components.
- A “helping teacher” is authorized to create, modify or delete only the components assigned to him. On the remaining courseware components he will have only the role of reviewing teacher”.
- A “reviewing teacher” is authorized only to read and/or comment the components assigned to him.

More details about architecture of the EASY authoring tool will be presented below in section VI.

V. THE COURSEWARE STRUCTURE AND SEQUENCING

Due to the differences in background knowledge, learning styles and preferences, individual students may take very different approaches towards learning. Therefore, Adaptive Educational Hypermedia (AEH) has been developed to offer students personalized learning content to improve their learning outcome.

The domain of AEH [17,18] is a relatively new direction of research on the crossroads of hypermedia and learner modeling. This domain is an alternative to the traditional "one-size-fits-all" approach in the development of hypermedia systems. Adaptive educational hypermedia systems build a model of the goals, preferences and knowledge of each individual learner, and use this model throughout the interaction with the learner, in order to adapt the hypertext to the needs of that learner.

The year of 1996, the start of the rapid increase in the use of the Web, could be considered a turning point in adaptive hypermedia research. The Web, with its clear demand for adaptivity, served to boost adaptive hypermedia research, providing both a challenge and an attractive platform. All the early systems were essentially lab systems, built to explore some new methods, which used adaptivity in an educational context [17,18].

The learning process in EASY system is organized around adaptive hypermedia components. The learning material is structured in three abstraction level hierarchy according to three level hierarchy of learning goals defined in [19]: Parts (satisfying the general goals), Chapters (satisfying the specific goals) and the Hypermedia Learning Units (HLU) (satisfying the operational goals).

To intelligently sequencing the curriculum and adapt it to each learner capacities, the management of these components, is ensured by rule based system that use five sets of production rules. Every set of rules has a specific function. These functions are the following: “Negotiation” of the start entry point in the course and/or the objectives to reach; “Deduction” of HLU units assumed to be understood after a negotiation phase; “Planning” the learning session; “Searching and filtering” the content of HLU; and finally “Assessment ” of the learner knowledge.

These rules (for which parameters can be set), called “Main Rules” (MRules), describe the different tutoring plans depending on the different learning situations. They constitute therefore a generic knowledge base that is instantiated in a suitable way for each Courseware created by EASY system [20].

The instantiation process, producing “Generated Rules” (GRules), is carrying out automatically by the system on the base of parameters delivered by authors. These Courseware parameters which are structured in predicates form, describe the quantitative aspect of the teaching material (number of parts, number of chapters, number of hypermedia learning units, number of questions, number of exercises, etc.).

For reusability and independence from the domains criteria, the Main Rules invoke abstracted structures called Hypermedia Learning Units (HLU). These HLU are supposed to receive all kinds of knowledge about the domain via instantiation, under all media types that are allowed by the XML/HTML languages (text, image, sound, animation, video, and applet). We consider five types of HLU: (courseware abstract, part abstract, chapter abstract, paragraph, and evaluation).

To summarize, we can consider, three levels of knowledge in the curriculum definition of the courseware as shown in (Fig. 3) [21]:

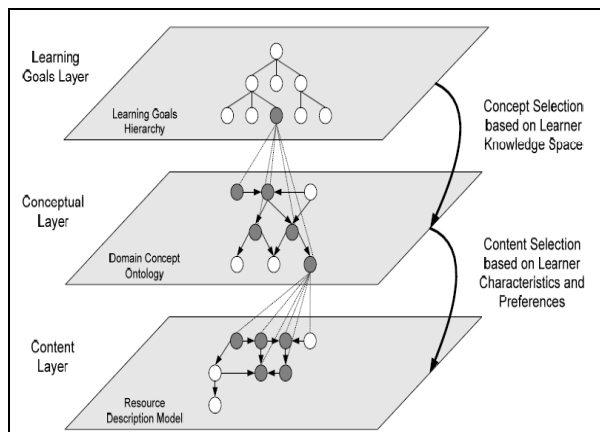


Figure 3. The three abstraction layers of adaptive courseware

As in Aroyo & Mizoguchi [22], adaptive HLU sequencing is based on two main processes, namely, the concept selection process and the content selection process.

