Developing Adaptive Elearning : An Authoring Tool Design

Said Talhi[1] and Mahieddine Djoudi[2]

[1] Department of Computer Science, University of Batna, Algeria

[2] XLIM-SIC Lab. & IRMA Research Group, University of Poitiers, France

Abstract

Adaptive hypermedia is the answer to the "lost in hyperspace" syndrome, where the user has normally too many links to choose from, and little knowledge about how to proceed and select the most appropriate ones to him/her. Adaptive hypermedia thus offers a selection of links or content most appropriate to the user. Until very recently, little attention has been given to the complex task of authoring materials for Adaptive Educational Hypermedia. An author faces a multitude of problems when creating a personalized, rich learning experience for each user.

The purpose of this paper is to present an authoring tool for adaptive hypermedia based courses. Designed to satisfy guidelines of accessibility of the W3C recommendation for authors and learners that present disabilities, the authoring tool allows several authors geographically dispersed to produce such courses together. It consists of a shared workspace gathering all tools necessary to the cooperative development task.

Keywords: *Elearning, Adaptive Hypermedia, Accessibility, Cooperative Authoring Systems.*

1. Introduction

One limitation of traditional "static" hypermedia educational applications is that they provide the same page content and the same set of links to all learners.

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) have been developed to offer students personalized learning content to improve their learning outcome.

Adaptive educational hypermedia is the answer to the "lost in hyperspace" syndrome, where the learner has normally too many links to choose from, and little knowledge about how to proceed and select the most appropriate ones to him/her.

The domain of AEH 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. Fig 1 summarizes the Brusilovsky's taxonomy of adaptive hypermedia technologies [1,2].

The year of 1996, the start of the rapid increase in the use of the Word Wide 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. [1,2].

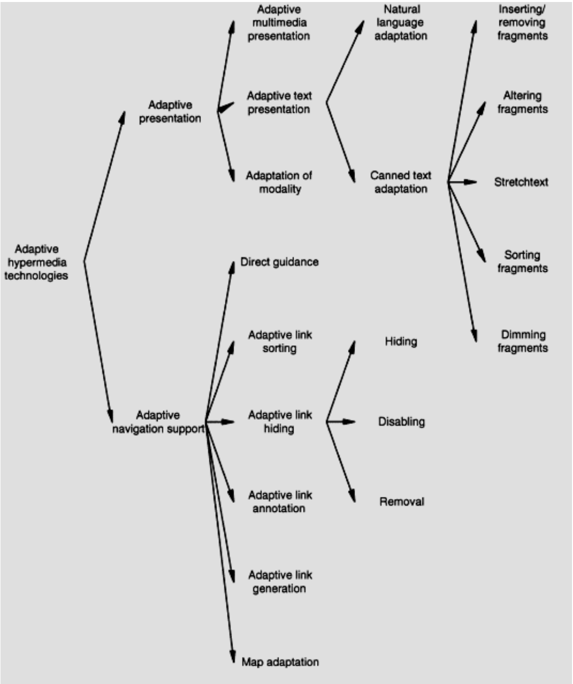


Fig 1. Brusilovsky's taxonomy of adaptive hypermedia technologies.

Despite the efforts carried out during the last few years, the AEH systems development remains a difficult task to undertake. This task requires often the constitution of interdisciplinary group. Experts from different fields such as education and psychology must cooperate with computer engineers to design such systems.

In order to improve the productivity in this domain and allow a wide community to be involved, AEH authoring systems are used and some of them allows the users to develop adaptive hypermedia courses, sometimes, without knowledge in programming art. Thus, the task is reduced in a way that the teachers need only to introduce course material into a generic AEH predetermined by the system.

Authoring tools can enable, encourage, and assist authors in the creation of accessible content through prompts, alerts, checking and repair functions, help files and automated tools. It is just as important that all people be able to author content as it is for all people to have access to it. The tools used to create this information must therefore be accessible themselves.

The authoring tool may be accessible to authors regardless of disability, it produces accessible content by default, and it supports and encourages the author in creating accessible content. Because most of the content of the Web 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

Some of authoring systems are discussed in [3,4], other examples of authoring tools are: [5,6]. However, all these authoring systems were designed to work in a single-user mode.

