

Des services Web pour le e-Learning

Ivan Madjarov

Laboratoire des Sciences de l'Information et des Systèmes

(LSIS) - UMR CNRS 6168, Université d'Aix-Marseille

Domaine Universitaire de Saint-Jérôme

Avenue Escadrille Normandie-Niemen

13397 Marseille Cedex 20

madjarov@iut-gtr.univ-mrs.fr

Résumé

La fonction principale des plates-formes e-Learning est de fournir à l'apprenant les bonnes activités avec les bons outils au bon moment en fonction de ses besoins. Cela nécessite l'application de mécanismes d'animation et de coordination des modules et des activités pédagogiques. Si un système e-Learning est une collection d'activités ou de processus, nous pouvons découper ses fonctionnalités en un certain nombre de fonctions autonomes qui peuvent alors être réalisées séparément sous la forme d'applications autonomes ou de e-services, en utilisant la technologie des services Web. Cet article présente notre approche de conception et développement d'outils e-Learning, approche expérimentée dans le cadre du projet XESOP.

Abstract

E-Learning platforms aim to provide learners with the most appropriate activities and tools that best fit their needs. This requires animation methods and coordination for modules and pedagogical activities. An e-Learning system can be seen a collection of activities or process, whose functionalities are autonomous functions that may be designed separately as autonomous applications or e-Services, in using Web. This article presents our approach for designing an e-Learning platform that complies with several standards. The work described in this paper has been conducted in the framework of the XESOP project.

Mots-clés

Services Web, Système distribué, e-Service, e-Learning.

Keywords

Format, style, component (style *Mots Clés*)

1. Introduction

A l'heure actuelle les concepts et les technologies e-Learning (e-Formation) ne sont pas figés. On ne trouvera donc nulle part de définition unifiée caractérisant ni le concept, ni l'architecture standard. Les auteurs tels que (Abel et al., 2003) rappellent que le e-Learning peut désigner des notions aussi variées que la gestion administrative d'une formation sur Internet, la diffusion d'un cours basée sur le Web ou la mise à disposition d'outils de conférences virtuelles. Dans la pratique, le terme e-Learning désigne souvent de nouveaux services techniques : on l'associe aux cours numériques, à l'enseignement par Internet, ou encore à l'apprentissage en ligne. Nous le voyons plus globalement comme une évolution des systèmes d'information et des dispositifs d'enseignement.

Les efforts de standardisation dans ce domaine se concentrent sur la réutilisation des documents pédagogiques, mais ne concernent pas la réutilisation des fonctionnalités des applications. Les plates-formes, aussi bien commerciales que libres, sont en général centralisées offrant des cours avec un contenu bien défini.

Notre approche se base sur la conviction qu'un système e-Learning est une collection d'activités ou de processus qui agissent d'une part sur les étudiants et d'autre part sur le contenu pédagogique convenablement choisi sous forme d'objets pédagogiques (Learning Objects). Ainsi, nous pouvons découper les fonctionnalités de base d'un système e-Learning en un certain nombre de fonctions, qui peuvent alors être implantées séparément sous la forme d'applications autonomes (stand-alone) ou sous la forme de e-Services en utilisant des services Web. La mise à disposition de ces services permet la réutilisation du contenu et des fonctionnalités dans une plate-forme e-Learning.

Cet article présente nos travaux qui sont basés d'une part sur des normes connues et partagées dans le domaine du e-Learning, et d'autre part sur des concepts technologiques clés comme les services Web pour la réalisation de notre plate-forme XESOP. Nos efforts sont orientés vers la réalisation d'un mode distribué de conception et de diffusion de cours d'enseignement, afin de proposer des formations mieux adaptées aux besoins des étudiants. Cela soulève un certain nombre de défis comprenant la maintenance des contenus pédagogiques et des services offerts.

Cet article est organisé comme suit : en section 2, nous passons en revue les composants souhaitables dans une plate-forme e-Learning (plate-forme en abrégé). La section 3 est consacrée aux mécanismes de diffusion d'objets pédagogiques sur le Web. En section 4, nous décrivons notre vision des services et tâches que doit fournir une plate-forme. La section 5 donne des détails techniques d'implantation. Enfin, nous concluons en section 6.

2. Les composants d'une plate-forme e-Learning

Il existe aujourd'hui de nombreux projets de recherche et de développement dans le domaine de e-Learning. Les plates-formes existantes se ressemblent par leurs fonctionnalités et leurs multiples extensions. Cela est dû surtout au fait que les organismes de standardisation LOM (IEEE-LOM, 2003), IMS (IMS-CP, 2004), SCORM (SCORM, 2004) concentrent leurs efforts sur la structuration et la réutilisation des documents pédagogiques en définissant un élément de base appelé objet pédagogique (IEEE-LOM, 2003) ou Learning Object (LO). Par contre, les problèmes du contenu de ces objets et la réutilisation des fonctionnalités des applications qui les animent ne sont pas traités.

L'objectif principal de e-Learning est d'améliorer la qualité de l'apprentissage et non de se substituer aux modes traditionnels d'enseignement. Les moyens pour atteindre cet objectif sont multiples, complémentaires et indépendants tout en gardant l'autonomie de ces différents

moyens : accès à des ressources variées, offres de services de tutorat à distance, outils de communication, résolution d'exercices, échanges, et collaboration à distance.

2.1 Différents type de projets

On peut classer les projets e-Learning en quatre catégories (Bouthry et al., 2003) :

- des outils d'aide aux formateurs qui utilisent le réseau pour préparer et suivre un cours présentiel. Cela concerne les multiples cours et exercices mis en ligne dans le format statique du HTML ;
- des moyens pour mettre à disposition des savoirs par des portails donnant accès à des documents pédagogiques. Ils jouent le rôle de complément de formation en libre service. Ce sont, dans la plupart des cas, des systèmes de gestion de contenu pédagogique capable de générer des cours et des exercices à la demande ;
- des guides à distance avec définition de parcours pédagogiques. Ce sont des systèmes et plates-formes plus complexes qui concentrent des modules de création de contenu, de suivi et d'évaluation des résultats et définissent pour certaines réalisations le parcours de l'apprenant en fonction de ses propres résultats ;
- des formations intégrées aux dispositifs de gestion des compétences et des connaissances où le e-Learning vient en support à la gestion, à la mise à disposition du savoir de l'entreprise et vient en support également au développement des compétences. Ce type de systèmes caractérise une formation dite d'entreprise.