- In the concept selection process, a set of learning goals from the Learning Goals hierarchy is selected by the learner.
- For each learning goal, related concepts from the Domain Concept Ontology are selected. These concepts are filtered by the pre-existing knowledge of the learner (Learner Knowledge model) creating a sequence of missing concepts that need to be covered in order to reach the selected learning goals.
- In the content selection process, learning resources for each concept of the concept sequence are selected from the Resource Space based on the content selection rules that relate the educational characteristics of learning resources with the cognitive characteristics and learning preferences of learners.

The result of this process is a personalized learning path that matches the selected learning goals. The Fig. 3 presents the abstract layers of the adaptive courseware sequencing, demonstrating the connection of the above mentioned processes.

VI. THE EASY AUTHORIZING TOOL

A. Design Principles

We design the EASY accessible authoring tool as an accessible web application for ease of use. A teacher or content creator can create, test, and deploy courses without installing any additional software. The tool is web-based in terms of the authoring tool itself, not just the output files (i.e., the tool uses the web browser as the application interface). The server-based applications have the advantage of enabling collaborative authoring and permission/role-based production workflows.

The authoring process should be made much simpler than in some existing GUI-based authoring tools. The authoring component should enable the straightforward

creation of concepts, the linkage of concepts by prerequisite relationships, and easy generation of the assessment exercises. It should be user-friendly enough to enable a person who is not a computer expert to design the courseware. That includes the development of a graphic editor necessary to build different network types as concept-network for example, which will enable the authors to define the relationships between elements with a drag-and-drop interface.

From an author point of view, building a courseware in EASY system consists in introducing a set of objects that will be manipulated in the Adaptive Learner Environment (ALE). These objects (see Fig. 3) are made up with learning material in the form of hypermedia learning units (HLU); hlu-prerequisite-networks, learning-goal-networks and concept-network in the form of an oriented graph; courseware parameters in the form of predicates; and finally pedagogical knowledge in the form of production rules.

The authoring tool enables also collaboration between teachers when editing courseware. The collaboration task in EASY is introduced at the editing level of the teaching material and at the editing level of the prerequisite-networks. These two components are well structured: the teaching material is organized as (Parts, Chapters and HLU), and the prerequisite-networks are organized as sub-networks form (part-prerequisite-network, chapter-prerequisite-network, HLU-prerequisite-network, learning-goal-networks and concept-network).

These structures are well convenient for the fragmentation and then constitute the basis of our collaborative editing approach as in Alliance [23]. The two concept-keys on which is based the design of EASY are the “fragmentation” and “edition roles” [23]. We defined three edition roles for the teachers: “main teacher”, “helping teacher” and “reviewing teacher”.

At the beginning of the courseware construction task, a negotiation step is necessary. The main teacher assigned the edition roles to different co-teachers around different fragments of course structure in accordance with their competences and availability. For example one teacher realizes chapter 1 and the second realizes chapter 2, etc.

Five learning principles have been incorporated into the authoring process of the courseware [24]: clear definitions of learning goals, definition of pre-requisite knowledge, providing a variety of presentation styles (tell, show and do), enhanced feedback and testing, and permitting the learner to control the direction of the learning session by choosing himself the learning goals. The accessibility features relatively to W3C guidelines must be respected by the teachers in order to produce accessible courseware.

B. The EASY authoring tool architecture

The authoring tool software is divided in two layers [25]: Editing layer and Presentation layer. Every layer is structured as a collection of modules where each module consists of several objects implementing some

functionality (Fig. 4). The need for information exchange between the two layers implies the presence for a Dialog Controller. Messages exchanged between the two layers transits automatically by the Dialog Controller. According to message-type, the convenient objects are then executed among those that are defined in a layer. The two layers are presented below:

B.1. Presentation Layer: This layer gathers an organized set of interactive objects defining the graphical user interface (buttons, icons, scrolling bar, pull-down menus, etc.). Thus, for every object, modeling a part of our application domain, we associate a presentation technique accomplished by a reactive object that reacts to the different authoring actions. Besides the pull-down menus achieving the different functions, we especially find a toolbox containing graphical icons that refer to the frequently used functions and specialized widget-based palette allowing the graphical construction of the prerequisite-networks.

