

Un système d'enseignement adaptatif à base du langage XML

Amar Balla*, KhaledWalid Hidouci*, Nouredine Ihadadene**,
Abdallah Hanouh**

* Institut National d'Informatique (INI)
BP 68M Oued-Smar, Alger, Algérie
{a_balla, hidouci}@ini.dz

** Institut National d'Informatique (INI)
BP 68M Oued-Smar, Alger, Algérie
{ihadaden_nouredine, ab_hanouh}@hotmail.com

Résumé

De nombreuses ressources pédagogiques sont produites ou utilisées dans le cadre d'une formation. Afin de faciliter la production et l'utilisation de ces ressources, nous proposons dans cet article un système d'enseignement adaptatif à base du langage XML. Nous présentons l'architecture générale du système, décrivons les modules du système et terminons par la présentation de quelques interfaces et scénarios réalisés.

Mots-clés : XML, hypermédia adaptatif, enseignement à distance, cours structurés.

Abstract

Many teaching resources are produced or used for the purpose of training. In order to facilitate the production and use of these resources, we propose in this paper an XML approach for the design of adaptive training system. We present an overview of the system architecture, and then describe its components including some interfaces and screen dump.

Keywords: XML, adaptive hypermedia, distance learning, structured courses.

1 INTRODUCTION

Les plates-formes d'enseignement peuvent être vues comme des systèmes d'apprentissage utilisés par trois types d'intervenants (ou acteurs) : les auteurs, les apprenants, les tuteurs.

Les auteurs produisent des cours, utilisés par les tuteurs qui guident les apprenants dans l'assimilation du contenu sémantique des cours. Souvent, quelques fonctions du tuteur peuvent être automatisées. D'un autre côté, et avec l'énorme développement d'Internet, la notion d'université virtuelle devient une possibilité croissante. Dans ce cas de figure, les trois acteurs cités plus haut peuvent être géographiquement éloignés.

La problématique actuelle lors de la conception de documents pédagogiques réside principalement dans la structuration des informations. Cette structuration oblige l'auteur à préciser et à définir les types d'informations qu'il veut faire passer et à préciser la ou les formes de présentation, qu'il faut leur associer pour atteindre l'objectif visé. Pour améliorer la qualité pédagogique de ces cours, plusieurs travaux s'intéressent à l'adaptation des informations aux profils des apprenants. Il s'agit des STIs (Systèmes Tuteurs Intelligents) sur le Web [5][11]. Pour ce type de système, avant qu'un apprenant ne démarre le cours, le système lui génère l'ensemble des parties du cours avec un séquençement bien défini selon son profil. Toutefois, cette adaptabilité est statique. En effet les interactions de l'apprenant avec le système ne sont pas prises en compte [5]. Si par exemple l'apprenant rencontre des difficultés au moment de l'apprentissage, le système n'intervient pas pour lui proposer d'autres parties du cours ou changer de séquençement.

Conscients de ce problème dans de telles situations d'apprentissage, il nous semble obligatoire de revoir l'architecture des outils de création et de présentation des activités pédagogiques multimédias proposées aux apprenants.

Dans ce cadre, on se propose de développer un système d'enseignement sur le Web permettant aux enseignants (les auteurs) de produire des cours et de les mettre à la disposition des apprenants à travers un serveur Web.

Notre modèle a été conçu comme un outil pédagogique et pragmatique d'archivage et de mutualisation de ressources pédagogiques indépendantes des plates-formes telles que WebCT, Space, Lotus Learning, etc. En effet, il nous paraissait inacceptable de faire dépendre notre modèle d'une quelconque solution commerciale. Ainsi, notre modèle offre à ses utilisateurs :

Un système d'enseignement adaptatif 3

- Un dépôt simplifié et rapide de documents ;
- Un archivage fiable, durable et sécurisé ;
- Une bonne visibilité des productions ;
- Un respect des standards du Web côté client ;
- Une production simplifiée des cours côté serveur.

Pour valider notre modèle, nous avons développé une plate-forme d'enseignement sur le Web nommée *WebSchool*, permettant aux apprenants de s'inscrire et de bénéficier des enseignements disponibles suivant leur profil. Le tuteur de *WebSchool* gère les profils de chaque apprenant et permet de suivre ainsi l'évolution de chacun afin de rendre l'enseignement adapté aux apprenants [2][3]. D'un autre côté, *WebSchool* offre aux auteurs (enseignants) les outils nécessaires pour produire les cours et les exercices, ainsi que la possibilité d'héberger ces ressources pédagogiques dans le serveur et de les modifier si nécessaire. La plate-forme permet aussi le contact direct entre les différents acteurs du système par voie de messagerie ou de conseils.

2 APPORTS DE XML

On a opté pour le langage XML (eXtensible Markup Language) pour représenter les cours car celui-ci est devenu un format standard d'échange de données [6]. En plus du fait qu'il permet de représenter la structure des documents et donc de faciliter les traitements portant sur cette structure, il permet de donner une définition sémantique aux documents que ce soit via une DTD ou par les schémas XML. De plus les documents XML sont de bons candidats pour la conservation à long terme, et ce pour les raisons suivantes :

- Le format est normalisé ;
- Les documents sont indépendants des plates-formes, des logiciels et des systèmes ;
- L'approche XML consiste à séparer le contenu de la présentation ;
- Les documents peuvent être présentés de différentes manières en utilisant la notion de feuille de style XSL ;
- Il est facile de créer des documents XML.