2.2 Les besoins des différents types d'utilisateurs

Une plate-forme est un système d'information qui doit satisfaire certains besoins et être en accord avec certains critères (ORAVEP, 2000) tels que : reposer sur les technologies de l'Internet, satisfaire aux normes en vigueur (LOM, AICC, IMS, SCORM et autres), permettre de gérer plusieurs types d'activités pédagogiques (cours, exercices, communication), ne pas exiger de très hauts débits de communication, ne pas exiger l'installation d'un logiciel particulier sur le poste client.

Ce type de système regroupe des outils nécessaires aux principaux utilisateurs, à savoir l'enseignant, l'étudiant et l'administrateur. Ces dispositifs assurent en finalité la consultation à distance de contenus pédagogiques, l'individualisation et le suivi de l'apprentissage, et les interactions entre les trois type d'utilisateurs.

Les besoins de l'enseignant consistent en : la création d'un contenu pédagogique multimédia, la création des parcours pédagogiques types ou individualisés et, finalement des moyens de suivi des activités des étudiants. Les enseignants intervenant dans une matière ont besoin d'échanger des objets pédagogiques pour la création en collaboration d'un contenu mieux adapté aux besoins d'un certain public ou d'une formation particulière.

L'étudiant de son côté a besoin de consulter en ligne ou de télécharger les contenus pédagogiques qui lui sont recommandés, d'effectuer les exercices qui lui sont soumis, de répondre à des questionnaires proposés et d'obtenir une évaluation de son parcours individualisé.

Les enseignants et les étudiants ont des besoins communs de communication mutuelle ou en groupe par des thèmes de discussion et de collaboration à des documents communs.

L'administrateur d'un système e-Learning installe et assure la maintenance de la plate-forme, gère les accès et les droits des uns et des autres, crée des liens avec les systèmes d'information externes (scolarité, catalogues, ressources pédagogiques, etc.).

Tous ces critères sont regroupés dans des systèmes d'information de type gestion de contenus pédagogiques ou Learning Content Management System (LCMS). Une solution de LCMS est un environnement permettant aux créateurs de cours de créer, stocker, réutiliser, gérer et

distribuer des contenus pédagogiques à partir d'un référentiel unique (PCBI, 2003). Ce référentiel stocke des objets pédagogiques, et la plate-forme LCMS permet de les associer, les ordonner afin de construire un cours cohérent. Ces solutions sont basées sur les postulats des systèmes de gestion de contenus (Content Management System - CMS) : stockage des contenus dans un référentiel unique ; séparation du contenu de sa présentation ; interface d'administration unique ; gestion des profils et des droits des utilisateurs. Ces outils sont utilisés pour la gestion de la création et de la publication de pages ou de documents sur des sites Web. Les solutions de gestion de contenu fournissent un ensemble d'éléments de base pouvant être assemblées entre eux et enrichis par des développements spécifiques. La centralisation des données dans des bases de données permet la réutilisation d'éléments existants dans d'autres contenus ; elle est le fondement du travail collaboratif sur le même contenu (Walckiers et al., 2004).

2.3 Les activités dans une plate-forme e-Learning

Des études sur les différentes plates-formes telles que (PCBI, 2003), (ORAVEP, 2000) et (Catalyst, 2003) démontrent que la majeure partie d'entre elles peuvent fonctionner sans exiger sur le poste de l'étudiant un logiciel autre que des navigateurs universels (tels que Netscape et Internet Explorer). Cependant, plusieurs plates-formes proposent des extensions sous forme de plug-ins, des modules externes, des outils associés, etc., et exploitent alors des fonctionnalités particulières. En général les plates-formes proposent des fonctionnalités de communication asynchrone (serveurs Web, serveurs d'application, forums, messageries) ou de communication synchrone (chats, NetMeeting).

L'étude des plates-formes les plus répandues (Docent, WebCT, TopClass, ATutor, Moodle, Dokeos et autres) nous a permis de dégager une activité (tâche) centrale. D'une part cette activité assure la création d'un contenu d'enseignement et d'autre part la construction d'un parcours pour l'apprenant. Le parcours définit un cadre temporel et une organisation pour les activités d'apprentissage des étudiants. A cette activité principale nous pouvons ajouter la construction de liens entre les parcours et des documents externes, la création de tests (questionnaires) et d'exercices, des moyens de communication synchrones ou asynchrones.

Les plates-formes en général n'autorisent l'intervention que d'un seul créateur de cours, ou alors elles autorisent plusieurs créateurs, mais sans gestion du partage dans ce cas. Cela veut dire que le mode collaboratif de création des cours est rarement disponible : cela est dû, à notre avis, surtout aux technologies informatiques utilisées dans ces plates-formes. L'administration des documents pédagogiques est une fonction distincte de leur création et concerne les procédures de stockage, de recherche, de partage et d'importation des documents pédagogiques. La gestion des évaluations assure des fonctionnalités permettant de faire passer des évaluations et d'administrer les résultats de ces évaluations.

3. Diffusion d'objets pédagogiques basée services Web

3.1 Les bases des services Web

Un service Web est un composant logiciel autonome possédant un URI (Uniform Resource Identifier) unique, et fonctionnant sur Internet. Les avantages des architectures services Web ont été bien utilisés dans le domaine des applications de type B2B (Business-To-Business) pour l'intégration d'applications d'entreprise et même dans des scénarios de type B2C (Business-To-Customer) (Alonso et al., 2004).

Les services Web sont des processus métiers ou des données accessibles via Internet par n'importe quel client. Ils permettent aux applications d'interagir entre elles via le Web. On peut envisager grâce aux services Web de segmenter les applications en plusieurs composants

ou services partagés, qui peuvent résider sur des machines différentes et de nature complètement hétérogènes. La communication entre les différents acteurs se fait via les langages XML et le protocole HTTP.