B.2. Editing Layer: This layer gathers many types of functionalities allowing every author to manipulate the objects that constitutes the courseware among which those that concern the course logical structure management, as well as the content of the course components. These functionalities include not only the support of individual actions, but also the sharing aspect and communication between teachers when they need collaboration.

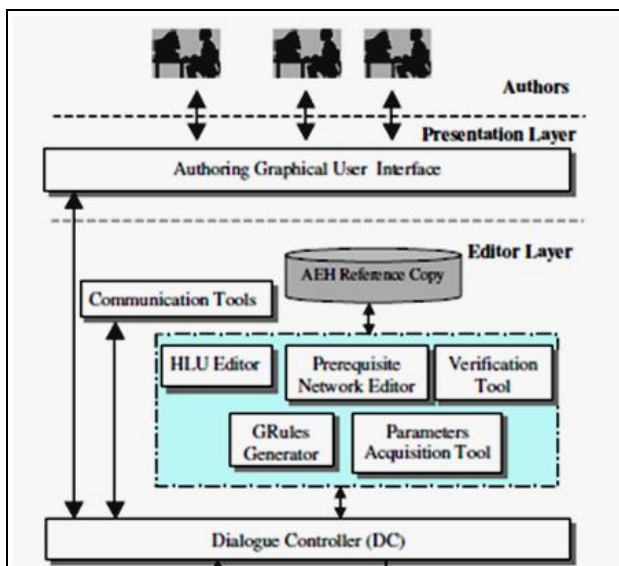


Figure 4. Easy authoring tool architecture

The components of the editing layer are:

B.2.1. HLU Editor and networks graphical Editor: Two modules are designed to implement this component software. They allow the creation task and the maintenance of different course objects. The first module allows the WISIWIG HLU edition using X/HTML language. The second module allows the author to edit the prerequisite-

networks in a graphical form. This oriented network is made up of linked nodes where the links indicates the different possible progressions between the teaching material components.

B.2.2. Parameters acquisition module: This module allows the main author to specify the course parameters that indicate the manner in which the teaching material is decomposed (number of parts, number of chapters in every part, number of HLU in every chapter, etc.). These parameters are saved in the predicates form and then used to instantiate the Main Rules. For example the predicate nbhlu (1,2,4) indicates that chapter 2 in part 1 contains 4 HLU.

B.2.3. GRules generator module: This module allows the author to generate the five packages of generated rules that represent different tutoring plans. Based on the course parameters introduced via the previous module, this generation consists of an instantiation of the five packages of the MRules.

B.2.4. Verification module: As most authoring systems, EASY offers a tool to help the author in the diagnosis of errors and bugs. It facilitates detection of incoherencies that can be occurred during the course construction. For example, at the end of the construction process of the course, it is necessary to check the compatibility of course parameters with the effective structure of the teaching material.

B.2.5. Communication Tools: Actually, three different communication applications are proposed for the first prototype: internal e-mail, virtual chat rooms, and shared/personal space (web-based virtual disks). All teachers can use them in order to collaborate while working on a group courseware projects.

B.3. Dialogue Controller: The dialog controller is composed of three independent modules performing respectively, the "message reception", the "message control" and the "message sending". The "Control" module allows the coordination and synchronization of the running of the different modules within the two layers, in accordance with the actions of the different authors. At any time, it uses all necessary information to determine exactly what the functionalities to invoke within the concerned layer are. Every time that an event occurs, the associated "Receiver" module delivers the message materializing this event to the "Control module". The "Control" module reacts then following three steps: analyze the event, draw up an action plan and then carry out the established plan.