En ce qui concerne notre application, XML nous permet de représenter les cours avec leur structure. Il nous a permis également de définir des modèles de documents associées à des activités pédagogiques (présentation de cours, exercices, QCM, etc.), et de spécifier les

différentes unités formant le contenu sémantique d'un cours. Ces unités peuvent être réorganisées dynamiquement lors de la présentation pour s'adapter aux profils des apprenants. La transformation du document XML initial avec ses facilités de navigation dans le document final est réalisée avec une transformation XSLT.

Pour l'auteur, créer une activité pédagogique peut se résumer à éditer sa structure et son contenu à l'aide d'un éditeur XML, puis à le transformer en fichiers HTML à l'aide d'un moteur qui exécute des règles de transformation écrites en XSLT.

3 ARCHITECTURE DU SYSTEME

Comme le montre la figure 1, l'architecture choisie pour notre système est une architecture client/serveur [13][14][16] sur laquelle repose la majorité des plates-formes d'enseignement actuelles car elle permet la séparation entre le poste client, le serveur d'applications et le stockage de l'information.

Le rôle de l'application client est de permettre aux utilisateurs (Administrateur, Enseignant et Apprenant) d'interagir avec la plate-forme. En plus des fonctionnalités de base décrites dans la figure.1, le système que nous avons développé offre à ses clients un ensemble d'outils (forum, chat) permettant un travail coopératif dont l'intérêt n'est plus à démontrer [8][17].

L'administrateur est une personne chargée de gérer et de maintenir les serveurs et l'accès à ces derniers par les enseignants et les apprenants. Elle n'a donc pas de rôle pédagogique particulier.

Le tuteur "logiciel" dans notre cas permet d'exploiter les cours faits par les enseignants auteurs et les profils des apprenants, en vue de réaliser le suivi et l'adaptation de l'enseignement ainsi que l'évaluation des apprenants pour la mise à jour de leurs profils.

Un système d'enseignement adaptatif 5

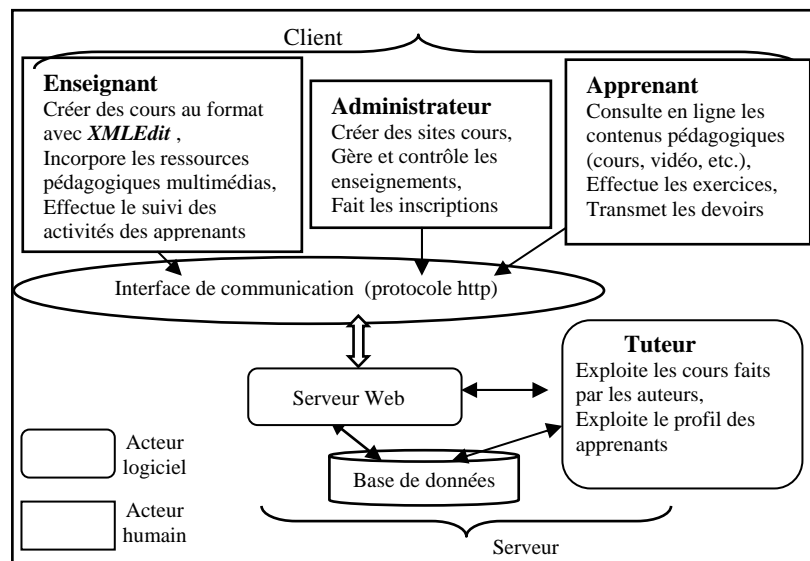


FIG. 1 – Architecture du système

4 DESCRIPTION DU SYSTEME

Nous présentons maintenant la description des composantes du système qui ont été conçues pour faciliter les activités de l'enseignant et de l'apprenant au sein d'un environnement éducatif d'apprentissage.

La Base de données

Notre base de données permet d'atteindre les objectifs suivants :

- Contrôle de l'accès au système d'apprentissage;
- Modélisation des connaissances relatives aux cours dispensés;
- Suivi et évaluation des apprenants;
- Adaptation des cours en fonction des profils des apprenants.

Pour le premier objectif, nous disposons de tables concernant les *auteurs*, les *cours*, les questions à choix multiples (*QCM*) ainsi que les *apprenants* inscrits dans le système. Les liens entre ces différentes entités sont dynamiquement mis en place ("tel" auteur a déposé "tel" cours qui

fait référence à "telle" liste de QCM et suivi par "tel" apprenant, etc. : voir les tables : *Aut_cours*, *Inscription* et *QCM*).

