Serveur de cédérom
Revision: 1.6 — proposé à TICE 2004

Christian Queinnec
Université Paris 6 — Pierre et Marie Curie
LIP6, 4 place Jussieu, 75252 Paris Cedex 05
France – Email: Christian.Queinnec@lip6.fr

Résumé

Fournir, sur un cédérom, des documents et des logiciels appropriés pour un enseignement est insuffisant pour permettre un véritable travail à distance. Cet article se fonde sur quatre années d'expérimentation de production de cédéroms pour l'enseignement de langages de programmation. Il décrit la notion de « serveur de cédérom » rendant dynamique et interactif un contenu a priori passif. L'article recense les fonctionnalités attendues ainsi que les contraintes techniques sous-jacentes (authentification notamment) et propose, enfin, des standards pour la communication entre les différents composants évoqués.

Mots-clés : Travaux pratiques à distance, Campus numérique, Environnement numérique de travail

Abstract

A CD-ROM full of documents and programs is not sufficient to ensure enough comfort for distance lab work. This article results from four years of experimentation during which we delivered educational CD-ROMs on computer languages to university students. We describe the concept of a "CD-ROM server", its services and some technical constraints (namely authentication). It also proposes some XML standards for the communication between the various components.

Keywords: Distance lab work, Implementation and uses of innovative systems in higher education, Standards and interoperability.

1 Introduction

Par rapport au polycopié classique, le cédérom a l'avantage d'être meilleur marché et de pouvoir contenir une grande quantité de documents et de programmes. Il a pour défaut d'être un support d'information non modifiable, non annotable et de ne pouvoir être lu dans les transports en commun. Néanmoins il devient, depuis quelques années, un support de plus en plus employé à l'UFR d'informatique de l'UPMC (environ 4000 exemplaires diffusés aux étudiants de DEUG MIAS et de licence d'informatique) qui permet de diffuser cours et exercices, environnements de programmation et bibliothèques additionnelles, bandes-son et même vidéo enrichie de certains cours. Grâce à ces cédéroms, les étudiants trouvent tout ce qui leur est nécessaire pour pouvoir travailler à distance, à leur rythme et avec les mêmes outils que ceux déployés en salle de travaux pratiques sur ordinateurs. Rappelons que plus de 85% des étudiants entrant en DEUG MIAS à l'UPMC ont un ordinateur à leur disposition chez eux mais que seulement 50% disposent d'une connexion à Internet.

Le cédérom est donc la matérialisation du lien unissant l'étudiant et son enseignant. Il se doit donc d'être interactif — tant pour l'étudiant qui doit pouvoir, connecté ou pas à Internet, travailler avec, se faire corriger plus ou moins automatiquement, faire connaître ses résultats — que pour l'enseignant qui doit pouvoir suivre pédagogiquement l'étudiant et, dans ce cadre, obtenir des informations sur sa progression dans son travail (quels exercices ont été tentés et avec quelles réponses, quelles pages ou quels sites ont été consultés, etc.)

Sur la base des expériences précédemment menées [Que00a, BDM+02, QC02], toutes liées à l'enseignement des langages de programmation ou utilisant un tel langage comme outil omniprésent, cet article résume les problèmes rencontrés, synthétise les objectifs souhaités et propose une architecture pour un tel « serveur de cédérom » adapté à cette problématique.

2 Objectifs

Il serait regrettable de produire aujourd'hui un cédérom plein de documents passifs qu'il ne s'agirait que de lire. L'interactivité est nécessaire autant pour naviguer entre les documents que pour interagir avec ces mêmes documents lorsque ce sont des programmes. Pour procurer quelque interactivité, des pages dynamiques (c'est-à-dire des documents dont l'affichage est le résultat d'un calcul, cf. par exemple [QG01a]) sont nécessaires ce qui impose la présence — d'un interprète de ces pages dynamiques afin de pouvoir les élaborer et — d'un visualiseur de ces mêmes pages. Nous avons choisi d'utiliser un navigateur pour la visualisation qui, bien configuré, sait s'accommoder d'une grande diversité de documents (HTML, PDF, PostScript, texte, GIF, PNG, etc.). Pour la génération de pages dynamiques, nous avons recouru à un serveur HTTP (Tomcat [The] écrit en Java) incluant servlets ou autres JSP (Java Server Pages).