Recently, thanks to the networks and groupware, virtual meetings involving many people are made possible. Several works in this area are already available in such domains as the cooperative writing [7,8,9], the multimedia, the cooperative design of objects, etc. The common point between all these systems is that they allow several participants to work together in synchronous or asynchronous manner to realize a common task.

Since the cooperative aspect, through a computer network, has been experimented successfully in a lot of domains, this leads us to think that it would be desirable that the designers of authoring tools should integrate this cooperation functionality for AEH production. This is the object of this paper. We investigate this idea through an authoring system called TALABAH (Teaching And LeArning By Adaptive Hypermedia).

We organize the rest of this paper as follows. Section 2 summarizes briefly the concept of accessibility and the authoring systems. Section 3 presents the general concept of authoring systems and discusses the different approaches used when designing cooperative authoring tools; the organizational aspect of our system TALABAH will be presented in this context. Section 4 describes the courseware model and the design of the adaptive hypermedia based course generated by this authoring tool. Section 5 presents the architecture of TALABAH and the different levels and the whole functionalities it covers. Section 6 describes the system implementation and shows some experimental results and discussions. Finally, Section 7 briefly concludes this paper.

2. Accessibility and authoring systems

There is a huge advantage in using authoring tools to create content. In theory, such tools actually promote Web accessibility by allowing easy access to Web content contribution from individuals without expertise in Web authoring.

However, content created by authoring tools can present problems. Often, they do not promote insertion of accessibility features such as alternative text for images. The lack of awareness of many content providers in accessible design issues is accentuated by the relative failure of popular authoring tools to promote the creation of accessible resources.

The W3Cs Authoring Tool Accessibility Guidelines (ATAG) [19] provides a checklist of features with which authoring tools should comply in order to ensure that the Web content they produce is as accessible as possible. A similar effect is noticeable in authoring tools aimed specifically at the learning technology sector, and accessibility of courseware authoring tools is now being addressed.

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 [20].

3. From individual authoring to cooperative authoring

The most important shortcoming of an AEH is the authoring part. Developing knowledge space in AEH is not simple and it is very time consuming. In addition, courseware is usually non transferable and non reusable [10]. Some research has been done to address the problems by developing generic authoring systems, for example My Online Tutor [10,11], based on LAOS authoring model [12, 13] that can be delivered by many AEH systems like AHA [14] and WHURLE [10, 15].

Therefore, several works has been taken on the design and implementation of AEH authoring systems during this last decade. Murray [2,3] 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: (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.

Given that AEH is often described as having four main components (domain model, adaptive model, learner model, and learner interface), the authoring systems must therefore theoretically include all the necessary tools for building these components. However, it has to be recognized that, very few systems requires from the author to construct every thing needed, the major systems are usually limited to tools for building one, two or at limit three components among the four. The remaining components are generally predefined in a pattern of AEH and the author is solicited only to introduce necessary parameters for their functioning.

TALABAH, the system presented in this paper, generates an adaptive based course that we classify as first and seventh category of the Murray classification mentioned above. This category of authoring systems generally structure the learning material as a network of Learning Units (LUs) where every LU satisfied some educational objectives. The LUs are linked together to show prerequisite-relations between them. Although that 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, given that AEH systems rely generally on large knowledge bases and subject expertise, it would make sense to develop them collaboratively. The models support collaboration works on domain related knowledge for adaptive learning.

To develop a cooperative AEH authoring system, several approaches can be proposed. We can classify them in two large categories [16]. A first approach, pragmatic, and more economic in implementation effort, consists to take an existing single-user authoring system and enrich it with other functionalities that makes it cooperative one. However, the rigidity induced by knowledge acquisition units of the single-user authoring systems, makes it very difficult to take into account group awareness control and the distributed management of the knowledge base. The produced authoring tools will lack certainly effectiveness and will use cooperation mechanisms only at a limited degree.

The second method, which we adopted in the design of TALABAH, consists in taking into account the paradigm of cooperation and the needed tools to do it, at the design step of the system architecture. This approach, although expensive, allows us to apply rigorously the mechanisms of the cooperation metaphor. Though, we must provide through this software architecture, a common work-space to the authors involved in the cooperative construction of an AEH. 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 AEH.

This organization facilitates also the manipulation of different components of the AEH during all steps of the project advancement. So, we define three roles through which the authors can participate during the AEH building process: main author, constructor coauthor and commentator coauthor.