Pour le second objectif, on utilise le concept de " graphe d'unités de connaissances " [13] qui permet de modéliser le contenu sémantique d'un cours par un graphe où chaque sommet représente une notion du cours (ou "unité de connaissance"). Ces notions sont reliées entre elles par des arcs formalisant deux types de liens : "hiérarchiques" et "de précédence". Les liens hiérarchiques définissent un sous-graphe de type arborescent où les sommets supérieurs représentent les notions thématiques d'un cours, alors que les sommets inférieurs de l'arborescence modélisent les connaissances atomiques. Les liens de précédence établissent un ordre de parcours des notions de même niveau par un apprenant donné. De plus, chaque unité de connaissance (ou sommet du graphe) possède une liste de notions pré-requises formalisée par une expression disjonctive de sommets.

Par exemple: $(N1 \wedge N2) \vee (N3 \wedge N4)$ indique que la notion courante ne doit être abordée par un apprenant que s'il a déjà maîtrisé les notions N1 et N2 ou alors les notion N3 et N4. La figure.2 montre un exemple d'une partie d'un graphe de connaissances relatives à un cours sur un langage algorithmique :

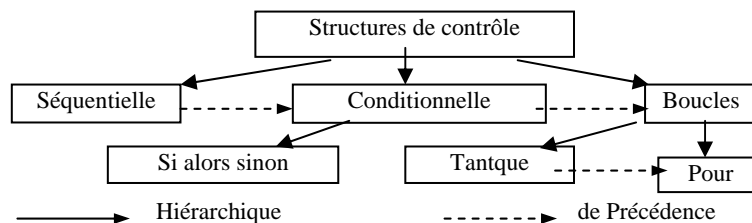


FIG. 2 – Exemple de graphe de connaissances

Quand un auteur dépose un cours dans le serveur, il met en place le graphe de connaissances associé en indiquant pour chaque sommet, la partie du cours qui lui est liée. Le graphe des connaissances est matérialisé par une table "*UC (Unité Connaissance)*" où chaque "tuple" représente une unité de connaissance et une table "*Liens*" représentant les deux types de liens entre unités de connaissances (lien hiérarchique et lien de précédence).

Pour pouvoir suivre et évaluer un apprenant, ainsi que pour adapter les cours en fonction du profil des apprenants (voir les deux derniers objectifs mentionnés plus haut), nous avons représenté l'état des

Un système d'enseignement adaptatif 7

connaissances d'un apprenant (ou modèle de l'apprenant) en utilisant la méthode dite de "l'overlay"[4][7]. C'est-à-dire que le modèle de l'apprenant peut être vu comme un sous-graphe du modèle du domaine (graphe des connaissances). Pour cela, nous avons associé, dans une table ***ConnaissancesApp***, le degré de maîtrise des unités de connaissances pour chaque apprenant aux nœuds du graphe de connaissances. Cette valeur (le degré de maîtrise) est calculée à partir des évaluations (par QCM) de l'apprenant. A chaque unité de connaissance (ou notion) peut être associée une liste de questions à choix multiples (QCM).

L'adaptation d'un cours pour un apprenant donné est possible car certaines parties du cours sont dites mobiles, c'est-à-dire que leur présence lors de la visualisation (DisplayCourse) dépend des connaissances acquises par l'apprenant concernant les pré-requis associés à l'unité de connaissance en cours. La mobilité d'une partie du cours est implantée par l'attribut " Mobile " de la balise <Brique> dans le fichier XML contenant le cours.

Le lien entre une unité de connaissance et une partie du cours est réalisé par le biais des attributs d'une table des unités de connaissances UC : *code_cours* et *balise* (qui permet de délimiter dans le fichier XML du cours, la partie en question).

Les QCM sont liés aux cours à travers les unités de connaissances (*code_uc* dans une table de ***QCM***).

Le module StoreCours

Ce module situé côté serveur permet :

- a) d'une part de vérifier la structure du cours en fonction de la DTD choisie ;
- b) et d'autre part de mettre à jour de la base de données afin de rendre le contenu accessible aux apprenants.

Ce dernier point consiste à générer automatiquement le graphe de connaissances à partir de certaines balises spécifiques que l'auteur aura préalablement introduites. Il est à noter que dans le fichier XML représentant le cours, il y a la structure du cours, les QCM ainsi que les informations nécessaires à la construction du modèle du domaine. En effet, les attributs de certaines balises utilisées telles que " sections, briques, exercices, etc." permettent de spécifier des unités de connaissances. A cet effet, l'auteur dispose d'un outil nommé ***XMLEDIT*** situé côté client, qui lui permet de générer interactivement le fichier XML.

Le module ListCours

Comme les apprenants diffèrent les uns des autres par leurs profils, le système doit attribuer une liste de cours appropriée à chaque profil d'apprenant. Par conséquent, nous avons conçu le module *ListCours* adaptable et dynamique [2][3] qui a pour tâche de récupérer la liste des cours adéquate à un type d'apprenant donné. Ce module retourne la liste des cours selon le profil de l'apprenant, en suivant les étapes de la figure.3.

