A support architecture proposal to hypermedia intelligent tutoring systems development in the Internet

Imad SALEH*, Fabrice PAPY**, Nasser BOUHAÏ*

Université Paris 8, Equipe Hypertexte dynamique, Département Hypermédia, Documentation***

2 Rue de la liberté, 93526 Saint-Denis Cedex 2,

Tél. : 01 49 40 67 58, Fax : 01 49 40 67 83

Email : isaleh@wanadoo.fr, nasser@labart.univ-paris8.fr, papy@labart.univ-paris8.fr

Abstract

This paper presents a support architecture to hypermedia intelligent tutoring development in the internet. A concrete implementation is discussed, some difficulties are figured out, and an architecture model is suggested.

1. Introduction

The relationship between hypertext systems and intelligent tutors, seems to inherit the polemic discussion existing (contradictory or complementary) between intelligent learning environments and ITS (Intelligent Tutoring Systems). Tutors are capable to control adaptive learning in several levels but generally can not provide support for free exploration. On the other hand, learning environments and microworlds support exploratory learning but are limited in the tutorial control activities. In other words, the excess of freedom given to the learner can hide important topics regarding the subject being learned.

However, it is important to mention that intelligent tutors and hypertext systems do not offer contradictory but complementary roads. The hypertext, through its non-linear structure and by allowing the navigation control to be learner guided, supplies a free discovery learning atmosphere, very desirable in certain cognitive styles. However, it introduces a problem on the individual abilities in cognitive styles that prefer to be guided. That mechanism of guided navigation should be controlled by the tutorial component of the system, facilitating to the learner the alternated experience of free exploration and discovery, and of instruction (explanation, exemplification, evaluation, advising, etc), setting up an atmosphere with characteristics similar of the ones found in GDL(Guided Discovering Learning) environments.

Since the Aspen project [1], which was the first project to implement a hybrid architecture mixing a training environment and a hypermedia system, very few projects have been concluded involving these two topics. However, the Internet's popularization added a new dimension to this discussion due to existence of two interesting characteristics that deserve to be pointed: first it has a democratic and non-bureaucratic structure permitting an educational system to reach worldwide boundaries, and second its World Wide Web service presents a materialization of a hypertext/hypermedia environment. This new paradigm turns feasible the already mentioned hybrid solution. However, two main problems remain unsolved:

- the Internet architecture still imposes some difficult barriers to effective multimedia presentation
- an ITS structure needs to be built in such a way that permits the stable coexistence of guide instruction and free navigation.

The objective of this article is to propose a distributed multimedia architecture that addresses both problems mentioned above, supporting the development of ITS in the Internet. The article is divided as follows: Section 2 presents a concrete example, an ITS implemented in the web; Section 3 exposes in more details the problems imposed by the Internet structure; Section 4 proposes some possible solutions addressing the current scenario and conceives an architecture, and, finally, Section 5 presents some conclusions and future works.

2. The WITS (WWW ITS): a concrete example

This system, implemented by the Japanese universities Saga and Wakayama [1], it is a differential calculus tutor for high-school students. The system integrates the WWW and ITS technologies and due to its implementation using just mature WWW development patterns, it offers the immediate advantage of accessibility in anywhere in the Internet just using a standard browser, without, therefore, needing any special software or special devices.

The system offers personal adaptation to each particular learner through a bug library model. The students are classified in states (regarding their origin, frequency and error type) and possess an individualized instruction through dynamically generated HTML pages, whose content and presentation form depend on those states.

The system’s instructional strategy is expressed through CGI (Common Gateway Interface) scripts, implementing the domain and the learner models in the WWW server side and implementing the interface model in the student WWW browser. The interactions produced by the learner in this interface, update the learner's model that will be used to define its state.

A first obstacle for individualization in this environment is the user's identification by components running in the server. That is due to the stateless nature of the HTTP (HyperText Transfer Protocol) protocol, what means that each client requisition and respective answer is considered as a complete cycle without any relationship with future requisitions and answers. The HTTP protocol does not possess a working session concept. In this system this difficulty was solved through hidden tags in the HTML code. The tags were used to transport the user's identification data.