- The role of the main author is to coordinate the whole work and to verify that the calendar is well respected. He defines the AEH logical structure to be produced by decomposing it in several components (chapters, LU, figures, images, etc.), then he affects the roles to different co-authors. He has free access to all AEH components.

- A constructor coauthor is authorized to create, modify or delete only the components assigned to him. On the remaining AEH components he will have only the role of commentator.

- A commentator coauthor is authorized only to read and /or comment the components assigned to him.

3. The adaptive hypermedia based course

3.1. Courseware model

Two learning modes are presented to the learner in the adaptive learning environment (ALE): "information mode" (free exploration) and "training mode" (learning with auto-evaluation). The learning process is organized around adaptive hypermedia components. The learning material is structured in three abstraction level hierarchy according to three level hierarchy of learning objectives defined in [17]: parts (satisfying the general objectives), chapters (satisfying the specific objectives) and the Hypermedia Learning Units (HLU) (satisfying the operational objectives).

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. 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 AEH created by TALABAH.

The instantiation process, producing "Generated Rules" (GRules), is carrying out automatically by the system on the base of parameters delivered by authors. These AEH parameters which are represented in predicates form, describe the quantitative aspect of the teaching material (number of parts, number of chapters, number of 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 (HLUs). These HLUs have no knowledge about the AEH domain. They are supposed to receive all kinds of knowledge about the domain via instantiation, under all media types that are allowed by the X/HTML language (text, image, sound, video, applet).

To summarize, we can consider, two levels of knowledge in the curriculum definition:

- **Level 1.** A higher level corresponding to the tutoring plans: These plans consist of five sets of rules that invoke HLU of the lower level. 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 HLUs assumed to be understood after a negotiation phase; "planning" the learning session; "searching and filtering" the content of HLU; and finally "auto-evaluation".

- **Level 2.** A lower level corresponding to the HLU space: This space consists of a hierarchical network that is constituted of six HLU sub-levels where the first four sub-levels correspond to the courseware-type HLU (module abstract, part abstract, chapter abstract, HLU classes) and the last two sub-levels correspond to evaluation-type HLU (questions and exercises).

3.2. Adaptive learning environment architecture

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

1. A "free-exploration module" that allows the learner to navigate freely through the different HLUs, as a book.

2. Three modules representing the "training mode":
   - A "domain-expert module" using generated rules to search and filter concept-indexed HLU asked by a pedagogical module at a given moment.
   - A "pedagogical module" that allows the negotiation of learning session objectives with the learner and generates in turn sequencing plan for the adaptive presentation of the lesson. Two sub-modules realize these two tasks: the "negotiator" using the negotiation generated rules and the "planer" using the planning generated rules.
   - A "diagnosis module" that allows the learner evaluation and the maintenance of an overlay type learner model. This module is made up of three sub-modules: an "evaluator" using evaluation generated rules; a "deduction agent" using the deduction generated rules; and a "learner model manager" managing its persistent content.

3. A "supervisor module" that allows on one side, the communication with the learner, and on another side, the coordination between the three modules: domain-expert module, pedagogical module and diagnosis-module. This coordination is carried out via message sending.

4. The cooperative authoring of the course

In order to motivate more authors to use the adaptive hypermedia, 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 test questions. 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 for concept networks, which will enable the authors to define the prerequisite relationships with a drag-and-drop interface.

From an author point of view, building a courseware using TALABAH consists in introducing, via a cooperative editor, a set of objects that will be manipulated in the adaptive learner environment (ALE). These objects are made up with learning material in the form of hypermedia learning units (HLU), prerequisite-network in the form of an oriented graph, course parameters in the form of predicates and pedagogical knowledge in the form of production rules.

The cooperation task in TALABAH is introduced at the editing level of the teaching material and at the editing level of the prerequisite-network. These two components are well structured: the teaching material is organized as parts, chapters and HLU, and the prerequisite-network is organized as sub-networks form (part-prerequisite network, chapter-prerequisite network, HLU-prerequisite-network and concept-prerequisite network).

These structures are well convenient for the fragmentation and then constitute the basis of our cooperative editing approach as in Alliance [7]. The two concept-keys on which is based the design of TALABAH are the “fragmentation” and “edition roles” [7]. As previously said, we defined three edition roles of participation for the authors: main author, constructor coauthor and commentator coauthor.