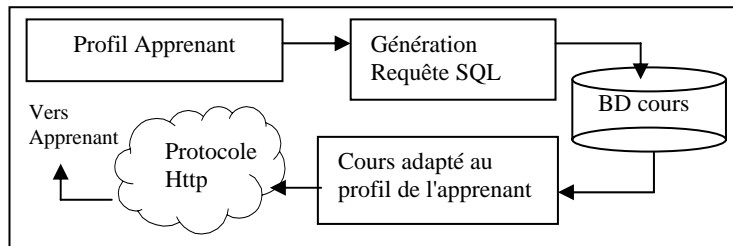


FIG. 3 – Processus d'adaptation d'un cours

Le module DisplayCours (Côté Auteur)

Un cours donné sera, lors de sa création, balisé uniquement en fonction de son contenu intrinsèque indépendamment de sa restitution future (présentation). Le rôle principal du module *DisplayCours* est d'assurer la restitution des cours qui sont aux formats XML grâce à une feuille de style XSL. La figure.4 montre les différentes phases d'affichage d'un cours donné (c'est une Prévisualisation).

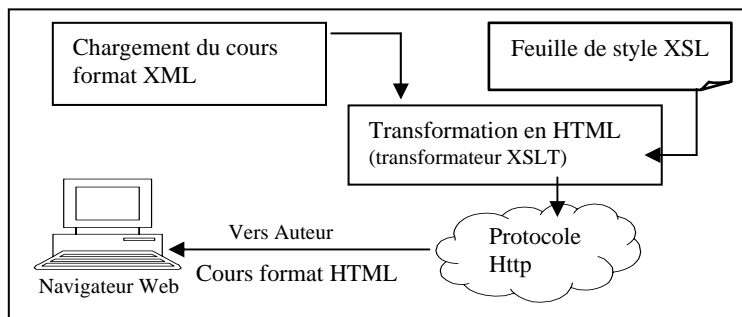


FIG. 4 – Processus de prévisualisation d'un cours

Le module Tuteur

Le tuteur est un module logiciel chargé d'exploiter le contenu des cours et exercices déposés par les enseignants au niveau de la base de données, de les adapter en fonction du profil de l'apprenant, de présenter des questionnaires et exercices à l'apprenant et de faire l'évaluation de ses connaissances.

Le module DisplayCours (Côté Apprenant)

La figure.5 montre les différentes phases d'affichage d'un cours.

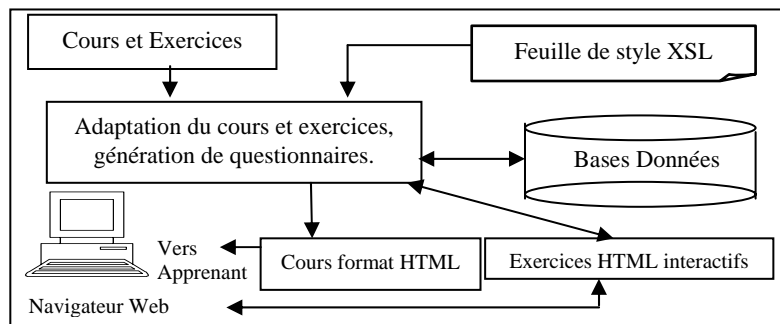


FIG. 5 – Processus d'affichage d'un cours "côté Apprenant".

Le Processus de production

Il a pour objectif d'assister l'auteur dans l'édition et la production des cours au format XML.

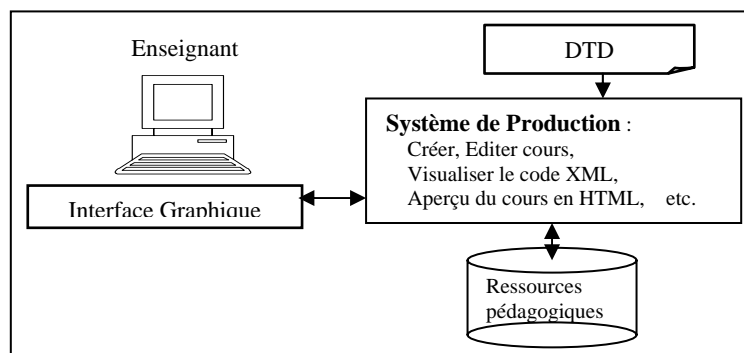


FIG. 6 – Processus de production et d'édition de cours.

Structure d'un cours

Le cours généré est un document XML qui est décrit par une DTD "Document Type Definition". Cette DTD définit les noms des balises et la façon dont elles sont organisées. Le système de production de cours permet en effet de produire des cours qui respectent les règles définies dans la DTD; il génère des documents valides. La syntaxe de la DTD est très claire. Un cours est composé de :

- a) Front : dans cette partie, on stocke toutes les informations concernant le cours comme :
 - Titre : le titre du cours ;
 - Auteur : l'auteur (enseignant) de ce cours ;
 - MotsCles : la liste des mots clés ;
 - Date : la date de création ou de la dernière modification du cours ;
 - Resume : le résumé du cours.
- b) Introduction : (facultative).
- c) Sections : sections et sous sections qui contiennent le cours.
- d) Conclusion : (facultative).
- e) Références : références bibliographiques (sites Internet, Livres, etc.).