VII. ACCESSIBILITY CONSIDÉRATIONS IN EASY SYSTEM

We have obviously taking into consideration the W3C ATAG guidelines in the design of the ALS. We have

considered some guidelines to make the authoring tool itself accessible and some guidelines to make content produced by the teachers accessible to the students. We have started by analyzing what are the needs of the different categories of the disabled students when using the internet based courses. Until now, we have considered three disability types (as shown in TABLE I) and we have fixed what accessibility features are needed in each situation [26].

We have also study the different accessibility solutions provided by the different operating systems (windows, Linux, etc.), solutions provided by the different web browsers used by the majority of people (Internet Explorer, Firefox, Chrome, Opera, etc.), solutions provided by a number of assistive technology software that are free and/or open source, and accessibility validation software solutions, especially those which are free.

We share the idea with Course Avenue developers [27] that “*making the interactions between web browsers, assistive technology (e.g., screen readers), learning content, and learning management systems work together can complicate development of accessible e-Learning courses. The likelihood and cost of producing non-compliant e-Learning is very high when inexperienced developers make their first attempts at accessibility*”.

As opposed to relying on each and every teacher to interpret and apply the accessibility standards, important standards are built directly into our ALS. The ALS with built-in accessibility provides a shorter and easier path for teachers to provide a single course that is accessibility compliant and usable by all learners, including those with disabilities.

TABLE I. DISABILITY TYPES AND ACCESSIBILITY FEATURES NEEDED

Disability Types	Categories	Accessibility features needed
<i>Visual</i>	<ul style="list-style-type: none"> - Blindness - Low vision - Color blindness 	<ul style="list-style-type: none"> - Don't write scripts that require mouse usage. Supply keyboard alternatives. - For images, provide text descriptions, in alt text and, if necessary, longer explanations - Allow for users to skip over navigational menus, long lists of items, ASCII art, and other things that might be difficult or tedious to listen to using screen readers. - Do not rely on color alone to convey meaning - Limit or eliminate text within graphics
<i>Auditory</i>	<ul style="list-style-type: none"> - Deaf - Hard of hearing 	<ul style="list-style-type: none"> - Provide transcripts for audio clips. - Provide synchronous captioning and transcripts for video clips
<i>Motor</i>	<ul style="list-style-type: none"> - Paraplegia, quadriplegia - Parkinson's disease - Arthritis - Multiple sclerosis - Muscular dystrophy - Cerebral palsy 	<ul style="list-style-type: none"> - Make sure that all functions are available from the keyboard (try tabbing from link to link). - Make sure that the web pages are error-tolerant (e.g. ask "are you sure you want to delete this file?"), do not create small links or moving links.

So, our approach to accessible e-Learning development is to embed compliance and usability into the software itself. With accessibility standards built-in, courses comply without additional work. The (Fig. 5) below for example enthused from the Jisc web site [28] shows the accessibility tools that appears on the interfaces of the Authoring Tool and the Learning Environment together.

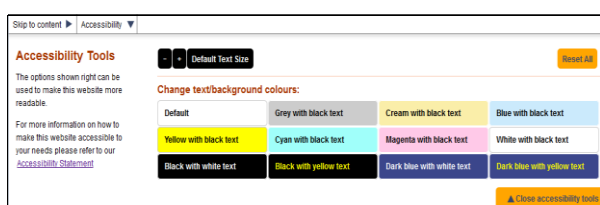


Figure 5. Accessibility Tools provided in the ALS

The Link “Skip to Content” that is spoken by screen readers allows users to skip over navigational menus and other things that might be difficult or tedious to listen to, using screen readers. This link will appear when “tabbed to” using a keyboard.