Figure 1, illustrates the system’s operation. The learner connects to the system through a standard http client (Netscape, IExplorer, Mosaic) and exchanges data with the server (httpd) through a HTTP requisition cycle (POST + URL + data). Among the correspondent data, the client will send identification and authentication information in the (already mentioned) hidden fields. The client will also update the server system with the results of its interactions (exercises answers, and so on). This way, the learner is identified by CGI scripts located in the server and its data is used to update the student's model, possibly changing his/her classification state. To respond to the learner's interaction, the CGI scripts uses the formalized knowledge in the domain module and also in the learner's module to choose an appropriate teaching strategy, generating individualized answers (in terms of form and content), dynamically generated in HTML format.

When the student makes a mistake, the system answers with an advice message, inserting in the result form, specific expressions generated in agreement to the domain, using an answer pattern more adapted to the user's state, to the origin of the mistake and to the chosen teaching paradigm. The system presented quite satisfactory results, but yet, improvements are recognized to be needed, especially regarding the learner's modeling approach and the advisement mechanism. Studies have been conducted looking for a more accurate learner's model. Also other improvements are being looked for: evolutions in the communication language (sometime, very similar messages can be shown to the same user), a mechanism to facilitate the information exchange with other educational systems in the Internet, and the construction of more general structures for the CGI scripts. Another important limitation is related to the poorly exploitation of multimedia resources use in this system.

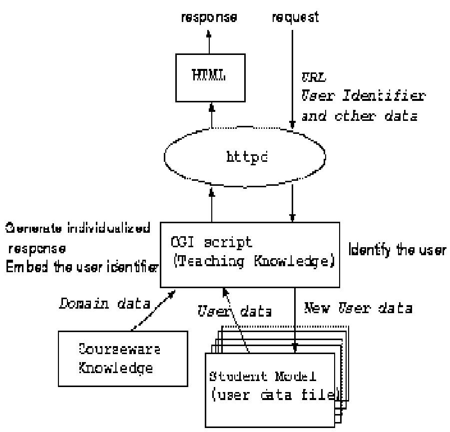

Figure 1 - WITS system operation

This is due to the limited Internet's bandwidth. A long waiting request definitely will compromise the pedagogical quality of the learning session. This issue is discussed in details in Section 3 and 4.

3. Some difficulties imposed by the Internet

The web is definitely a materialization environment, making possible to construct an ITS, in which sometimes the student learns through free discovery and sometimes is instructed by the system. Such a system is capable to exemplify, test, present contents, advise and drive the user's navigation into "necessary roads" during the learning process. The web also offers some other advantage, if considered as a popular means for educational tools. In the spite of all these advantages, it was verified some difficulties introduced by the merging of hypertext, ITS and Internet.

The first difficulty is related to the structure of the hypertext [2] because the learner can sometimes get lost in the hyperspace and, therefore, loose his time visiting knowledge nodes that either he is not prepared to understand or already has visited. This also contributes to make him loose the focus in the real objective of the learning session. All this, consequently compromises the student's learning performance.

The two other difficulties are directly related to the Internet somehow. The first one, related to the stateless nature of the HTTP protocol, which introduces a student recognition obstacle [3]. The second difficulty is related to the limited bandwidth available in the Internet that seriously compromises the transmission of continuous media.

Many of the systems that implement ITS in the Internet solve the first problem by using the so-called *cookies* (hidden fields), facilitating this way the creation of the working session concept between client and server and, consequently, adapting standard HTML for that purpose. Besides that, the HTML possesses other more serious limitations regarding specific characteristics of streaming media data. The hypermedia document model, specified in this standard, presents a very limited dynamism which eliminates the use of more sophisticated interaction mechanisms. The model imposes some difficulties to the presentation of continuous media (audio, video and animations) and consequently to the introduction of hyperlinks inside these media (e.g. links inside of a video). These limitations restrict the teaching/learning strategies that would deals with cognition through several perception means (vision, audition, etc).

4. Possible solutions for the current scenario and proposal of a distributed architecture : HyWebMap and HyWebMapVision Systems [2,5]

In this section some possible solutions will be pointed, handling separately the two groups of problems: the first group related to the nature of the hypertext and the second group related to the architecture and problems of the Internet regarding the continuous media transmission.