5 MISE EN ŒUVRE

5.1 Introduction

Pour mettre en œuvre les concepts présentés précédemment, nous avons implémenté une application d'aide à la production de cours adaptatifs en fonction du profil de l'apprenant. Nous avons développé :

1. Une Interface permettant à un enseignant de générer des cours au format XML stockés dans un serveur de cours.
2. Une application (au niveau du serveur de cours) jouant le rôle de "tuteur" du système d'EIAO, permettant d'adapter un cours à un apprenant en fonction de son profil.
3. Une interface (au niveau client) permettant à un apprenant d'accéder aux différents cours disponibles.

Pour la réalisation nous avons opté pour la démarche suivante :

Côté Client : Sur ce côté, un enseignant dispose de deux interfaces, la première est un assistant de production des cours, au format XML que nous avons nommé *XMLEdit*. Le rôle de *XMLEdit* se limite à l'édition

des cours au format XML. Quant à la deuxième interface, elle permet de se connecter au serveur via une requête http afin de déposer les cours édités avec *XMLEdit* et d'autres contenus pédagogiques. Concernant les apprenants, ils disposent d'une seule interface qui leur permet de se connecter au serveur afin d'accéder aux différents cours et contenus pédagogiques mis à leur disposition par les enseignants. L'administrateur dispose aussi d'une interface qui lui permet de se connecter au serveur afin d'administrer la plate-forme.

Côté Serveur : Sur ce côté, nous avons stocké les différentes tables qui constituent la base de données de notre système, et un ensemble de modules qui permettent d'interagir avec les clients afin de répondre à leurs besoins. Le tuteur permet d'exploiter le contenu des cours et exercices déposés par les enseignants, de les adapter en fonction du profil de l'apprenant, de présenter des questionnaires et exercices à l'apprenant et de faire l'évaluation de ses connaissances.

5.2. Outils utilisés

A part *XMLEdit*, tous les autres modules de *WebSchool* ont été développés en PHP et HTML et représentent donc un site WEB géré par un serveur APACHE et un serveur de bases de données MySQL. *XMLEdit* est une application indépendante, fonctionnant au niveau des postes clients (hors connexion) afin de permettre aux auteurs de produire, sous format XML, des cours adaptables. Deux versions de l'application ont été développées respectivement pour Linux et Windows et ce grâce à l'utilisation de la bibliothèque graphique multi plateforme QT.

5.3 Présentation des applications

Nous présentons dans ce qui suit quelques interfaces et scénarios d'utilisation du système réalisé en respectant la structure du modèle.

5.3.1. Présentation de l'interface XMLEdit

XMLEdit (comme le montre la figure.7) est une interface (côté client) qui permet d'assister un auteur (enseignant) afin d'éditer les cours au format XML. L'auteur doit seulement, introduire les informations que doit contenir son cours, sans se préoccuper du balisage car ce dernier est généré automatiquement par l'application. L'application se compose d'une barre de menus, d'une barre d'outils et d'une barre à onglets.

Un système d'enseignement adaptatif 12

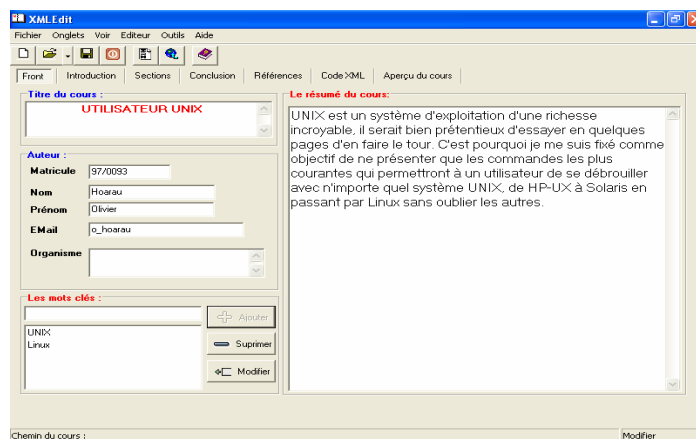


FIG. 7 – Interface de XMLEdit sous Windows

Description de la barre à onglets

Elle comporte un certain nombre d'onglets implémentés sous Windows et sous Linux qui permettent:

Onglet Sections : c'est l'onglet principal car il offre un ensemble d'outils pour rédiger les différentes sections et sous-sections d'un cours et d'insérer des paragraphes, listes à puces, listes de numérotations, des images et les différents fichiers multimédias.

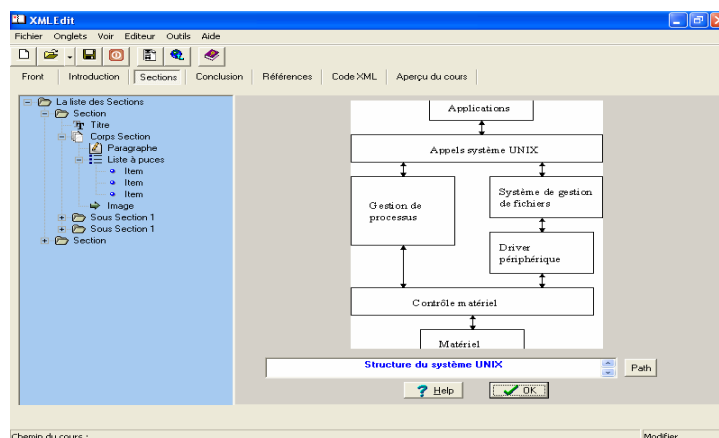


FIG. 8 – Onglet Sections

Dans la figure 8 est présenté un exemple de construction d'une section de type « image » dont le contenu apparaît dans la partie droite. La hiérarchie du cours (le plan) est matérialisée par l'arborescence de la partie gauche de la figure. C'est à ce niveau que l'auteur insère, supprime ou modifie les sections de son cours.

