

Vue globale des ressources disponibles sur Dokeos et Moodle

R. Babel, A. Ouadahi

May 24, 2012

1 Introduction

L'Université de Genève dispose de deux plate-formes d'apprentissage en ligne (ou LMS), à savoir Moodle et Dokeos. Les professeurs peuvent selon leurs préférences utiliser l'une ou l'autre de ces plate-formes, voir les deux. Pour les étudiants, cela amène le problème suivant: il n'existe pas de moyen de synthétiser l'ensemble des cours auxquels il est inscrits.

Il s'agit là d'un problème réel, que nous avons eu nous même l'occasion de rencontrer: un cours existait à la fois sur Dokeos et sur Moodle, mais seul l'un des deux était utilisé. Cette mésaventure est l'une des raisons qui nous a poussé à choisir ce sujet.

Le projet que nous avons choisis tâche donc d'apporter une réponse à cette problématique, via la création d'un outil qui réunisse des informations sur les cours présents sur les deux plate-formes.

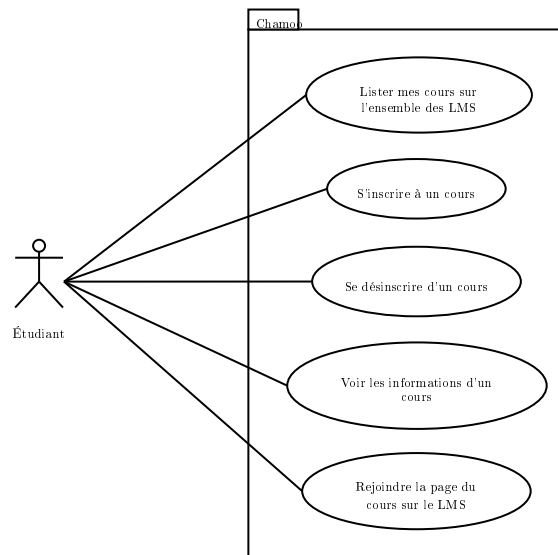


Figure 1: Diagramme des cas d'utilisation

2 Présentation des LMS

Un LMS (pour Learning Management System), qu'on appelle plate-forme d'apprentissage en ligne en français, est un système web créé pour accompagner les personnes impliquées dans un processus d'apprentissage dans leur gestion de parcours pédagogique.

Les deux LMS utilisés à l'Université de Genève sont Moodle et Dokeos (voué à être remplacé par l'un de ses successeurs, Chamilo). Ils sont essentiellement utilisés pour la mise à disposition des matériels de cours, mais disposent d'une large gamme de possibilités pas toujours exploitées.

Dans la mesure où notre projet visait à synthétiser les ressources présentes sur les deux plate-formes, une bonne connaissance de leur fonctionnement, et en particulier de leur code source était nécessaire. La prise en main de ces deux LMS a donc été une part importante de notre travail.

2.1 Chamilo

Chamilo, qui est un fork de Dokeos (c'est à dire un nouveau logiciel issu de son code source), est principalement développé en PHP. Pour notre projet, nous avons travaillé avec la version 1.8.8.4.

Le développement de Chamilo est très orienté utilisateur, puisqu'il consistait à rassembler les différentes modifications effectuées par ses utilisateurs pour élaborer les nouvelles versions. De ce fait, son code source est parfois un peu hétéroclite. Cette politique est facilitée par le fait que son code source soit distribué sous licence libre. Néanmoins, il présente l'avantage de coller au plus proche des attentes de son public.

2.2 Moodle

Moodle, qui est à ce jour le LMS le plus répandu au monde, est lui aussi développé en PHP. Nous l'avons utilisé dans sa version 2.2.2.

Bien que Moodle soit lui aussi un logiciel libre, son processus de développement est plus structuré, et de ce fait son code source est nettement mieux organisé. Il possède de plus des outils très puissants comme, entre autres mais c'est le point qui nous intéressait le plus, un excellent système de services web.

3 Solutions Choisies

Nous avons choisis de baser la conception de l'interface de notre projet sur une partie du mémoire de Hamid Fatine et Reto Sutter, "Prototypage d'un Portlet E-learning à l'Université de Genève". Celui-ci contenait en effet le prototype d'un module "Mes Cours", qui présentait une grande partie des fonctionnalités dont nous avons besoin, et avait l'avantage de reposer sur une étude faite auprès du public cible de notre projet, à savoir les étudiants de l'université.