L'architecture des services Web s'est imposée grâce à sa simplicité, à sa lisibilité et à ses fondations normalisées. Le concept des services Web s'articule autour de trois acronymes, à savoir : SOAP, WSDL et UDDI (Chauvet, 2002). Un protocole léger basé sur XML est utilisé pour échanger des informations (SOAP). Les paramètres du service Web sont décrits avec un langage WSDL toujours basé sur XML. Une architecture répartie détient la description des services fournis (UDDI).

3.2 L'approche services Web

Nos travaux répondent à un besoin existant dans toute institution de formation où la collaboration entre plusieurs entités d'enseignement est indispensable. Pour maintenir un certain niveau, déjà acquis, l'intervention en présentiel de plusieurs enseignants est demandée, mais n'est pas toujours possible.

Notre projet XESOP propose une étude et des solutions sur les moyens technologiques des environnements e-Learning. Les résultats de ce projet se situent entre les moyens pour mettre à disposition des savoirs et les guides à distance. Les éléments essentiels du projet consistent d'une part en des supports sémantiques pour la création, la présentation et le stockage des objets pédagogiques basés sur des technologies XML (Madjarov, Betari, 2004) et d'autre part en des moyens pour la gestion du parcours de l'étudiant (Madjarov et al., 2004) et l'exécution d'exercices à distance. Etant un élément essentiel dans une formation e-Learning, l'objet pédagogique exercice est doté d'un environnement de développement à distance. Pour un exercice en algorithmique et programmation, par exemple, cela veut dire que l'apprenant serait capable d'accomplir l'exercice à distance sans avoir à sa disposition ni l'outil de développement, ni l'outil de compilation.

L'évolution de nos travaux dans ce projet est guidée par la perception que nous avons un système e-Learning, à savoir, une collection d'activités ou de processus qui agissent sur les étudiants et les objets pédagogiques. Ces processus peuvent être décomposés d'abord en éléments autonomes et par la suite réalisés et proposés comme des e-Services.

La décentralisation des fonctionnalités d'un système e-Learning classique passe par la conception des composants du service Web. Une telle approche pose certains problèmes et offre des possibilités nouvelles (Vossen et al., 2004) comme, par exemple, le contrôle du contenu proposé à l'étudiant. En effet, dans un système distribué les objets pédagogiques ne peuvent pas être simplement importés dans un système de gestion d'enseignement (LMS). Les objets pédagogiques doivent être importés à la demande. Une telle vision des choses demande de combiner les aspects techniques avec les efforts récents de standardisation (IMS Content Packaging) dans le domaine de gestion du contenu d'apprentissage visant l'échange de contenu et sa réutilisation efficace (IMS-CP, 2005).

Une des difficultés pour l'enseignement à distance est le manque d'outils pour rechercher et réutiliser des ressources pédagogiques. La spécification IMS Meta-data (IMS-Meta-data, 2005) est le processus apportant des solutions à ces problèmes.

La publication et la recherche des objets pédagogiques peuvent se faire dans un cadre UDDI faisant partie de l'architecture des services Web. Ses caractéristiques assurent le stockage de données concernant la description des objets pédagogiques, c'est-à-dire, des métas-informations, alors que le contenu réel de ces mêmes objets pédagogiques est gardé dans des sites distribués des auteurs de cours ou des intervenants d'enseignement. A partir de ce schéma de fonctionnement nous pouvons dégager certains des problèmes surgissant dans la réalisation d'une plate-forme de e-Services : le stockage du contenu pédagogique de manière distribuée, et l'échange dynamique de ce même contenu si nécessaire. En utilisant un tel

schéma fonctionnel, le contenu pédagogique serait publié et organisé pour être échangé, et serait accessible dans un environnement basé sur des services Web.

Dans un système e-Learning la variété des dispositifs et des composants peut être perçue comme une variété de processus et par conséquent réalisée en tant que services Web atomiques ou composés. Notre étude, à partir de différentes sources : (PCBI, 2003) et (Vossen, 2004), donne une liste non exhaustive composants d'un système e-Learning : conception d'un contenu pédagogique ; publication d'un cours à partir d'un contenu choisi ; gestion des objets pédagogiques ; mise à jour d'un contenu pédagogique ; adaptation d'un contenu sur demande ; recherche et présentation d'un contenu pédagogique ; inscription d'un étudiant et la gestion de son compte et profil ; évaluation des connaissances acquises, mise en place et gestion d'une classe virtuelle ; gestion d'un système de communication synchronisé de type chat. Ainsi, son fonctionnement est décomposé en différentes activités ou groupes d'activités qui peuvent être mises en application de façon indépendante sous la forme de services. Le fonctionnement intégral d'un système peut être reconstitué par une composition appropriée des services définis et choisis.

En conclusion, cette approche fournit des avantages pour un grand nombre d'apprentissages et d'étudiants. En particulier dans l'enseignement supérieur ces avantages peuvent se ressentir dans le cadre du modèle LMD (Licence, Master, Doctorat) et pour une poursuite de formation à la demande mieux adaptée aux besoins du moment. En effet, dans la société contemporaine, des besoins de formation sont nécessaires tout au long de la vie professionnelle, à cause de la mobilité et des mutations technologiques. Le développement des technologies Web apporte une flexibilité pour l'étudiant dans son choix de formation.

4. Définition des tâches et des services correspondants

La pratique usuelle démontre que les enseignants échangent rarement des informations sur le contenu de leurs cours, même quand ils sont similaires. Il y a certainement beaucoup de raisons, certains de caractère purement technique, et d'autres de caractère psychologique. Les raisons techniques résident essentiellement dans le format d'origine des documents pédagogiques. En effet, ceux-ci ont été créés dans le passé par des moyens informatiques différents incompatibles avec les formats actuels, et ne se prêtent donc pas à une interopérabilité avec les documents plus récents. D'autres affirment que l'enseignant qui fait son cours pense qu'en le partageant, il perdra le contenu ou il sera évalué par ses pairs. La raison technique évoquée n'est certainement pas le premier blocage. Avec nos travaux nous essayons d'améliorer le "climat" technique en espérant qu'avec une pratique collaborative cela conduirait à une amélioration du "climat" psychologique. Les systèmes et les plateformes dans leurs sens propre, l'idée même de l'enseignement à distance évoquent le partage des savoirs non seulement avec les apprenants, mais aussi la collaboration au sein de l'équipe pédagogique. E-Learning est un processus complexe et collaboratif par définition.