Onglet Code XML : permet de visualiser le code XML généré.

Onglet Front : offre un espace qui permet à l'enseignant d'introduire les informations sur le front d'un cours, c'est-à-dire le titre du cours, la liste des mots clés, les informations relatives à l'auteur (nom, prénom, matricule, email et l'organisme) et un résumé du cours.

Onglet Introduction et conclusion : ils offrent des espaces qui permettent de rédiger une éventuelle introduction ou conclusion (facultatives).

Onglet Références : permet d'introduire les éventuelles références bibliographiques du cours, titre d'un livre, URL et/ou autres.

Onglet aperçu : permet de visualiser le cours généré en appliquant au code XML une transformation vers le code HTML à l'aide d'une feuille de style XSL. La figure 9 présente un exemple d'aperçu.



FIG. 9 – Onglet aperçu

5.3.2. Présentation de la plate-forme WebSchool

La plate-forme *WebSchool* est une application Web accessible à travers un navigateur web indépendamment du système d'exploitation du poste client. La figure 10 montre le point d'entrée de la plate-forme *WebSchool*.

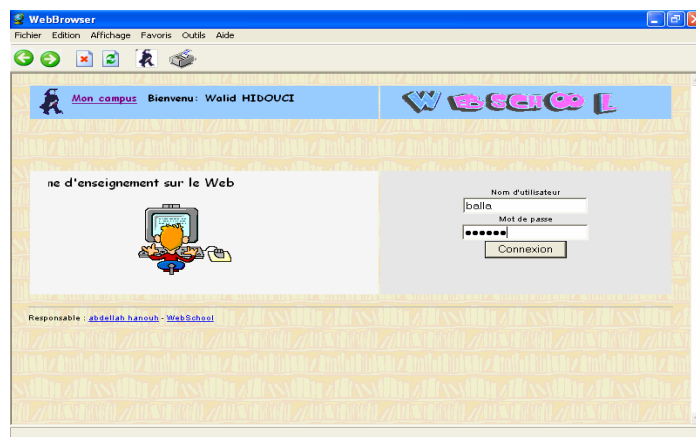


FIG. 10 – Page d'accueil de WebSchool

Durant une connexion, un utilisateur doit s'identifier par un nom de login et un mot de passe. Un certain nombre de variables de connexion permettent de gérer les utilisateurs durant leur la connexion :

- Uid : Contient le mot de passe crypté ;
- Login : Contient le nom d'utilisateur ;
- Nom : Contient le nom de famille de l'utilisateur ;
- Prenom: Contient le prénom de l'utilisateur ;
- Statut : Administrateur, Enseignant, ou Apprenant ;
- Email : l'adresse électronique de l'utilisateur ;
- Etude : Contient l'année scolaire et l'option de l'apprenant ;

5.3.3. Description des interfaces utilisateurs

5.3.3.1. Utilisateur Apprenant

Lorsqu'un apprenant est connecté, une interface lui donne accès à son environnement de travail, qui contient différents outils et informations :

- a) La liste des différents modules de son profil (créés par l'administrateur)
- b) Un menu qui contient les différentes options :
 - *Modifier mon profil* : permet à l'apprenant de modifier son profil (nom d'utilisateur, adresse électronique et mot de passe).
 - *Suggestion* : permet d'envoyer une suggestion à l'administrateur de la plate-forme.

- *Messagerie interne* : permet d'échanger des messages entre les utilisateurs de la plate-forme.
- *Entrer dans le chat* : permet de communiquer en direct avec d'autres utilisateurs connectés.
- *Aide* : l'accès au système d'aide de la plate-forme.
- *Quitter* : permet de fermer la connexion en cours

Pour chaque module choisi par l'apprenant, le système lui donne accès à son "cartable électronique" (comme le montre la figure 11) qui contient :

- Les différents cours disponibles concernant le module choisi ;
- Les différents exercices disponibles pour ces cours ;
- Un espace de stockage des travaux et fichiers personnels de l'apprenant ;
- Un espace d'annonces faites par l'administrateur ou l'enseignant responsable du module (ex : planning des examens, etc.) ;
- Un ensemble de liens Internet utiles déposés par l'enseignant responsable ;
- Un espace pour les forums de discussion, permet de déposer des questions, ou de répondre à des questions.
- Un agenda pour mémoriser les dates importantes,
- Un espace qui permet d'accéder aux fichiers multimédias mis à disposition par l'enseignant responsable ou l'administrateur.