If the tools provided are not sufficient and the user (teacher or student) has any problems accessing the page, he is oriented through the Link “Accessibility Statement” to the BBC’s My Web My Way site [29] which provides free “how to guides” to all users. The different guides classified by disability types (Fig. 6) provide accessibility help, enabling computer users to make the most of the internet whatever their ability or disability. Especially, it provides advice on how to access content for various browsers and operating systems. It also includes guides and factsheets that introduce the range of assistive

technologies that are available. The site “My web my way” has been developed and maintained through a partnership between “BBC Online” and “Ability Net” [30], a leading UK computing and disability charity.



Figure 6. How to Guides of the BBC My Web My Way

VIII. CURRENT STATE OF THE WORK

The ALS is being developed with PHP/MySQL technologies. The authoring tool allows teachers to collaborate in the construction of the courseware. The shared space is represented by a set of structured components (HLU, prerequisite-networks, course parameters and the five packed rules) and tools, which make it possible the edition and communication tasks.

Five learning principles have been incorporated into the authoring process [24]: clear definition of learning goals, definition of pre-requisite knowledge, providing a variety of presentation styles (to tell, to show and to do), enhanced feedback and testing, and permitting the learner to control the direction of the learning session by choosing himself the learning goals.

Two different approaches were used to initially test the validity that the system actually incorporated pedagogy and effective collaboration and accessibility design concepts as part of the development process. To initially evaluate the system, a group of five teachers were surveyed to seek their opinion if the authoring system did incorporate the five learning principles into its design. Their survey results validated that the system would prompt developers to build a course based on pedagogy. In addition a high agreement was noted in the self-direction of the lesson offered to the learner.

In a second means to validate the system, five teachers geographically dispersed were invited to develop a course on the “Relational Data Bases” (Fig. 7), via local network, and were surveyed to seek their opinion if the authoring system offers all collaborative tools necessary to build the courseware.

Although the system does exhibit positive results after a pilot test in the local network context, a question for

future research is the experimentation of the system in the internet/web context. This research would provide evidence that the concepts incorporated into the system do impact learning in a positive manner. On the positive side the survey results from the two different experimentations provides indication that the system is a positive benefit to teachers and developers of adaptive educational hypermedia.

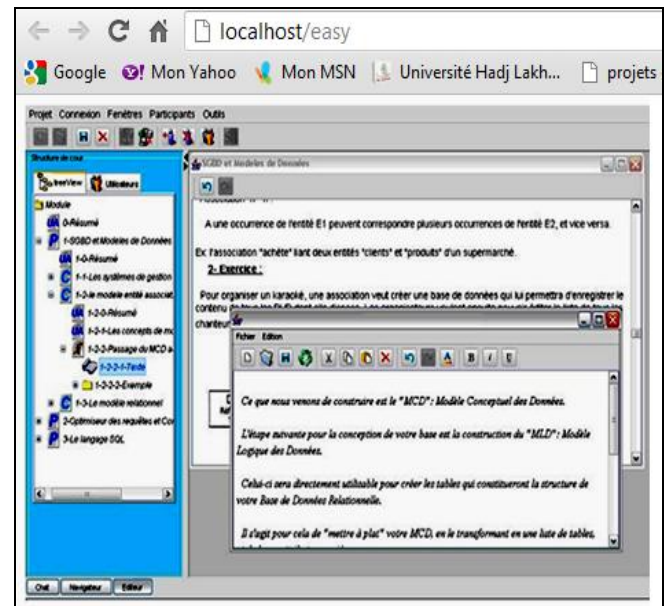


Figure 7. Course Developing on “Relational Data Bases”

IX. CONCLUSION

Adaptive learning technologies have gaining ground in online learning environments. These technologies allow for the development of more personalized online learning experiences with materials that adapt to student performance and skill level. We also find the term adaptive used to describe Assistive Technologies that increase the usability of computer systems for those with disabilities and special needs.

The legislation in Algerian universities was introduced to ensure that disabled people have the same opportunities as non-disabled people and it is expected that the educational community should do as much as possible to ensure that this happens. Assistive Technologies have an important role to play in ensuring that inclusive learning is available to all students without discrimination.