4.1 Création de documents pédagogiques

L'étude effectuée tout au long du développement de ce projet nous a amenée à la conclusion suivante : les enseignants ont besoin d'un environnement générique et standardisé, lié à des technologies nouvelles favorisant les échanges. La conception d'un tel environnement s'appuie sur l'existence de standards qui traitent la structure d'un document pédagogique (SCORM, IMS, LOM). Par exemple, à l'aide d'un éditeur sémantique XML (Figure 1), l'enseignant peut, indépendamment de ses collègues ou avec leur collaboration, créer le contenu pédagogique de son cours se basant sur les objets pédagogiques de LOM [IEEE-LOM, 2003]. Cela implique une conception de la structure arborescente d'un document

pédagogique générique et une grammaire de validation de type XML Schema (Madjarov, Betari, 2004) conforme aux normes de IMS Content Packaging (IMS-CP, 2004)

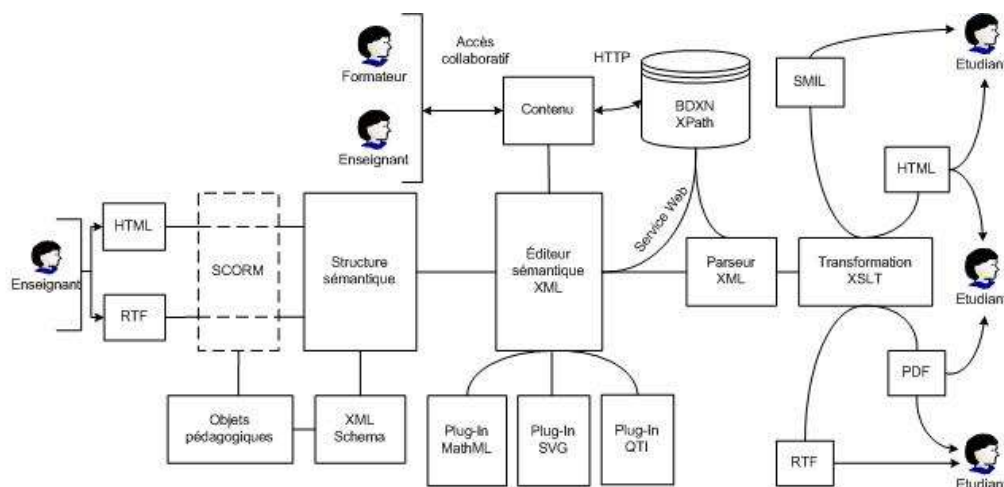


Figure 1. La gestion du contenu pédagogique dans le projet XESOP

Il est rare que le contenu d'un cours soit constitué uniquement de texte brut. Suivant les matières, l'enseignant pourrait avoir besoin de présenter des schémas, des formules mathématiques ou des informations sous forme de tableaux. Pour ces besoins nous avons trouvé autour de XML des métalangages très pratiques pour la conception d'un éditeur d'expressions mathématiques (MathML), ou bien pour la création d'un éditeur de graphisme vectoriel (SVG) et un schéma pour la réalisation des tableaux.

4.2 Stockage des objets pédagogiques

Le cours ainsi créé peut être enregistré sous forme d'un fichier XML et partagé par de simples copies en local, ou par les protocoles de l'Internet. Cela par contre n'implique pas la création d'un entrepôt de cours, pouvant être partagés et recherchés par plusieurs entités d'enseignement. Selon l'évolution de la matière, les enseignants peuvent éprouver le besoin de modifier le contenu d'un ou plusieurs cours. Ainsi, le bon fonctionnement d'un système collaboratif pour la création de cours nécessite le stockage des documents pédagogiques.

L'utilisation d'une base de données contribue à une meilleure réutilisation et diffusion de ces documents. Le choix d'une base de données adaptée s'impose donc : nous avons opté pour une base de données XML native qui permet le stockage de documents XML dans leur format d'origine, sans transformations (mappings). Notre choix, par opposition à celui d'une base de données relationnelle, s'explique par la nature même des documents pédagogiques qui sont en général de types narratifs c'est-à-dire orientés documents (document-centric) et non pas données (data-centric).

La mise en disposition d'un service de création en mode collaboratif de documents pédagogiques est certainement plus fonctionnelle. Ainsi le service d'accès collaboratif aux bases de données distribuées sur les sites des auteurs se justifie pleinement et contribue pour une meilleure réutilisation des objets pédagogiques (Figure 2).