At the beginning of the course construction task, a negotiation step is necessary. The main author assigned the edition roles to different co-authors around different fragments of course structure in accordance with their competences and availability. Five learning principles had been incorporated into the authoring process [18]: a clear definition of educational objectives, 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 educational objectives.

4.1. Cooperation modes and group awareness

The cooperative developing process of the course is characterized by a steps-sequence during which the authors can work either individually or collectively. In this way, we defined three cooperation modes: individual responsibility, alternate version and collective responsibility. The first two modes are typically asynchronous cooperation modes. Especially, the second one is inspired from the real principle “let us reflect separately on the question and then compare our results after”.

The last one is a typically synchronous mode that allows, to relatively reduced number of authors chosen by main author, to finalize the course version when the project reaches its final phase [21]. The notification and group awareness functions constitute an important point in the cooperative application design [22]. It includes all the interface functions and all systems functions that allow the users to perceive the activities of the other users, as well as to control and to act on the distributed environment.

4.2. The cooperative editor architecture

The cooperative editor is organized according to centralized client/server architecture where all the communications pass automatically through the central site (the server). We associate to every client-site a client process (CPR) that accomplishes all the tasks that are processed locally (the editing tasks for example). We define a server process (SPR) that manages all the communications between the different CPRs and keeps up to date the content of the course central copy and the course logical structure.

The software architecture offers several functionalities that we can decompose them in three layers: server layer, editor layer and presentation layer. Every layer is structured as a collection of modules where each module consists of several objects implementing some functionalities (see Figure 2 for client side and Figure 3 for server side).

The need for information exchange between the two client layers on one side, and between the client and the server on the other side, implies the presence for “dialog controllers”. We interpose therefore between every presentation layer and every editor layer a Dialog Controller (DC), and between the server layer and every editor layer a Main Dialog Controller (MDC). Messages exchanged between layers transits automatically by the dialog controllers.

According to message-type, the convenient objects are then executed among those that are defined in a layer.

1 - Presentation Layer (Figure 2): 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.

2 - Editor Layer (Figure 2): This software layer gathers many types of functionalities allowing every author to manipulate the objects that constitutes the course. These functionalities include not only the support of individual actions, but also the sharing aspect and transparency management. For example, the access to a file in a single user editor delivers directly its content. But in our case, this process consists in several tasks such as access rights verification, locking state of the object and warning the authors working on this file in the same time.

At each author site, some associated functionalities allow the author to save locally the objects that are accessible to him. He will solicit regularly the server to update the versions of these objects. The components of the editor layer are:

a) **HLU/prerequisite-network-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-network 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. Four levels are used in the network. One level shows the concept-prerequisites, the second shows parts-prerequisites, the third one shows the chapter-prerequisites of a particular part, and the fourth level shows the HLU-prerequisites of a chapter.

b) **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 HLUs.

c) **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.

d) **Verification module:** As most authoring systems, TALABAH 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.

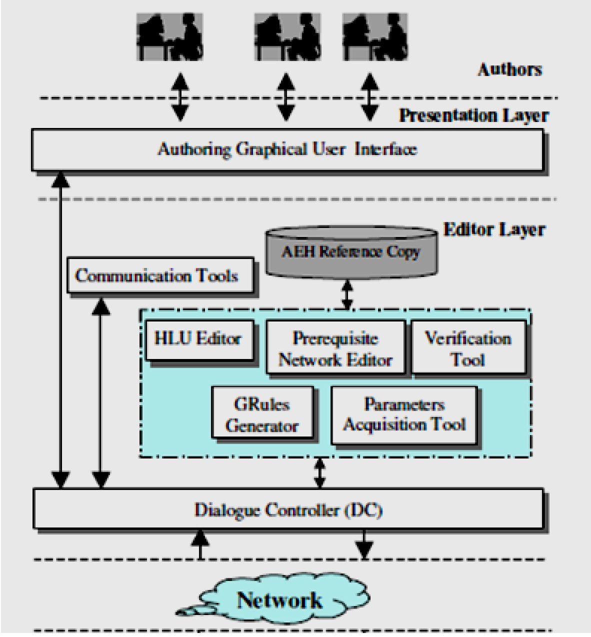


Fig 2. Architecture of the system - Client Side

3 - Server Layer (Figure 3): This software layer gathers several types of functionalities, among which those that concern the course logical structure management, as well as the content of the course components. They allow the authors to save and retrieve course objects whose logical structure is declared, as well, at the central level as at the local level. This software layer is responsible for access rights control, events handling and events notification. In the case of events notification, for example, the concerned module manages a set of queues such as engagement queue, locking-queue, etc. At every time, if an event occurs, this process identifies the concerned authors and proceeds to structure the notifications as a message form to transmit. These messages then will be made available to another sender module that sends the message.