In this way, we have designed an adaptive learning system that assists disabled students to access learning activities over the web. The collaborative authoring tool is designed to satisfy guidelines of accessibility of the W3C recommendation. Three concept keys have guided the design of the system: adaptivity, accessibility and collaboration.

Adaptive learning allows a course to automatically adapt to learners' needs. Based on the outcome of pre-

assessment or individual questions, the learner is given individual study advice to either improve knowledge on a weak point or to skip areas that already assess at the desired level.

Accessibility generally describes the degree to which a product, device, service, or environment is available to as many people as possible. When referring to accessibility of online content, such as e-Learning, accessibility refers to the degree to which all people, especially those with disabilities, can access the content.

Through collaboration, authoring is a team process between teachers. The EASY authoring tool supports this multi-user process in its functionality and checks and balances of version control and review cycles.

Students can be involved in their learning through adaptive learning. They create their own learning path as they complete pre-assessments so they can be directed to specific course areas to learn what they don't know or where they may need improvement or a better understanding.

ACKNOWLEDGMENTS

The work reported in this paper is supported by the PNR project funded by the Algerian Ministry of Higher Education and Scientific Research.

REFERENCES

- [1] Dagger D., Wade V., Conlan O. "Personalisation for All: Making Adaptive Course Composition Easy". *Educational Technology & Society*, 8 (3), 9-25, 2005.
- [2] Murray, T. "Authoring Intelligent Tutoring Systems: an analysis of the state of the art", *International Journal of AI in Education*, vol. 10, 98-129, 1999.
- [3] W3C WAI Website. "Introduction to web accessibility". <http://www.w3.org/WAI/intro/accessibility.php>, accessed July 2012.
- [4] WebAIM Website. "Constructing a POUR Website, putting people at the center of the process". <http://webaim.org/articles/pour/>, accessed July 2012.
- [5] WebAIM Website. "Introduction to Web Accessibility". <http://webaim.org/intro/>, accessed July 2012.
- [6] WebAIM Website. "Essential Components of Web Accessibility". <http://www.w3.org/WAI/intro/components.php>, accessed July 2012.
- [7] W3C WAI Website. "Authoring Tool Accessibility Guidelines (ATAG) Overview". <http://www.w3.org/WAI/intro/atag.php>, accessed July 2012.
- [8] W3C WAI Website. "Authoring Tool Accessibility Guidelines (ATAG) 2.0". <http://www.w3.org/TR/ATAG20/>, accessed July 2012.
- [9] Slatin J. M. and Rush S., "Creating accessible e-learning content, Maximum Accessibility: Making Your Web Site More Usable for Everyone", Addison Wesley Editor, 2002.
- [10] Stewart C., Cristea A. and Brailsford T.J. "Authoring Once, Delivering Many: Creating Reusable Adaptive Courseware". In 4th IAESTED International Conference on Web Based Education (WBE'05), Grindewald, Switzerland, 2005.
- [11] Ghali F. "Collaborative Adaptation Authoring and Social Annotation in MOT". Warwick Postgraduate Colloquium in Computer Science, 2008.
- [12] Cristea A., and Mooij A.D. "LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators". in WWW 2003. ACM Budapest, Hungary, 2003.
- [13] Weibel N. "Towards Adaptive Hypermedia Authoring from the Dexter Model to Laos". Institute for Information Systems, ETH Zurich, 2006.
- [14] Cristea A., Smits D., and de Bra P. "Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia". Faculty of Mathematics and Computer Science, Eindhoven University of Technology, 2005.
- [15] Meccawy M. "WHURLE 2.0: Adaptive Learning Meets Web 2.0". Third European Conference on Technology Enhanced Learning, Springer: Maastricht, the Netherlands, 2008.
- [16] Murray T. "Authoring knowledge-based Tutors: Tools for content, instructional strategy, student model and interface design". *Journal of the Learning Sciences*, vol. 7, n°1, 1998.
- [17] Brusilovsky P. "Adaptive hypermedia, methods and techniques of adaptive hypermedia". *International Journal of User Modeling and User-Adapted Interaction*, 11 (1/2), 87-110, 2001.
- [18] Brusilovsky P. "Developing adaptive educational hypermedia systems: From design models to authoring tools". In Murray, T., Blessing, S., Ainsworth, S. (Eds.), *Authoring Tools for Advanced Technology Learning Environment*, Dordrecht: Kluwer Academic Publishers, pp. 377-409, 2003.
- [19] Hameline D. «Les objectifs pédagogiques en formation initiale et en formation continue » Edition ESF, 8i ème édition, Paris, 1990.
- [20] Talhi S., Djoudi M., et Batouche M.C. «Intégration des technologies d'intelligence dans une plate-forme d'EAD ». Troisième colloque euro méditerranéen pour l'approfondissement de la formation à distance, Cemaforad'3, ISET de Sousse, Tunisie, 13-15 novembre, 2006.