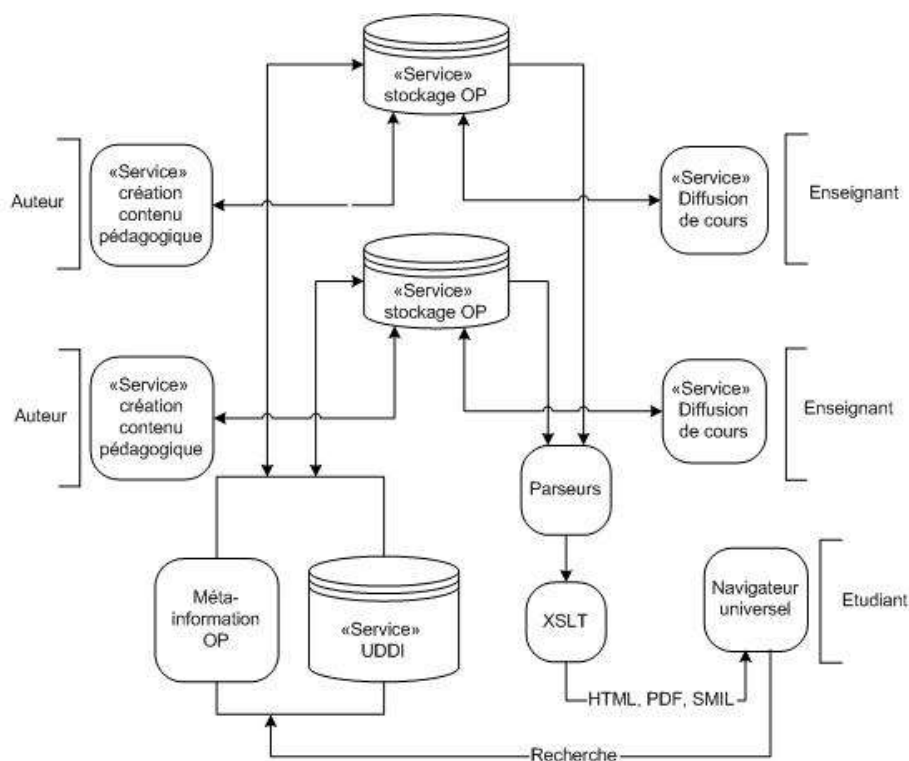


Figure 2. Les activités e-Learning présentées par des services Web

4.3 Présentation des documents pédagogiques

Un cours est destiné à un certain public d'étudiants qui n'est pas toujours homogène par son niveau de connaissances. Le cas d'un cours en algorithmique et programmation, par exemple, est assez parlant. L'enseignant en Informatique est confronté aux problèmes de différents niveaux pour les différentes options suivies par un groupe d'étudiants. Il est évident que son cours doit être adapté au niveau demandé. Cela implique dans notre système la possibilité d'extraire (par simple marquage) des parties choisies d'un cours, déjà préparé et enregistré dans la base de données, pour la composition d'un contenu adapté aux besoins du moment (transparentes pour la présentation du cours en format SMIL, pages Web pour les supports de cours en ligne – format HTML, photocopiés pour une impression et consultation ultérieure en format PDF ou RTF).

Toutes ces transformations sont effectuées à l'aide d'outils connus et accessibles : les parseurs XML. Dans une action de diffusion des documents pédagogiques à partir de la base de données chaque transformation appelle un processus transformateur qui peut être implanté comme un service Web (Figure 2).

4.4 Implantation des exercices

Partie intégrante d'un cours, l'exercice apporte une pédagogie d'apprentissage et un suivi d'acquisition des savoirs. Dans notre conception l'exercice est un objet pédagogique à part. L'auteur du cours définit l'énoncé de l'exercice et son emplacement dans sa présentation. Le cours publié par un serveur Web, devient accessible aux étudiants via un simple navigateur. A l'emplacement de l'exercice apparaît le cadre d'une applet. C'est un environnement de développement à distance (EDD), réalisé côté client, pour accéder aux ressources d'un serveur d'application. L'exécution de l'exercice en temps réel est associée à un outil basé sur un service Web. L'intérêt de cet environnement se résume en une indépendance de l'étudiant par

rapport à son système d'exploitation en lui apportant une solution efficace des étapes d'installation, de configuration des logiciels et des problèmes de licences.

4.4.1 Environnement de développement à distance

Dans le cadre d'une formation en programmation par exemple, l'étudiant aura besoin, suivant le choix du langage, d'une copie de l'interpréteur ou du compilateur pour son système d'exploitation qu'il doit installer et configurer. Il lui faudra ensuite un éditeur de texte de préférence avec la colorisation des lignes du code pour une meilleure lisibilité et compréhension du contenu. Enfin, il doit maîtriser les options de la ligne de commande de ces outils. Cela demande en plus la maîtrise d'un ensemble minimal de commandes du système d'exploitation pour la gestion des fichiers, des répertoires et de son propre compte.

Tous ces "petits problèmes" qui sont bien encombrants et qui ne portent pas sur l'essentiel de la formation, peuvent distraire l'étudiant et le détourner de l'objectif pédagogique. Par exemple lorsqu'il doit abandonner le cours pour chercher les outils de développement recommandés par l'enseignant. S'il trouve une version différente ou si l'installation du logiciel dans son système d'exploitation est incomplète il risque de se confronter à des messages et à des problèmes peu communs. En effet ces outils peuvent dans certains cas s'avérer inaccessibles car chers, ou impossible à installer sur une machine donnée, que ce soit à cause d'une contrainte de ressources, de système d'exploitation ou autre, d'où la nécessité d'un outil qui nous permettra de pallier à ces difficultés et d'améliorer l'ambiance de formation à distance. La solution apportée par l'EDD (que nous détaillons en section 5) résout ce genre de problèmes.

4.4.2 Evaluation du parcours

Le parcours effectué par un étudiant tout au long d'un enseignement est contrôlé par un service d'agent assistant. L'agent joue le rôle de médiateur entre l'apprenant et le service qui assure l'évaluation des exercices. L'intervention de ce service-agent consiste en une gestion des séquences proposées à l'apprenant suivant les réponses données à chaque pas d'exécution d'un exercice. Tout au long de son parcours des notes sont évaluées et enregistrées [Madjarov et al., 2004].