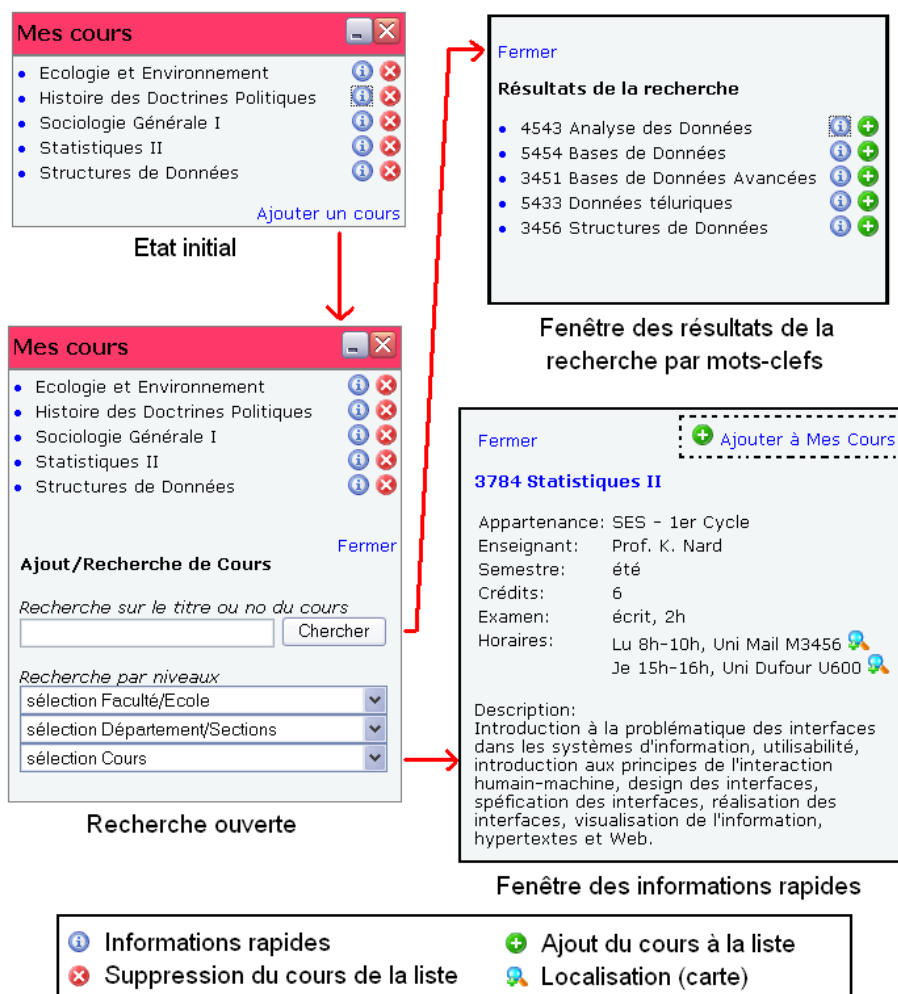


Figure 2: Prototypé du module “Mes Cours”

Nous avons donc conçu notre applicatif comme un module pouvant être réutilisé dans un cadre plus large, et calqué son ergonomie sur celle du prototype du le mémoire cité précédemment et présenté à la figure 2.

Du fait que ce prototype n'avais pas été conçu comme un lien entre les deux LMS, nous y avons ajouté deux fonctionnalités pour mieux remplir ce rôle: La liste des cours indique via une icône la plate-forme à laquelle il correspond, et la page du cours sur le LMS est rendue accessible par l'intermédiaire d'un lien.

Pour réaliser notre projet, nous avons besoin de moyens techniques pour récupérer des données issues des deux LMS différents, afin de les réunir à l'intérieur de notre applicatif unique. L'utilisation de services web pour assurer ce besoin nous a paru la solution la plus évidente et la plus propre, d'autant plus que Moodle possédait déjà des possibilités très complètes en la matière.

Nous avons donc choisi d'utiliser le système de services web de Moodle, paramétré avec le protocole XML-RPC pour Moodle, et de développer un simple service web REST pour Chamilo, bien que l'existence d'un service à base de SOAP pour cette plate-forme aie été mentionné.