- [21] Karampiperis P., and Sampson D. "Adaptive Learning Resources Sequencing in Educational Hypermedia Systems". *Educational Technology & Society*, 8 (4), 128-147, 2005.
- [22] Aroyo L., and Mizoguchi R. "Towards Evolutional Authoring Support Systems". *Journal of Interactive Learning Research*, 15 (4), 365-387, 2004.
- [23] Decouchant D., and Martínez A.M. "A Cooperative, Deductive and Self-Adaptive Web Authoring Environment". In *Proceedings of Mexican International Conference on Artificial Intelligence (MICAI-2000)*, Springer Verlag, Acapulco (Mexico), pp. 443-457, 2000.
- [24] Janicki T., and Liegle J.O. "Development and evaluation of a framework for creating web-based learning modules: a pedagogical and systems perspective". *JALN Journal*, Volume 5, Issue 1. 2001.
- [25] Talhi S., Djoudi M., and Batouche M.C. "Authoring Groupware For Intelligent Tutoring Systems". *Information Technology Journal (ITJ)*, 5(5), 860-867, 2006.
- [26] WebAIM Website. "Disability Types". <http://webaim.org/articles/>, accessed July 2012.
- [27] Course Avenue Website. "Course Avenue's Accessibility Player". <http://courseavenue.com/accessibility-player/>, accessed July 2012.
- [28] JiscTechDis Website. <http://www.jisctechdis.ac.uk/techdis/aboutus/ourwebsite/accessibility>, accessed July 2012.
- [29] BBC Website. "My Web My Way". <http://www.bbc.co.uk/accessibility>, accessed July 2012.
- [30] Ability Net Website. "Accessibility Statement". <http://www.abilitynet.org.uk/accessibility.php>, accessed July 2012.

Said Talhi received a PhD in Computer Science from the University of Batna, Algeria, in 2007 and he is currently an Associate Professor at this University. His research interests began with intelligent tutoring systems and knowledge based systems in 1992. As Information and Communication Technology became an integral component of any successful education process, his research interests focused on elearning/distance education, authoring systems, collaborative learning and adaptive hypermedia systems where some papers are published. He also published a book in these research fields at EUE (Editions Universitaires Européennes) publisher in 2011.

Fairouz Khadraoui received a Master in Computer Science from the University of Batna, Algeria, in 2005 and she is currently an Associate Professor at this University. Her research interests focused on elearning/distance education, virtual laboratories, collaborative learning and adaptive hypermedia systems.

Mahieddine Djoudi received a PhD in Computer Science from the University of Nancy, France, in 1991. He is currently an Associate Professor at the University of Poitiers, France. He is a member of SIC (Signal, Images and Communications) Research laboratory. He is also a member of IRMA E-learning research group. His PhD thesis research was in Arabic Continuous Speech Recognition. His current research interests is in E-Learning, Web mining and Information Literacy. His teaching interests include Laboratory Information Management System, Data Bases, Quality Management, Web Technology and Computerized System Validation. He started and is involved in many research projects which include many researchers from different Algerian Universities.