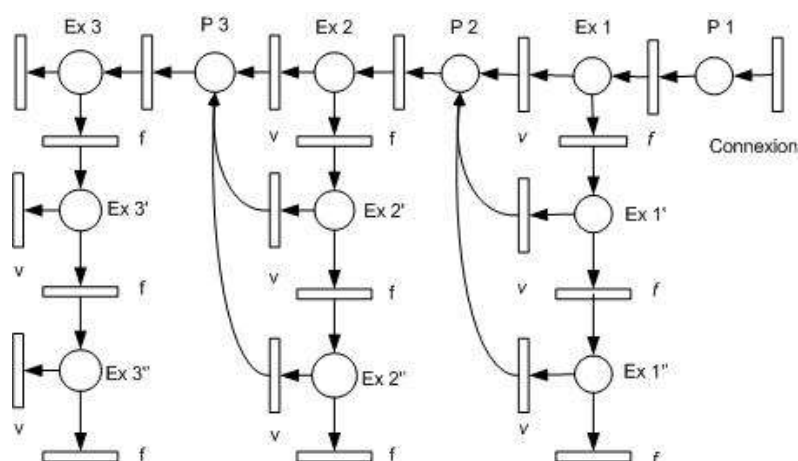


Figure 3. Modèle de gestion d'un cours représenté par un réseau de Petri

Une simulation de son fonctionnement peut être présentée par l'exemple suivant : supposons qu'un cours est composé de trois pages et neuf exercices, et qu'à chaque page correspond un exercice principal (Figure 3). A sa première connexion, l'étudiant reçoit la première page

(P1). Par la suite il passe au premier exercice (Ex1). La réponse donnée à l'exercice est renvoyé par le service et est analysée par l'agent assistant. Si la réponse est « juste » (v) le service envoie à l'étudiant la page suivante (P2) du cours. Une réponse considérée comme « fausse » (f) provoque l'envoi à l'étudiant d'un autre exercice (Ex1') du même type. Si la réponse au deuxième exercice est correcte, il lui est envoyé la deuxième page (P2) du cours, sinon c'est le troisième exercice (Ex1'') du même type qui lui est envoyé. Ce processus s'enchaîne pour les trois pages du cours et leurs exercices correspondants. Le modèle de cet enchaînement peut être décrit par un réseau de Petri. On construit le réseau de Pétri en associant une place à chaque page et une transition à tout changement de page.

5. Conception de l'environnement de développement à distance

L'EDD est un environnement programmable destiné d'une part aux auteurs de cours et d'autre part aux étudiants. L'enseignant peut spécifier un exercice dans le cadre de son cours mis à distance et l'étudiant peut exécuter cet exercice tout en poursuivant son cours.

5.1 Les fonctionnalités retenues

Pour des étudiants en informatique par exemple, cet outil propose des fonctionnalités pour l'édition de code, échange avec le système presse-papier, un exécuteur de programmes qui récupère la sortie standard du système d'exploitation en agissant en temps réel avec l'utilisateur.

Les outils de cet environnement sont développés en Java sous forme d'applet (client léger). Ils sont configurables avec un minimum de paramètres et peuvent être lancés sous plusieurs plates-formes d'enseignement à distance (Figure 4).

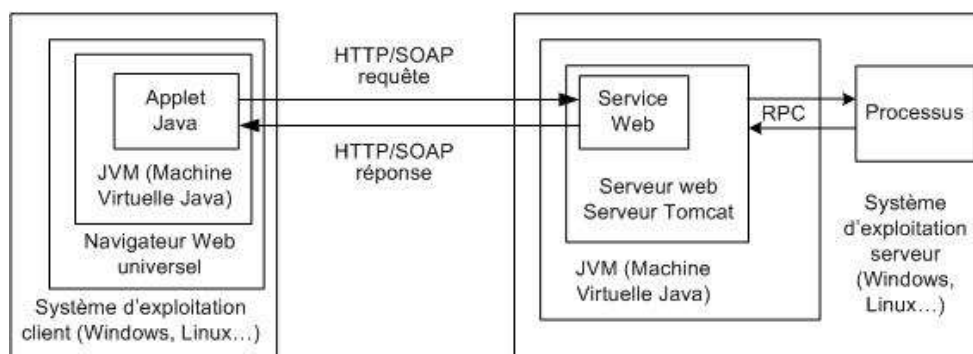


Figure 4. Schéma de fonctionnement de l'EDD

5.2 Technique d'implantation

Sur le plan technique, nous avons remarqué que les applets Java se faisaient rares dans le monde d'Internet. L'inconvénient principal qui est souvent mis en avant est la lenteur de chargement ou d'exécution de certaines d'entre elles, à cela s'ajoutent les problèmes de compatibilité entre navigateurs (SUN-Applet, 2003). Ces difficultés proviennent des différentes JVM (Java Virtual Machine) disponibles selon les versions ou les types de navigateurs. Pour des raisons de sécurité évidentes, les applets ne sont pas tout à fait des applications Java comme les autres. En effet, l'applet n'a pas accès au système de fichier local et ne peut pas non plus établir de connexion réseau autre que celle qui existe entre elle et la machine qui l'héberge. L'applet a par contre le pouvoir de réagir aux actions de la souris, du

clavier ou du système d'exploitation. Malgré cela, cette technologie n'est pas dénuée d'intérêt et nous estimons qu'avec une conception bien réfléchie et une réalisation de pointe ces inconvénients pourraient être contournés. Pour notre part, nous nous sommes orientés vers une solution basée services Web.

Le problème de sécurité ne se pose pas dans notre conception, étant donné qu'il n'est pas prévu de développer une application client pour la présentation d'un cours à distance. Ainsi l'utilisation d'un navigateur standard et gratuit nous a paru judicieuse car, c'est la technologie principale utilisée par l'enseignement à distance. Donc le développement d'un service du côté serveur et l'utilisation d'une applet pour réaliser un client léger se justifient pleinement.

L'exécution de programmes, sans que l'utilisateur ait à sa disposition les outils de développement, devient possible par la création d'un Service Web dont l'objectif est de fournir les fonctions nécessaires à la compilation et l'exécution (interprétation) de ces programmes. Sur un serveur Web hébergeant le service, sont placés les outils de développement transparents à l'utilisateur (compilateurs et interpréteurs pour différents langages de programmation). Cette technologie est ouverte (non propriétaire) et elle est développée pour les protocoles standard d'Internet (SOAP, TCP/IP). Dans notre système, la réalisation de ce service Web est basée sur un serveur Tomcat avec des interfaces de programmation en Java.

Un des objectifs de l'EDD est l'interactivité. Alors, l'interface utilisateur doit rester active lors de l'exécution des programmes. Il faut donc améliorer les algorithmes de synchronisation des Threads assurant les entrées/sorties de communication entre le client et le serveur. Dans notre implantation nous avons effectué des mesures algorithmiques qui assurent le fonctionnement stable du système d'interaction entre le client et le serveur (Madjarov et al., 2004 S).

