ICTA'07, April 12-14, Hammamet, Tunisia

Authoring Intelligent Tutoring Systems for Disabled Learners

Said Talhi[1], Mahieddine Djoudi[2], Salima Ouadfel[1], Samir Zidat[1]
[1]Department of Computer Science, University of Batna, Algeria
[2]Laboratoire SIC & Equipe IRMA, University of Poitiers, France
E-mails: s_talhi@yahoo.fr ; mahieddine.djoudi@univ-poitiers.fr
souadfel@yahoo.fr ; samir_zidat@yahoo.fr

Abstract

The purpose of this paper is to present an authoring tool for the development of a distance learning hypermedia intelligent tutoring system (HITS). 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 cooperate to produce such tutors together. It consists of a shared workspace gathering all tools necessary to the cooperative development task. After describing the structure of the generated ITS, the client-server architecture of the authoring tool and the mechanism needed to manage the notification and group awareness is outlined.

1. Introduction

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

The Intelligent Tutoring Systems [1] development remains a difficult task to undertake, despite all the efforts carried out during the last few years. This interdisciplinary task often requires the cooperation between experts from different fields such as education, psychology and computer science engineering to design such systems. In order to improve the productivity in this domain and allow a wider community to be involved, authoring systems allow the users to develop Intelligent Tutoring System (ITS), sometimes, without writing a single line of code. Thus, the task is reduced in a way that the experts need only to introduce knowledge to generic ITS predetermined by the system. Some of these authoring systems are discussed in [2], however, they were all designed to work in a single-user mode.

Recently, thanks to the networks and groupware, virtual meetings involving many people are made possible [3][4]. Several works in this area are already available in such domains as the cooperative writing [5][6], 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 future authoring tools should integrate this cooperation functionality for ITSs production.

This is the purpose of this paper. We investigate this idea through an authoring system dedicated to Algerian universities called TALHITS (Teaching And Learning by Hypermedia Intelligent Tutoring Systems). The system allows 10% of the students that present disabilities to learn at home via Internet.

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) [7] 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 [8].

2.1. Single-user authoring systems

During this last decade, several works has been taken on the design and implementation of ITS authoring systems. Tom Murray [2] (Murray, 1999) listed more than twenty references in his recent paper and presented his authoring system named Eon [9] (Murray, 1998). These systems are classified in seven categories according to the type of ITSs 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 ITSs are often described as having four main components (domain module, tutoring module, student model, and student 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 require from the author to construct every thing needed, as in the case of Eon system for example. The other systems are usually limited to tools for constructing one, two or at limit three components among the four. The remaining components are generally predefined in an ITS pattern and the author is solicited only to introduce necessary parameters for their functioning.
TalHits, the system presented in this paper, offers cooperative functionalities to the authors and generates ITSs classified as first and seventh category of the Murray classification mentioned above. These authoring system types, structure generally the teaching 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 these systems do not use any explicit representation of domain knowledge, 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 [10] [11].

2.2. Towards a cooperative authoring system

Several approaches can be proposed to develop a cooperative authoring system [12]. We can for example take an existing single-user authoring system and enrich it with other functionalities that makes it cooperative one. But the produced authoring tools will lack certainly effectiveness and will use cooperation mechanisms only at a limited degree.
The second method, which we adopted consists in taking into account the paradigm of cooperation and the needed tools to do it, at the design step of the system architecture.
However, we should notice that the software component does not constitute the only parameter in the success of such cooperative system. Also, we must have to take into account the human factors involved due to the group activities because of their importance [13]. Thus, to avoid the inherent conflicts due to the human nature, we propose a group organization that facilitates the ITS development process.
This organization facilitates also the manipulation of different components of the ITS during all steps of the project advancement. So, we define three roles through which the authors can participate during the ITS development process: main author, constructor coauthor and commentator coauthor.

– The main author coordinates the whole work. He defines the ITS logical structure to be produced, by decomposing it in several components (parts, chapters, LU, figures, images, etc.) and then assigns the roles to different co-authors. He has free access to all ITS components.
– A constructor coauthor is authorized to create, modify or delete only the components assigned to him. On the remaining ITS 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. Generated hypermedia ITS

The distance-learning environment developed consists of two learning modes offered to the learner: "exploration mode" (free exploration) and "training mode" (learning with self-evaluation) managed by the intelligent tutor [14].

The architecture of the generated ITS is similar to that of a traditional tutoring system. It is composed of the common three modules: the pedagogical module, the expert module and the learner model module.

The difference between our ITS and a classical ITS is the fact that our ITS resides on the server and can therefore be accessed distantly by learners. Its implementation in Php/MySql has helped in spreading its use as a distance learner system.

The teaching material is structured in three abstraction level hierarchy according to three level hierarchy of pedagogical objectives defined in [15]: 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 adapting it to each learner capacities, the management of these components is ensured by an expert system based on a 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 ITS created by TalHits.