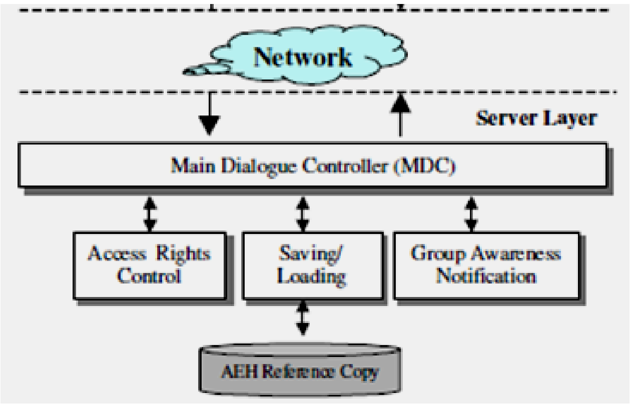


Fig 3. Architecture of the system - Server Side

4 - Dialogue Controllers DC and MDC (Figure 4): Each dialog controller is composed of three independent modules performing respectively, “message reception”, “message control” and “message sending”. The Control module allows the coordination and synchronization of the running of the different modules within the three layers, in accordance with the actions of the different authors. At any time, it used all necessary information to determine exactly what are the functionalities to invoke within the layers for which it is responsible. Every time that an event occurs, the associated receiver 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.

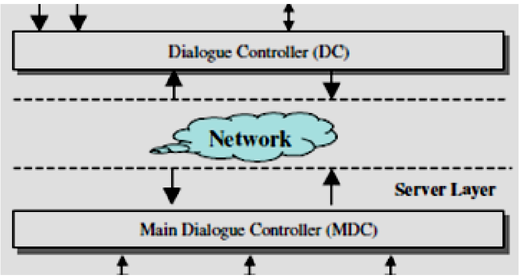
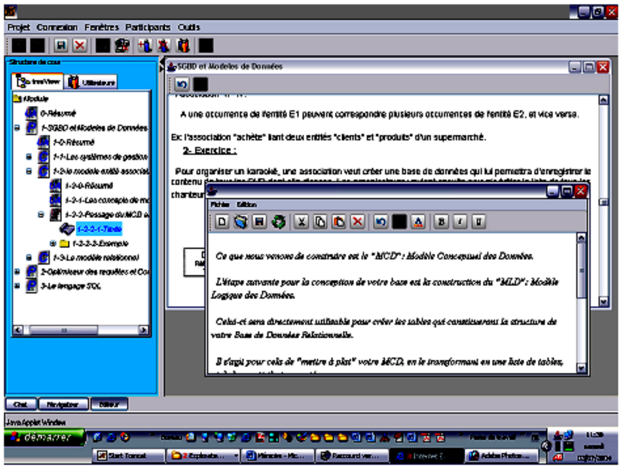


Fig 4. Dialogue Controllers between server and client layers

5. Discussion

The AEH (course) pattern is implemented in PHP/MySQL and resides on a server; it can therefore be accessed simultaneously by different distant learners. The authoring tool, implemented in JSP and Java, is organized as centralized client-server architecture (Figure 5 shows an interface screen). It makes it possible to several authors to be connected to a working session characterized by a cooperative space and a control strategy. The cooperation 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.


Projet Connexion Fenêtres Participants Outils
SGBD et Modeles de Données
Fichier Edition
démarrer

A une occurence de l'entité E1 peuvent correspondre plusieurs occurences de l'entité E2, et vice versa.

Ex: l'association "achète" liant deux entités "clients" et "produits" d'un supermarché.