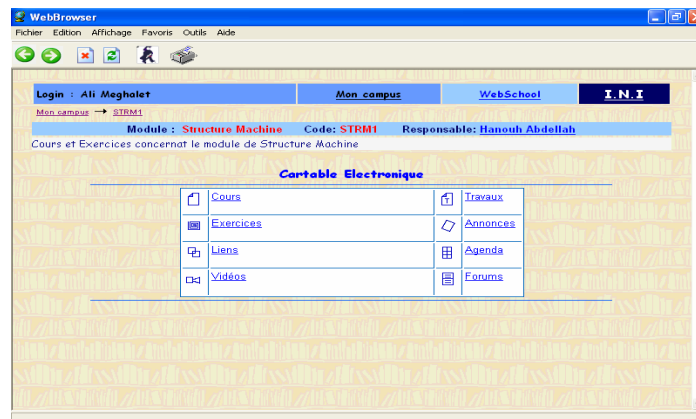


FIG. 11 – Cartable électronique de l'apprenant

5.3.3.2. Utilisateur Enseignant

Un enseignant qui se connecte à la plate-forme *WebSchool*, dispose également d'une interface graphique qui va lui permettre d'accéder à son

"casier électronique". Celui-ci est une base de données qui renferme les cours que l'enseignant a construits à l'aide de l'interface *XMLEdit*, ainsi que les exercices qu'il crée, modifie ou supprime. Ces cours et exercices seront utilisés à terme pour alimenter les cartables des apprenants. Comme pour les utilisateurs de type apprenant, les enseignants disposent aussi d'un menu qui leur permet d'accéder aux différentes options de la plate-forme à savoir la modification de leur profil, la messagerie interne, le système d'aide et le chat. Pour alimenter les cartables des apprenants les enseignants disposent d'une interface qui leur permet d'héberger les cours qu'ils ont créés à l'aide de *XMLEdit*, et une interface pour créer des exercices de type QCM. La figure 12 montre la page associée à l'hébergement d'un cours par un enseignant.

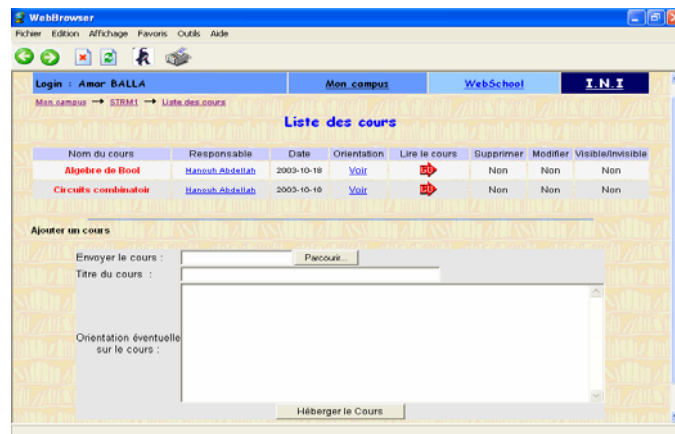


FIG. 12 – Hébergement d'un cours

5.3.3.3 Utilisateur Administrateur

L'administrateur de la plate-forme *WebSchool* a la responsabilité de la gestion des comptes des utilisateurs apprenants et enseignants. Quel que soit le type d'utilisateur, un compte se caractérise par un nom d'utilisateur et un mot de passe pour l'identification et l'autorisation d'accès au système. Pour ajouter un compte utilisateur de type apprenant, le niveau de celui-ci doit en plus être renseigné puis le cartable de l'apprenant est créé. Un casier électronique est créé lorsqu'un utilisateur de type enseignant est ajouté. L'administrateur a la charge de créer les différents modules pédagogiques à la demande des enseignants. Pour chaque module créé l'administrateur doit donner le statut de ce module, à savoir le niveau.

CONCLUSION ET PERSPECTIVES

Notre démarche est très comparable aux projets: "*NURAXI Web-based learning environment architecture*" [15] et SMART-Learning [3] et des travaux de H.Habieb-Mammar [9][10]. Il s'agit également d'utiliser les technologies de l'information et de la communication pour le traitement des activités pédagogiques. De plus, nous soulignons que de telles recherches doivent se préoccuper de la conception des documents avant leur présentation aux apprenants. Ainsi nous proposons une granularité assez fine du document afin qu'il puisse constituer une source effective de connaissances pour les diverses activités pédagogiques proposées aux apprenants. Par ailleurs, pour que l'adaptabilité soit gérée facilement, nous avons incorporé dans le document toutes les descriptions susceptibles de produire dynamiquement différentes présentations d'une même information renforçant ainsi son intérêt d'apprentissage.

Ainsi, le but de ce travail est de concevoir un système d'enseignement sur le Web en utilisant la norme XML comme moyen de structuration des contenus pédagogiques destinés aux apprenants. Le prototype réalisé nous a permis de valider l'approche de conception retenue et l'architecture définie dans la figure.1, qui nous paraît répondre à nos exigences. Il est à noter que notre prototype est complètement opérationnel et peut être utilisé aussi bien sur des plateformes Linux que Windows, sur du matériel à faible coût.