The instantiation process, producing "generated rules" (GRules), is carrying out automatically by the system, based on parameters delivered by authors. These ITS 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 the goal of reusability and independence from the domains, the MRules invoke abstracted structures called HLU. These HLUs have no knowledge about the ITS domain. They are supposed to receive all kind of knowledge about the domain via instantiation, under all media types that are allowed by the HTML language (text, image, sound, video, applet).

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

– 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 with the learner for the entry point in the course (where to start) 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 auto-evaluation.
– A lower level corresponding to the HLU universe: This universe consists of a hierarchical network constituted of six HLU sub-levels where the first four sub-levels correspond to the course type HLU (module abstract, part abstract, chapter abstract, HLU classes) and the last two sub-levels correspond to evaluation type HLU (questions and exercises).

4. Cooperative authoring tool

To be efficient, cooperation requires not only a shared space, but also some support. The shared space must be structured, each user must be aware of the activity of others, some direct communication between people should be provided to allow them to discuss their common task, and coordination means should be provided.

In "authoring mode", we have proposed some necessary tools for ITS cooperative authoring. From an author point of view, designing an ITS using TalHits consists in introducing, via a cooperative editor, a set of objects that will be manipulated in the "learner mode".

These objects are made up with teaching material in the form of hypermedia learning units (HLU), prerequisite-network in the form of an oriented graph, ITS parameters in the predicates form and pedagogical knowledge in the form of production rules.

The cooperation task in TalHits 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 [5]. The two concept-keys on which is based the design of TalHits are the "fragmentation" and "edition roles" [16]. 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 ITS construction task, a negotiation step is necessary. The main author assigned the edition roles to different co-authors around different fragments of ITS structure in accordance with their competences and availability. Five learning principles has been incorporated into the authoring process [10]. These principles are: a clear definition of pedagogical objectives, definition of pre-requisite knowledge, providing a variety of presentation styles (tell, show and do), enhanced feedback and testing and, finally, permitting the learner to control the direction of the learning session by choosing himself the pedagogical objectives.

4.1. Cooperation modes and group awareness

The cooperative building process of ITS 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 allow, to relatively reduced number of authors chosen by main author, to finalize the ITS version when the project reaches its final phase [12].
The notification and group awareness functions constitute an important point in the cooperative application design [17]. 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. Software architecture

The cooperative editor is organized according to centralized client/server architecture [18]. So, all the communications transits automatically through the server that keeps up to date the content of the ITS central copy and the ITS logical structure.
The software architecture offers several functionalities that we can decompose in three layers: server layer, editor layer and presentation layer (see Figure 1).
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).

Server layer: This software layer gathers several types of functionalities, among which those that concern ITS logical structure management, as well as the content of ITS components. They allow the authors to save and retrieve ITS 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. 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.

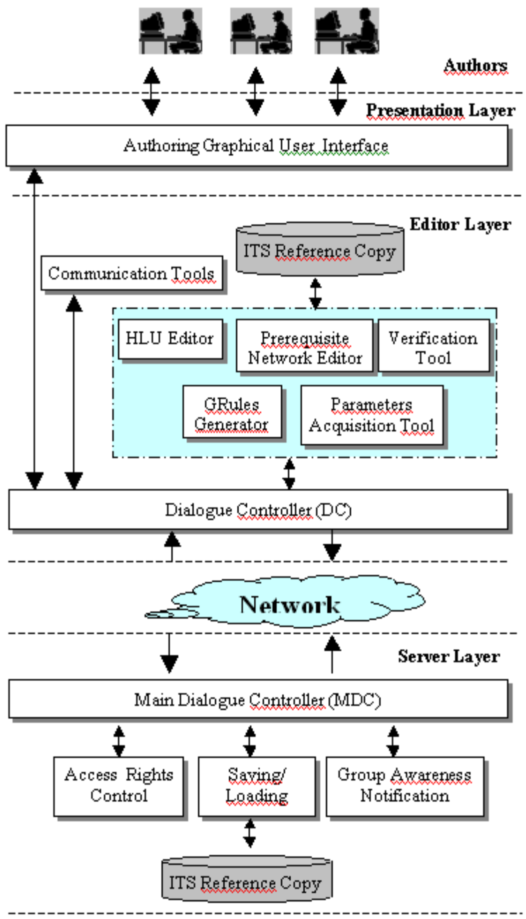

Figure. 1. Architecture of TalHits

Editor Layer: This software layer gathers many types of functionalities allowing every author to manipulate the objects that constitutes the ITS. These functionalities include not only the support of individual actions, but also the sharing aspect and transparency management.

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 different components of the editor layer are:

1. HLU/prerequisite-network-Editor: Two modules are designed to implement this component software. They allow the creation task and the maintenance of different ITS objects. The first module allows the wisiwig HLU edition using 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.