2- Exercice :

Pour organiser un karaoké, une association veut créer une base de données qui lui permettra d'enregistrer le contenu

Ce que nous venons de construire est le "MCD" : Modèle Conceptuel des Données.

L'étape suivante pour la conception de votre base est la construction du "MLD" : Modèle Logique des Données.

Celui-ci sera directement utilisable pour créer les tables qui constitueront la structure de votre Base de Données Relationnelle.

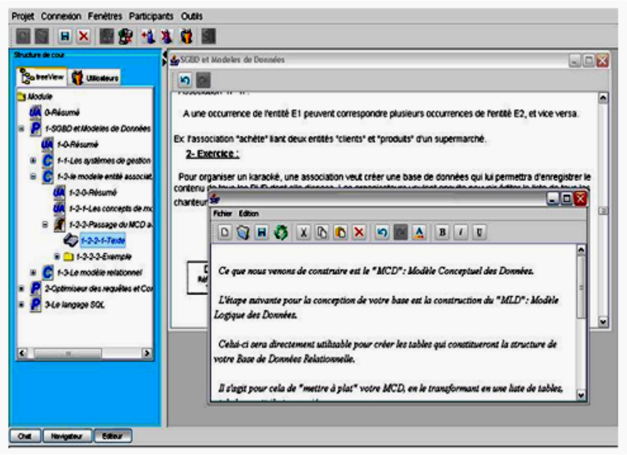
Il s'agit pour cela de "mettre à plat" votre MCD, en le transformant en une liste de tables,

Fig 5. An interface-screen of the authoring tool

The control strategy manages the negotiation of the access right to a component of the AEH and then participation of users during the work session. Five learning principles had been incorporated into the authoring process [19]. These principles are: a clear definition of educational objectives, 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 objectives.

Two different approaches were used to test the validity that the system actually incorporated pedagogy and effective cooperative design concepts as part of the developmental process. To evaluate the system, a group of four 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”, via local network, and were surveyed to seek their opinion if the authoring system offers all cooperative tools necessary to construct the course in a synchronized manner (Figure 6). We were also interested in the group interaction through accounting of various exchanges operated between the authors during a work session. Especially, we record the aspects related to notification and group awareness.


Projet Connexion Fenêtres Participants Outils
Structure de cour
treeView
Utilisateurs
Module
0-Résumé
1-SGBD et Modeles de Données
1-0-Résumé
1-1-Les systèmes de gestion
1-2-le modele entité associat
1-2-0-Résumé
1-2-1-Les concepts de mc
1-2-2-Passage du MCD a
1-2-2-1-Texte
1-2-2-2-Exemple
1-3-Le modèle relationnel
2-Optimiseur des requêtes et Co
3-Le langage SQL
SGBD et Modeles de Données
Fichier Edition
Chat Navigateur Editeur

A une occurrence de l'entité E1 peuvent correspondre plusieurs occurrences de l'entité E2, et vice versa.

Ex: l'association "achète" liant deux entités "clients" et "produits" d'un supermarché.

2- Exercice :

Pour organiser un karaoké, une association veut créer une base de données qui lui permettra d'enregistrer le contenu

chanteur

Ce que nous venons de construire est le "MCD" : Modèle Conceptuel des Données.

L'étape suivante pour la conception de votre base est la construction du "MLD" : Modèle Logique des Données.

Celui-ci sera directement utilisable pour créer les tables qui constitueront la structure de votre Base de Données Relationnelle.

Il s'agit pour cela de "mettre à plat" votre MCD, en le transformant en une liste de tables,

Fig 6. One “Data bases” course developing screen

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.

6. Conclusion

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 presented an authoring tool that assist disabled users to access teaching and learning activities over the web. The cooperative authoring system, called TALABAH, is designed to satisfy guidelines of accessibility of the W3C recommendation for disabled authors and learners especially with mobility impairments. Integrating cooperation paradigm in AEH authoring systems is the original idea of this paper. This authoring tool allows geographically distant disabled authors to cooperate to produce an accessible courseware according to a predefined course pattern.

Acknowledgments

The work reported in this paper is supported by the CNEPRU project funded by the Algerian Ministry of Higher Education and Scientific Research via contract no. B*01320080006.

References

[1] P. Brusilovsky, “Adaptive hypermedia. methods and techniques of adaptive hypermedia”. International Journal of User Modeling and User-Adapted Interaction, 11 (1/2), 2001, pp. 87-110.