L'approche de conception itérative et incrémentale fait que notre travail ne s'arrête pas là et que des perspectives d'évolution, à court et à moyen termes s'offrent à nous. Nous travaillons actuellement sur l'expérimentation en vraie situation de formation en cherchant à évaluer les impacts de l'utilisation de notre plateforme.

A moyen terme, nous envisageons d'enrichir et d'étendre la plateforme par d'autres fonctionnalités telles que la simplification des outils auteurs afin de faciliter leur utilisation par des enseignants non informaticiens. Nous pensons également ajouter certaines fonctionnalités aux interfaces des apprenants en proposant à ceux-ci une vue sur leurs propres activités et également sur des activités collectives. Enfin, l'amélioration de l'enseignement peut se faire en utilisant une approche multi agents [16]. A cet effet, nous travaillons actuellement sur la décomposition de la fonction du tuteur en agents autonomes et coopérants [1], en proposant des agents accompagnateurs, co-apprenants virtuels et perturbateurs.

RÉFÉRENCES

- [1] E. Aimeur, C. Frasson, H. Duffort, Co-operative learning strategies for Intelligent Tutoring Systems (ITS), *Applied Artificial Intelligence, an International Journal*, VOL. 14(5), pages 465-490, 2000.
- [2] A. Balla, M.T. Laskri, S. Laoudi, HYPERGAP: un hypermédia éducatif dynamique générant des activités pédagogiques, *Document Numérique*, Volume7-n°1-2, pages 39-57, Hermes, 2003.
- [3] Y. Belqasmi, Z. Bentaleb, A. Benkiran, R. Ajhoun, An author system for the production of adaptive system for the production of adaptive courses, *ITHET 2001* July 4-6, pages 299-330, Kumamoto, Japan, 2001.
- [4] P. Brusilovsky, Methods and techniques of Adaptive Hypermedia, *Adaptive hypertext and hypermedia*, pages 1-43, Kluwer Academic Publishers, 1998.
- [5] P. Brusilovsky, Adaptive Educational Systems on the World Wide Web: A review of available technologies, *Proceedings of the Workshop "WWW-Based Tutoring" at the 4th International Conference on Intelligent Tutoring Systems, ITS'98*, San Antonio (USA), 1998.
- [6] J.P. David, Modélisation d'activités pédagogiques avec le langage XML, *EIAO'01*, Paris (France), 2001, *Sciences et Techniques Educatives S.T.E.8.(1-2)/2001*, pages 176-177, Hermes, 2001.
- [7] N. Delestre, METADYNE, un Hypermedia Adaptatif Dynamique pour l'Enseignement, *thèse de l'université de Rouen*, France, 2000.
- [8] A. Dufresne, Conception d'une interface adaptée aux activités de l'éducation à distance ExploraGraph, *Sciences et Techniques Educatives S.T.E.8 (3)/2001*, pages 301-320, Hermes, 2001.
- [9] H. Habieb-Mammar, F. Tarpin-Bernard, P. Prevot, Adaptive Presentation of Multimedia Interface, case study: "Brain Story" Course, *Lecture Notes in Computer Science*, pages 15-24, Springer-Verlag Heidelberg, J.G. Carbonell and J.Siekman (eds), Vol 2702, Janvier 2003.
- [10] H. Habieb-Mammar, F. Tarpin-Bernard, P. Prevot, Modélisation XML des interfaces adaptatives intégrant le profil cognitif de l'utilisateur, *DVP'2002 Documents Virtuels Personnalisables*, Brest, Juillet 2002.
- [11] Kinshuk, A. Patel, Intelligent Tutoring Tools: Redesigning ITSs for Adequate Knowledge Transfer Emphasis, *Proceeding of 1996 International Conference on Intelligent and Cognitive Systems*, pages 221-226, Tehran, 1996.
- [12] A.MBALA, C. Reffoy, T. Chanier, SIGFAD: un système multi-agents pour soutenir les utilisateurs en formation à distance, *EIAH2003*, pages 319-330, Strasbourg (France), 2003.

- [13] R. Nkambou, G. Gauthier, C. Frasson, Un modèle de représentation des connaissances relatives au contenu dans un système tutoriel intelligent, *Sciences et Techniques Educatives S.T.E.4(3)/1997*, pages 299-330, Hermes, 1997.
- [14] H. Raad, B. Causse, Modelling of an Adaptive Hypermedia System Based on Active Rules, *ITS2002*, pages 149-157, Biarritz France, SanSebastian Spain, 2002.
- [15] L. Sommaruga, N. Catenazzi, S. Giroux, C. Moulin, The NURAXI Web-based learning environment architecture, *Proceedings of ED-MEDIA 2000*, pages 1478-1479, Montréal Canada, 2000.
- [16] H. Wu, G-J. Huben, P. De-Bra, AHAM: A reference Model to Support Adaptive Hypermedia Authoring, *Proceeding of the Zesde Interdisciplinaire Conferencie Informatiewetenschap*, pages 77-88, Antwerp, 1998.
- [17] A. Zidani, M. Djoudi, S. Zidat, S. Talhi, CHELIA: Un environnement coopératif pour l'apprentissage sur Internet, *CARI'02*, pages 339-346, Cameroun, 2002.