5.3 Approche pour un développement SOAP

SOAP est un protocole de transmission de messages. Il est particulièrement utile pour exécuter des dialogues requête-réponse RPC (Remote Procedure Call). Il n'est pas lié à un protocole de communication particulier, ni à un système d'exploitation, ni à un langage de programmation (Apache-SOAP, 2003).

L'approche, adoptée dans notre projet, pour un développement SOAP est d'encapsuler le contenu du code, destiné à ce service, dans une méthode Java et ensuite, de démarrer un processus qui écoute les messages adressés à ce service (listener). Les messages SOAP contiennent le nom du service et les paramètres requis. La couche de transport est le protocole HTTP. Le listener décode le message SOAP entrant et le transforme en un appel de méthode. Ensuite il récupère le résultat et l'encode dans un message SOAP réponse (Figure 5).

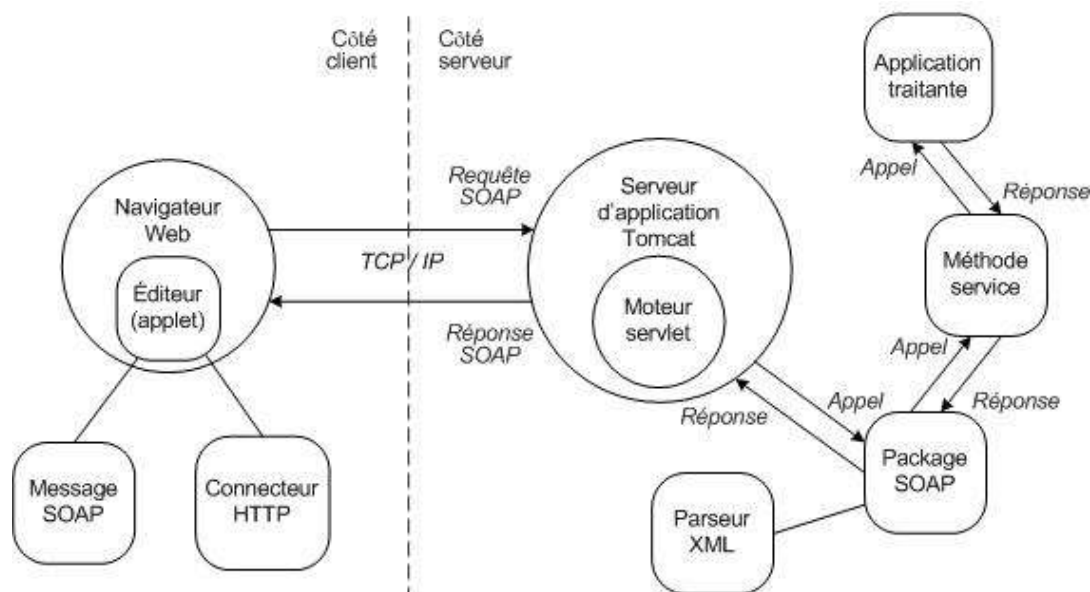


Figure 5. Implémentation du service Web pour l'exécution des exercices

L'implémentation du côté client (l'applet) demande un regard plus attentif sur les détails de sérialisation SOAP et de l'encodage de HTTP. Pour cela nous employons un package SOAP en étendant ses fonctionnalités pour assurer la stabilité de l'applet et les interactions de l'apprenant avec le système. Nous invoquons le service en spécifiant l'URL du service, le nom du service et tous les paramètres requis. Le serveur renvoie un message encodé HTTP contenant la réponse SOAP. Du côté client, nous nous reposons sur le même package SOAP pour exécuter l'inverse. Le package décode le message HTTP et extrait le message SOAP, ensuite le désérialise et obtient la valeur de retour.

5.4 Mise en œuvre du service

Les paramètres de configuration du service Web sont dans un fichier XML qui contient les informations nécessaires pour l'exécution des programmes dans chaque langage spécifié. A titre d'exemple nous présentons la configuration pour le traitement des programmes de trois langages : C/C++ et Java.

L'élément racine du document de configuration est `<coursconfig>`. Il peut contenir un ou plusieurs éléments `<language>`. Chaque élément n'a qu'un seul attribut `name` pour le nom du langage que le service reconnaît. La définition d'un langage peut contenir un ou plusieurs éléments (`<step>`) chacun décrivant la commande à exécuter pour le traitement du code. La balise `<command>` définit la commande à exécuter (`name = "lcc"`) par un chemin d'accès relatif ou absolu à l'environnement système du côté serveur. La balise `<file>` définit le nom du fichier par l'attribut `id`, l'extension de ce fichier par l'attribut `extension = ".cpp"` et un contenu obligatoire par l'attribut `hasContent = "yes"`.

Le cas des langages C/C++ contient trois "step" (étapes) à exécuter. En premier on définit la compilation du code source à partir d'un fichier ".cpp", en deuxième le processus d'assemblage des fichiers ".obj" et la création du fichier exécutable, et en troisième l'exécution du programme. En cas d'erreurs à la compilation ou aux "step" suivants l'arrêt du processus est assuré par l'attribut `stopOnError = "yes"`.

```
<?xml version = "1.0"?>
<coursconfig>
  <language name = "c">
    <step stopOnError = "yes">
```

```
<command name = "lcc"/>
<file id = "FichierC" extension = ".cpp" hasContent = "yes"/>
<file option = "-Fo" id = "fichierObj" extension = ".obj"/>
</step>
<step stopOnError = "yes">
  <command name = "lclnk"/>
  <file id = "fichierObj"/>
  <option name = "-o"/>
  <file id = "fichierExe" extension = ".exe"/>
</step>
<step>
  <file id = "fichierExe"/>
</step>
</language>
</coursconfig>
```