Il va sans dire que, qui dit services web, qui plus est services web qui permettent d'effectuer des actions relativement sensibles comme l'inscription ou la désinscription aux cours sur les LMS, dit nécessité de sécuriser l'application.

Tout d'abord, l'accès aux services web eux mêmes. Pour ce qui est de Moodle, nous avons choisis d'utiliser la solution proposée par la plate-forme, fonctionnant sur un système de jetons attribués à un utilisateur, et pour Chamilo nous avons choisis d'effectuer une vérification par IP, pour simplement interdire les requêtes autres que celles provenant de sources autorisées.

Ensuite, nous avons choisis de penser l'application comme devant nécessiter une identification à travers le système Shibboleth qui gère également les autres ressources de l'université.

4 Implémentation

4.1 Service Web Moodle

La plupart des fonctions dont nous avons besoins étaient déjà présentes dans le service web proposé par Moodle, à l'exception d'une fonction permettant de faire la correspondance en l'username et l'id base de donnée d'un utilisateur. Dans la mesure ou nous devons de toute façon créer un plugin de service web pour ajouter cette fonctionnalité, nous avons également ajouter une série d'alias aux fonctions préexistantes, afin de n'avoir à gérer qu'un seul service pour le fonctionnement de notre application.

Le service web se présente donc sous la forme d'un plugin Moodle, c'est à dire un dossier à placer dans le répertoire "local" de Moodle. Celui-ci le proposera alors à l'installation. Pour fonctionner, il nécessitera encore l'activation du protocole XML-RPC, ainsi que la création d'un token associé à un utilisateur possédant les droits correspondant aux actions utilisées (inscription/désinscription, consultation de tous les cours, consultations des cours d'un utilisateur).

4.2 Service Web Chamilo

Pour Chamilo, nous avons donc choisi de créer nous même le service web, néanmoins, nous avons toujours utiliser les objets et méthodes fournies par les librairies de chamilo pour effectuer les actions que nous voulions, afin d'avoir une bonne intégration à la plate-forme avec tous les avantages qui en découlent (compatibilités avec les versions futures, vérifications en aval sur l'intégrité des données, robustesse...).

Le service web se présente sous la forme d’un dossier à mettre dans le répertoire “plugin” de Chamilo. Une fois placé, il fonctionnera sans manipulation supplémentaire.

Si l’on désire modifier les IP acceptées par le service web, il faudra modifier le fichier “sec.inc.php” présent dans le dossier du service web. Actuellement, seule une IP identique à celle du serveur sur lequel est installé Chamilo est acceptée.

4.3 Librairie de fonctions uniques

Afin d’avoir à éviter de distinguer les actions effectuées sur Moodle ou sur Chamilo, nous avons choisis de mettre en place une librairie de fonctions uniques communes aux deux plate-formes, que nous réutiliserions dans le reste de notre application. Cette librairie est contenue dans le fichier “courses.inc.php”. La distinction entre les deux plate-formes n’est dès lors plus fait qu’à travers un attribut “platform” associé à chaque objet décrivant un cours.

Cette librairie met à disposition les fonctions suivantes:

- “get_user_courses”, qui retourne la liste des cours de l’utilisateurs dont on a fourni l’username
- “get_courses”, qui retourne la liste de tous les cours présents sur les deux LMS
- “get_course_info”, qui retourne les informations associées à un cours donné
- “subscribeToCourse”, qui tente d’inscrire l’utilisateur à un cours
- “unsubscribeFromCourse”, qui tente de désinscrire l’utilisateur d’un cours

Ces cinq fonctions résumes les cinq actions différentes que nous effectuons sur les deux LMS. La totalité de notre application est basées sur les utilisations permises par ces cinq actions.

4.4 Interface dynamique avec AJAX

Si l’affichage au chargement de la page est effectuée par un simple script PHP générant du XHTML, et les fonctionnalités les plus simples (affichage du formulaire de recherche, drag&drop) par du Javascript “standard”, l’essentiel des possibilités de notre interface sont rendues possibles grâce à AJAX.

Nous avons donc implémenté cinq scripts PHP voués à être appelés via AJAX, les quatre premiers correspondent aux fonctions de la librairie unique à l’exception de “get_user_courses” dont nous n’avions pas l’utilité, et le cinquième permet de récupérer les variables de configuration, afin de n’utiliser qu’un seul fichier commun pour le code PHP comme pour le code Javascript. Néanmoins, pour des raisons évidentes de sécurité, seules les variables de configuration “non sensibles” sont transmises.