To guide the learner navigation through the reticular non-linear structure of the hypertext, as well as promoting adaptation to his personal characteristics, it is chosen a configuration similar the one described in [3]. Figure 2 below illustrates this scenario.

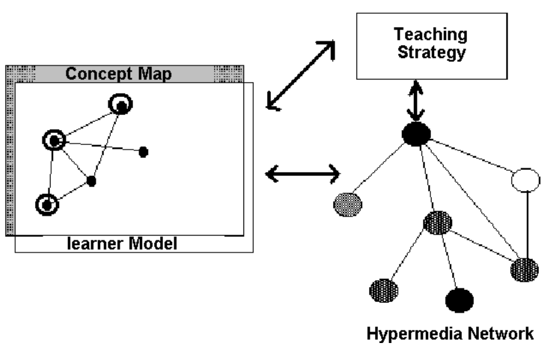

Figure 2 - Architecture of the tutorial part of the system

In the domain model, the content units are structured in a concept map, relating contents to be taught/learned with their pre-requirements. Each concept possesses several representations, either in different media, or in different presentation forms (example, exercises, explanation, etc.) structured in a hypertext map. The teaching strategy of the tutor is composed by scripts (CGI, JavaScript) or Java Applets that are responsible for the update of the learner model, the choice of the pedagogic paradigm, the dynamic formatting of the presentation, and finally for providing to the user a macro vision of the hypertext, in other words, differentiate the knowledge nodes based in the learner model and the purpose of the session.

To achieve this goal, an overlay structure is used in the student model, classifying the content units in four levels, namely "learned", "ready to be learned", "not ready to be learned " and "learning". Based on these four levels, the routines of the teaching strategy would point out the differences among the nodes, painting them with three different colors and hiding those that represented a “not ready to be learned" concept (represented in the figure by the black node). This way, the student can navigate freely (but consciously) in the hyperspace, until he decides (or is imposed by the structure of the node) to call the tutor which then assumes the control and begins the instruction session.

To teach a concept, the tutor chooses (based on the domain knowledge and on the student's abilities) the best way to do it. It can choose, for example, between using a existent presentation node in the hypertext, or dynamically assemble a new one, or choose the type of presentation (example, exercise or explanation) or yet, choose the presentation form (which media will be presented) and with which content.

we have introduced the concept of renaming links between nodes. This function allows the learner to personalise his hypertext. He can name the links between nodes according to his knowledge of the content of the nodes. This function enables the learner to have a more reflexive and planned navigation.

Several tools to help the learner navigate in his hypertext:

- Navigation by history: A classic tool, it enables the learner to go back to a node already visited.

- **Navigation by level of arborescence**: The learner can choose the level he wants to visit in the arborescence of his hypertext.

- **Navigation by date**: The HyWebMap system saves the navigation history of the hypertext for two weeks. The learner can ask to see the visited nodes during a period of time defined between two dates.

HyWebMap enables all learners to widen their networks (hypertext) by creating their personal annotations on one or several nodes. This annotation can be a text, links or a text between links. An annotation has 3 distinct fields:

- A **comment field** to make comments concerning texts, pictures or a particular link.

- A **key words field** (not really part of the node) key words can be created by the learner and can be used during the interrogations on the network.

- A **variation field** used for synonym or syntax correction. Also used for format modification.

For a given node, the usage of each of these fields is optional.

They are also completely independent. This distinction between annotations is very useful when performing a search. The annotations are in HTML format. Each time a node is revisited, the attached annotation appears in the navigator.

The architecture just described, addresses the system problems in the ITS point of view, which are: how to integrate in a complementary way the learning paradigms for free discovery and guided instruction, and also how to increment the navigation mechanism to be conscious and adaptive. Besides that, the obstacles tied to the web environment, concerning the presentation and interaction through continuous media remain unsolved.

Nowadays, these issues are handled through the use of plug-ins such as RealPlayer, CosmoPlayer(VRML), and others. This solution does not satisfy the set of requirements mentioned in this work. The main reasons are :

- plug-ins are constituted of proprietary code, what makes very difficult (and sometimes even prevents) the representation of continuous media components as part of the hypermedia model (e.g. using the HTML model can be very difficult to represent in the same page, audio, video, text and still images in a structured way).