L'exemple ci-dessous - pour les programmes écrits en langage Java - illustre le fonctionnement de l'attribut `folderid = "testFolder"` de la balise `<file>` qui indique au service Web de placer le fichier traité dans un répertoire séparé.

```
...
<language name = "java">
  <step stopOnError = "yes">
    <command name = "javac"/>
    <file id = "FichierJava" name = "test.java" folderid = "testFolder"
      hasContent = "yes"/>
  </step>
  <step>
    <command name = "java"/>
    <option name = "-classpath"/>
    <file folderid = "testFolder"/>
    <option name = "test"/>
    <file id = "FichierClass" name = "test.class" folderid =
      "testFolder"/>
  </step>
</language>
```

...
Nous avons déjà mentionné que l'utilisation d'une applet du côté client est soumise à plusieurs contraintes de sécurité. Une applet évolue dans le cadre d'un navigateur Web et ne peut pas avoir d'accès direct aux ressources systèmes sauf exception liée à la procédure de signature des droits d'accès (SUN-Applet, 2003). Dans notre cas, nous avons besoin d'accéder aux ressources réseaux, et au presse-papier (clipboard) de la machine client. A cela s'ajoute le besoin de créer des fichiers sur le serveur. Ces possibilités seront offertes à condition que le client accepte la signature à la demande du navigateur.

La procédure de signature est une technique réalisée par les packages Java avec des outils de création des clés cryptées pour les certificats émis à chaque connexion du client (keytool). La signature elle-même se fait par la signature de l'archive `.jar` où se trouvent les classes accédant aux ressources autorisées (jarsigner).

Pour faciliter cette procédure nous avons créé un outil `genkey`. Ainsi l'utilisateur pourra désormais créer ses propres clés enregistrées dans une base de données. L'identification d'un client se fait alors à partir de cette base de données avec son nom d'utilisateur et son mot de passe.

6. Conclusion

Dans cet article, nous avons défini les composants et les fonctionnalités de base d'un système e-Learning. Nous avons montré qu'un système d'enseignement à distance peut être décomposé en plusieurs activités réalisées comme des applications autonomes sous la forme de services. Nous avons décrit aussi comment un contenu pédagogique peut être créé et diffusé dynamiquement dans un système e-Learning distribué basé sur des services Web.

Nos études ont montré que la plupart des solutions existantes, qui peuvent gérer d'une manière dynamique les contenus pédagogiques, se situent toujours dans le cadre des systèmes centralisés et emploient des formats propriétaires. Cela pose d'autres problèmes, dont celui de l'interopérabilité des systèmes e-Learning. Nous estimons que les technologies des services Web peuvent contribuer à la résolution de ce genre de problèmes.

Les quelques cas d'utilisation des services Web dans un système e-Learning distribué (Vossen, 2004) restent pour le moment à un niveau conceptuel, et sont déployés sur les "gros moyens" des systèmes dits d'entreprises. Notre approche s'appuie sur les principes simples en utilisant le plus possible des concepts standards et outils "open source".

Nous avons représenté un exemple d'implantation des services Web dans un système e-Learning pour l'animation d'exercices. L'environnement que nous avons réalisé reste dans les normes des objets pédagogiques et permet aux étudiants de pratiquer des apprentissages divers. Un prototype du système est en cours de teste et évaluation.

Bibliographie

Abel M.-H., Lenne D., et al., (2003) Gestion des ressources pédagogiques d'une e-formation, Document numérique, vol. 7, 111-128.

Alonso G., Casati F. et al. (2004) Web Services – Concepts, Architectures and Applications, Springer Verlag Berlin, ISBN 3-540-44008-9.

Apache, Project Apache SOAP. 2003, En ligne : <http://jakarta.apache.org/soap> .

Bouthry A., Jourdain C., (2003) Construire son projet de formation en ligne, Éditions d'Organisation, ISBN 2-7081-2854-X.

Catalyst (2003), Technical Evaluation of selected Learning Management Systems. The Open polytechnic of New Zealand.

Chauvet J. M., (2002) Services Web avec SOAP, WSDL, UDDI, ebXML..., Eyrolles, Paris.

IEEE (2003), Position Statement on 1484.12.1, 2003, Learning Object Metadata (LOM) Standard Maintenance Revision, En ligne : <http://ltsc.ieee.org/wg12/>.

IMS GLC. (2004) IMS Content Packaging v1.1.4 final specification, November. <http://www.imsglobal.org/content/packaging/> .

IMS GLC. IMS Learning Resource Meta-data Specification v1.3 Public Draft. May. <http://www.imsglobal.org/metadata/> .

Madjarov I., Betari A., (2004) Un éditeur XML sémantique pour objets pédagogiques stockés dans une base de données XML native, In Les Nouvelles Technologies de la Répartition, (NOTERE 2004), Juin, Saidia, Maroc, 218-233.

Madjarov I., Boucelma O., Betari A., (2004) An Agent- and Service-oriented e-Learning Platform, Proceedings of Third International Conference on Web-Based Learning (ICWL 2004), August, Beijing, China, LNCS 3143, Springer-Verlag, 27-34.

Madjarov I., Betari A., Shishedjiev B., (2004) S) Remote Development Environment In An E-Learning System, Proceedings of 18-th International Conference SAER 2004, September, Varna, Bulgaria, 178-182.

ORAVEP, Étude comparative technique et pédagogique des plates-formes pour la formation ouverte et à distance, Rapport d'étude pour DT/SDTETIC, 2000.

PCBI, (2003) Etude des outils de gestion de ressources numériques pour l'enseignement, Ministère de la jeunesse, de l'éducation nationale et de la recherche, juin.

SCORM, (2004) Sharable Content Object Reference Model, Advanced Distributed Learning, <http://www.adlnet.org/>.

SUN-Applet, (2003) Advanced Programming for the Java 2 Platform, Chapter 10: Signed Applets, <http://java.sun.com/>.

Vossen G., Westerkamp P. Maintenance and Exchange of Learning Objects in a Web Services Based e-Learning System. *Electronic Journal of e-Learning* Volume 2 Issue 2 2004. 292-304.

Walckiers M., De Praetere T., L'apprentissage collaboratif en ligne. *Distances et savoirs*. Volume 2 - n° 1, 2004, 1-23.