2. Parameters acquisition module: This module allows the main author to specify the ITS parameters that indicate the manner in which the teaching material is decomposed (number of parts, chapters in every part, HLU in every chapter, etc.). These parameters are saved in the predicates form and then used to instantiate the main rules.

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 ITS parameters introduced via the previous module, this generation consists of an instantiation of the five packages of the MRules.

4. Verification module: As most authoring systems, TalHits offers a tool to help the author in the diagnosis of bugs. It facilitates detection of incoherencies that can be occurred during the ITS construction.

Presentation Layer: This layer gathers an organized set of interactive objects defining the graphical user interface (buttons, icons, cursor, scrolling bar, task-progression 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.

Dialogue Controllers DC and MDC: 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.

4.3. Implementation

The ITS pattern is implemented in Php/MySQL as a web based application; it can therefore be accessed simultaneously by different distant learners. The "authoring mode" software, implemented in Java Server Pages and Java language, is organized as centralized client-server architecture. It makes it possible to several authors to be connected to a working session characterized by a cooperative space and a control strategy. Figure 2 shows one interface-screen of the "authoring mode".

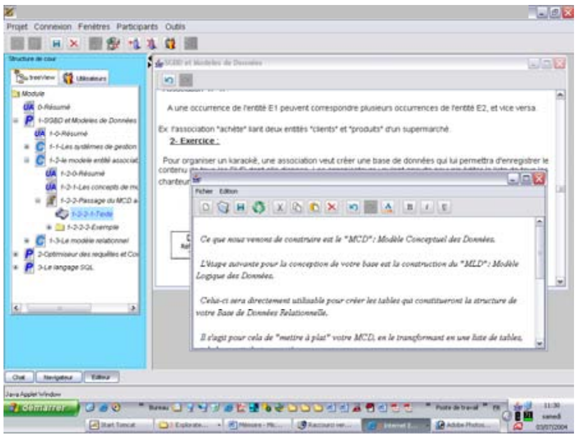

Figure 2. An interface-screen of "authoring mode"

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

To evaluate the system, five disabled teachers geographically dispersed were invited to develop a data base tutorial, via local network, and were surveyed to seek their opinion if the authoring system offers all cooperative tools necessary to construct the tutorial in a synchronized manner.

Although the system has exhibit preliminary 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.

6. References

[1] Wenger E., "Artificial Intelligence and Tutoring Systems", Addison-Wesley Eds., USA, 1987.

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

[3] Attaran M. and Attaran S., "Collaborative computing technology: the hot new managing tool", Team Management: An Int. Jor., 8(1), 2002, pp. 13-20.

[4] Mills K. L., "Computer-Supported Cooperative Work Challenges", Enclyclopedia of Library and Information Science (2nd Edition), Marcel Dekker, New York, 2003, pp. 678-684.

[5]Decouchant D. and Martínez A. M., "A Cooperative, Deductive and Self-Adaptive Web Authoring Environment", In Proc. MICAI-2000 Mexican Int. Conference on Artificial Intelligence, Springer Verlag, Acapulco (Mexico), 2000, pp. 443-457.

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

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

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

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

[10] Janicki T., 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.

[11] Nkambou R., Frasson C. and Gauthier. G., "CREAM-Tools : An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems". In Murray T., & al.. (Eds): Authoring Tools for Adv. Tech. Learning Env.: Toward cost-effective adaptative, interactive, and intelligent educational software, Kluwer Publishers, 2003, pp. 93-138.

[12] Talhi S., Djoudi M., and Batouche M., "Authoring Groupware For Intelligent Tutoring Systems", Information Technology Journal (ITJ), Vol. 5, No. 5, 2006, pp. 860-867.

[13] Greenberg S. and Roseman M., "Groupware Toolkits for Synchronous Work", In M. Beaudouin-Lafon, editor, Computer-Supported Cooperative Work (Trends in Software 7), Chapter 6, John Wiley & Sons Ltd, 1999, pp.135-168,

[14] Talhi S., Djoudi M., Zidat S. and Batouche M., "Un système tuteur intelligent hypermedia pour l'apprentissage à distance asynchrone", Congrès international en informatique appliquée, CIIA'05, Bordj Bou Arréridj, 2005, Algérie.

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

[16] Decouchant D., Martínez A. M. and Martínez E., "Documents for Web Cooperative Authoring", In Proc.CRIWG'99, 5th CYTED-RITOS International Workshop on Groupware, IEEE Computer Society, Cancun (Mexico), 1999, pp. 286-295.

[17] Muhammad A., Martínez A.M. and Decouchant D., "Awareness and Coordination for Web Cooperative Authoring". In Proc. 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,

[18] Orfali R., Harkey D. and Edwards J., "Essential Client/Server Survival Guide", John Willeys &Sons Inc., 2nd edition, New York, 1997.