- plug-ins do not allow hyperlinks to be inserted within the respective media.

- they need to be acquired and installed in the customer machine and (with some exceptions) are tied to a specific platform and operating system. This characteristic can eliminate the advantages of open standard, low cost and interoperability that have popularized the WWW service.

The figure 3, shows an architecture proposal that addresses these problems directly. The bottom layer is the *HyWebMapVision* system that can handle the digital video playback, synchronization, decoding and interaction (through the virtual VCR commands – play, pause, fast forward, rewind, gain control, mute). The system also stores the video material in a structured manner providing to the user a querying and selection graphical interface.

HTML
MHEG-5
HyWebMapVision

Figure 3 – layered architecture approach

In the current version, the video playback environment is set based on a particular script format that constructs a multimedia presentation by relating the individual medium. In a object-oriented fashion [6], this functionality is encapsulated in a class, which prevents the whole system structure to be affected by a change in the script format. This will be important to integrate it to the MHEG-5 engine described bellow.

The prototype now is being updated to use the Real Time Protocol as its transport protocol. Due to the protocol's QoS(Quality of Service) handling mechanisms, the playback quality over the Internet will be increased.

According to the figure 3, the student uses the system through a standard HTTP client (Netscape, IExplorer), accessing the HTML pages, in which the educational contents (coming from the content nodes) are presented. The page can be composed of static medium (e.g. text and images) - responsibility of the browser's functionalities as well as dynamic media (synchronized audio, video and animations handled by the MHEG-5 engine). The MHEG-5[7] is a presentation and exchange standard for multimedia objects that allows a hyperdocument to be composed of several media, related to each other in a spatial and temporal dimension. This standard also allows the insertion of hyperlinks inside the media.

These relationships involving the objects are formalized through scripts (a hypermedia model), which in this case is dynamically generated by the teaching strategy's routines. For performance purposes, just a minimum part of the standard should just be implemented by the engine. The MHEG-5 engine is implemented in a Java applet. We believe this integration process will not be problematic are: first, both systems are object-oriented, with specific media objects and therefore specific responsibilities and purposes. Second, they are both implemented in the same language (Java). The compound environment (MHEG/*HyWebMapVision*), thus, runs inside the HTML page.

5. Conclusions

This article has discussed some difficult problems encountered concerning the implementation of hypermedia tutoring systems in the world wide web. One of them was found to be related to the structure of the hypertext and the others directly related to the Internet. It has also been proposed an architecture capable to solve those problems. That architecture will allow the system to be open and interoperable. Through that architecture the resulting system would be composed of different media, related to each other in a formal and structured manner. In that system the learner is free without be lost and can navigate through out the hyperspace, acquiring knowledge by manipulating the different available media (including the continuous ones) in a very free fashion. When in need of help, by his own decision or by imposition of the structure of the concept requirements, the learner may call the tutor who then takes the control of the instruction and presents the content in a very individualized fashion.

6. References

[1] Okazaki, Y. et al. "An implementation of the WWW Based ITS for guiding differential calculations", Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web", 8th World Conference of the AIED Society, Kobe, Japan, 18-22 Ago 1997.

[2] Imad Saleh, Naserddine Bouhai, Fabrice Papy « HYWebMap : a système to create interactive networks » Conférence CAPS'3, Paris 12-14 Décembre 2000, 16 pages.

[3] Stern, M.K. *"The difficulties in Web-based tutoring and possible solutions"*, Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web", 8th World Conference of the AIED Society, Kobe, Japan, 18-22 Ago 1997.

[4] Brusilovsky, P. "ISIS-Tutor: An intelligent learning environment for CDS/ISIS Users", University of Trier, Germany, 1997.

[5] Imad Saleh, Naserddine Bouhai, Soufiane Bouyahi " Système HyWebMap : Outil de navigation et de restructuration des connaissances " Conférence TICE 2000. Troyes, 5 pages.

[6] Sullivan S.C. et al. "Programming with the Java Media Framework", Wiley Books, New York, 1998.

[7] ISO/IEC "MHEG-5: Support for base-level interactive applications", 13522-6, September 1996.