Nous avons choisi de faire apparaître les diverses informations (description de cours, résultat de recherche) pouvant être obtenues par le biais de notre application dans des boîtes d’informations déplaçables. Ces boîtes ne peuvent exister qu’en un seul exemplaire, pour éviter de trop surcharger visuellement l’application de manière inutile. L’utilisateur peut également refermer les boîtes dont il n’a plus besoin.

5 Conclusion

5.1 Objectifs Atteints

Nous avons réalisé une application simple, efficace et utilitaire, qui effectue l’essentiel des points prévus dans notre environnement de test.

L’application a été pensée afin de pouvoir être sécurisée de manière satisfaisante.

L’application a été testée (nous l’espérons de manière suffisamment exhaustive) avec les navigateurs les plus courants (Firefox, Chrome, Internet Explorer).

L’application respecte à priori les standard du W3C (il est néanmoins difficile de valider efficacement une page dont une grande partie du contenu est généré via Javascript).

Paramètres faciles à adapter. Un seul fichier de configuration, possibilité d’utiliser le fichier de configuration pour changer les classes CSS associés aux divers éléments pour unifier l’application avec un projet de plus grande envergure. Les parties du code pouvant nécessiter des modifications sont placées dans des fichiers séparés et très courts (“sec.inc.php”, “getUser.inc.php”).

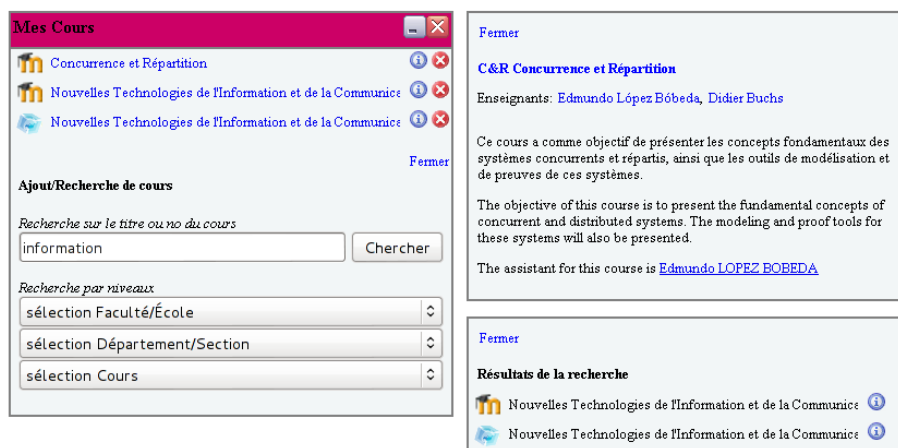


Figure 3: Rendu final de l’application

5.2 Améliorations Possibles

Implémentation de la recherche par niveau. Nécessite potentiellement l'utilisation de ressources extérieures à Moodle et Chamilo.

Description des cours plus détaillée (et plus proche du prototype) en utilisant des ressources extérieures à Moodle et Chamilo.

Alerte/Indication pour prévenir de la présence de nouveau contenu dans un cours.

5.3 Difficultés rencontrées

Prise en main du code source des LMS. En particulier pour Chamilo, qui est très hétéroclite, et parfois un peu redondant aussi. Moodle est bien structuré, une fois qu'on a compris le principe.

Une série de problèmes d'encodage (apparemment les deux LMS étaient pas configurés de la même manière). Résolu à prioris, même si peut être un réencodage en trop d'effectué dans le service web pour Chamilo.

5.4 Avis Personnels

Un projet intéressant et motivant (un résultat relativement concret à un problème que nous avons personnellement rencontré), qui utilise un éventail assez large des technologies du web abordées pendant le cours (xHTML, CSS, PHP, Javascript, AJAX, service web).

Des contacts instructifs, en plus d'être sympathiques et encourageants avec nos référents de projet.

Bien sur, en temps qu'étudiants en master d'informatiques, nous n'étions pas à notre premier pas dans l'univers de la programmation web, mais néanmoins, nous avons eu l'occasions de faire un certains nombre de choses nouvelles au cours de ce projet.