[2] P. Brusilovsky, 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, 2003 , pp. 377-409.

[3] T. Murray, “Authoring knowledge-based Tutors: Tools for content, instructional strategy, student model and interface design”, Journal of the Learning Sciences, vol. 7, n° 1, 1998.

[4] T. Murray, “Authoring Intelligent Tutoring Systems: an analysis of the state of the art”, International Journal of AI in Education, vol. 10, 1999, pp. 98-129.

[5] A. Cristea, “Adaptive Course Authoring: My Online Teacher”, in Faculty Mathematics and Computer Science, TU Eindhoven, 2003.

[6] P. De Bra, “AHA! The Adaptive Hypermedia Architecture”, in the ACM Hypertext Conference, Nottingham, UK, 2006.

[7] D. Decouchant, and A.M. Martínez, “A Cooperative, Deductive and Self-Adaptive Web Authoring Environment”, In Proceedings of Mexican International Conference on Artificial Intelligence (MICAI-2000), Springer Verlag, Acapulco (Mexico), 2000, pp. 443-457.

[8] F. Pacull, A. Sandoz, and A. Schiper, “Duplex: a distributed collaborative editing environment in large scale”, Proceedings of the ACM Conference CSCW’94, ACM Press, 1994.

[9] A. Zidani, M. Boufaida, and M. Djoudi, “JamEdit: un outil interactif et coopératif pour l'édition coopérative de documents”, Revue Technique et Science Informatiques, Vol. 19, n°1, Hermes, 2000, pp. 1-23.

[10] C. Stewart, A. Cristea, and T. J. Brailsford, “Authoring Once, Delivering Many: Creating Reusable Adaptive Courseware”, in 4th IAESTED International Conference on Web Based Education (WBE'05), Grindewald, Switzerland,. 2005.

[11] F. Ghali, “Collaborative Adaptation Authoring and Social Annotation in MOT”, Warwick Postgraduate Colloquium in Computer Science, 2008.

[12]. A. Cristea, and A.D. Mooij, “LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators”, in WWW 2003. ACM Budapest, Hungary, 2003.

[13] N. Weibel, “Towards Adaptive Hypermedia Authoring from the Dexter Model to Laos”, Institute for Information Systems, ETH Zurich., 2006.

[14] A. Cristea, D. Smits, and P. de Bra, “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] M. Meccawy, "WHURLE 2.0: Adaptive Learning Meets Web 2.0", in Third European Conference on Technology Enhanced Learning, Springer: Maastricht, The Netherlands, 2008.

[16] S. Talhi, M. Djoudi, and A. Zidani, "Un système auteur de tuteurs intelligents : évolution du mono-usager vers la coopération", Revue Techniques et Sciences Éducatives, Volume 8, n° 1-2, 2001, pp. 127-138.

[17] D. Hameline, Les objectifs pédagogiques en formation initiale et en formation continue, Edition ESF, 8ième édition, Paris, 1990.

[18] T. Janicki, and Jens O. Liegle, "Development and evaluation of a framework for creating web-based learning modules: a pedagogical and systems perspective", JALN Journal, Volume 5, Issue 1. 2001.

[19] W3C, "Authoring Tool Accessibility", Guidelines, available from: www.w3.org/TR/ATAG10., 2000.

[20] J. M. Slatin, and S. Rush, Maximum Accessibility: Making Your Web Site More Usable for Everyone, Addison Wesley Editor, 2002.

[21] S. Talhi, M. Djoudi, and M. Batouche, "Authoring Groupware For Intelligent Tutoring Systems", Information Technology Journal (ITJ), vol. 5, n°. 5, 2006, pp. 860-867.

[22] A. Muhammad, A. M. Martínez, and D. Decouchant., "Awareness and Coordination for Web Cooperative Authoring". In Proceedings of AWIC'2005, The 3rd International Atlantic Web Intelligence Conference, Lecture Notes in Artificial Intelligence, no 3528, Springer Verlag, Lodz, Poland., 2005, pp.327-333.

Said Talhi received a PhD in Computer Science from the University of Batna, Algeria, in 2007 and he is currently an Associate Professor at the University of Batna, Algeria. 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 Europennes) publisher in 2011.

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.