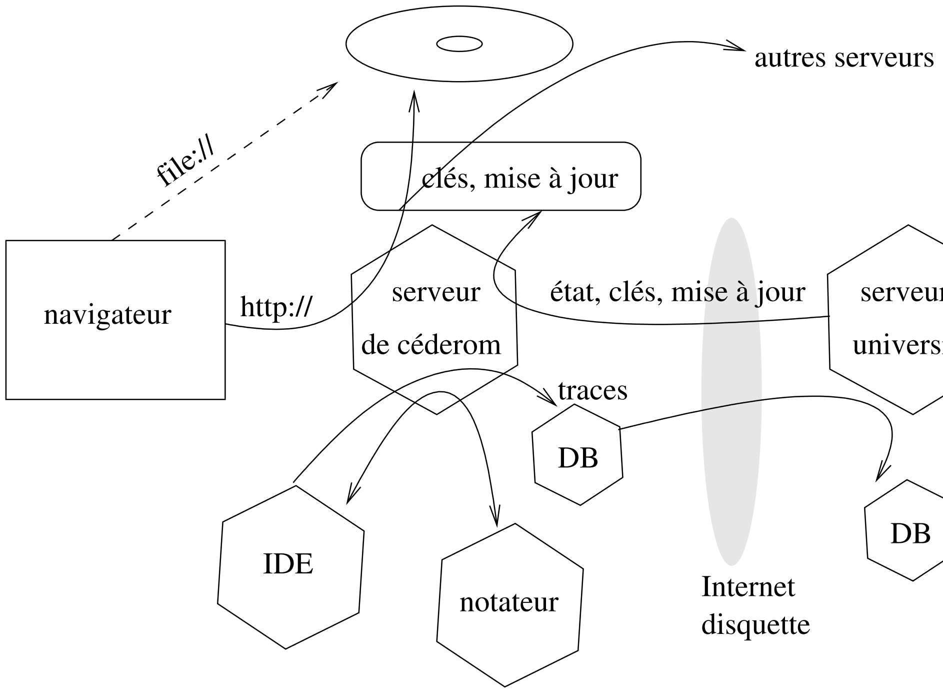

FIG. 1 – Grand schéma du serveur et de son contexte

Si le navigateur est un outil usuel largement diffusé, ce n'est pas le cas du serveur HTTP qui doit, lui, venir avec le cédérom. Pour composer avec la multiplicité des systèmes d'exploitation auxquels le cédérom doit faire face : les multiples Windows, Unix et autres MacOS, il n'y a pas grand chose à faire sinon utiliser un langage portable sur ces systèmes (Java est un bon choix), développer des procédures de reconnaissance fiables de ces systèmes et de leurs caractéristiques enfin placer les programmes en de multiples versions binaires.

2.1 Services techniques

Une fois lancé, le serveur de cédérom et tous les services qu'il procure (parfois sous la forme de serveurs indépendants mais harmonieusement couplés) doivent pouvoir être administrés à partir du navigateur. Ces services peuvent être simplement paramétrés par des pages dynamiques agissant sur le serveur et sa configuration. Le premier service est de permettre l'affichage des documents statiques ou dynamiques, présents sur le cédérom ou recopiés en disque dur afin d'améliorer la vitesse d'accès. Le serveur de cédérom peut également servir de relais (ou *proxy*) pour le navigateur.

Un problème observé avec ce service est que la consultation directe des documents du cédérom hors serveur (consultation qui ne peut être interdite si l'on utilise un explorateur de fichiers, cf. figure 1) reste possible mais ne fonctionne qu'avec les documents statiques (HTML, PDF mais pas JSP). Maquiller le type de tous les documents afin de les rendre tous illisibles directement est peu robuste surtout si un étudiant n'arrive pas à installer le serveur de cédérom : le cédérom serait alors totalement inutilisable ce qui est un risque trop grand. Une autre solution est d'enseigner aux étudiants les différences (de plus en plus ténues) entre les protocoles `file:` (accès direct) ou `http:` (accès via le serveur), entre un explorateur de fichier (accès direct) et un navigateur (accès via le serveur) : c'est incompréhensible pour le plus grand nombre ! L'expérience nous a fait rejeter ces deux solutions au profit d'une troisième (qui sera expérimentée l'an prochain) où le serveur de cédérom s'affichera, lorsque lancé, sous la forme d'une icône. L'icône changera d'aspect lorsqu'elle traitera un document. Ce retour visuel devrait aider à faire comprendre ce qui se passe.

Une fois gravé, un cédérom n'est malheureusement plus modifiable. Un second et important service dit de mise à jour, pallie ce défaut. Le serveur de cédérom dispose d'un espace où il peut écrire en disque dur, il masque alors certaines parties du cédérom avec de nouvelles versions, des rectificatifs (des programmes de correction) ou de totalement nouveaux documents. Ces mises à jour doivent pouvoir être téléchargées sur le site de l'université (en présence d'une connexion à Internet) ou être convoyées par disquettes lorsque de faible taille.

Le troisième service gère le déverrouillage de documents présents sur le cédérom mais dont la lecture, ou plus généralement l'accès, n'est permis que sous certaines conditions (de dates ou de réussite à certaines épreuves). Ces documents sont simplement cryptés et le serveur sait gérer les clés de déchiffrement. Le déverrouillage se fait étudiant par étudiant : une clé de déchiffrement d'un document d pour l'étudiant e n'est pas réutilisable pour un autre étudiant.

Le trousseau de clés fait partie de l'état de l'étudiant vis-à-vis du serveur de cédérom. Cet état doit être transmissible à d'autres machines afin d'assurer des possibilités de nomadisme, ainsi qu'être sauvegardable sur le serveur de l'université en cas de réinstallation du cédérom sur machines personnelles.

Le cinquième service dit de parcours permet de définir des enchainements de pages et d'exercices élaborés. Ce service est à base de continuations et est implanté par un interprète Scheme. Ce service a été largement décrit en [Que00b, Que02].

2.2 Traces

Le travail que l'étudiant effectue à l'aide du cédérom fait l'objet de traces qui enregistre sa progression : choix d'exercice, réponses diverses, notations ou annotations synthétisées par programme et obtenues en retour, pages consultées, etc. En effet, nos cédéroms axés sur l'enseignement de langages de programmation comportent des questionnaires et des exercices. Les questionnaires sont des QCM plus ou moins élaborés puisque certains d'entre eux font appel au notateur automatique afin d'apprécier les réponses à des questions comme « quelle est la valeur de telle expression ? » ou « écrire une expression dont la valeur est ... ? ». Les parcours liant ces questions sont eux-mêmes élaborés car il est possible de limiter le nombre de mauvaises réponses et de sauter l'exercice si besoin, de décider dynamiquement en fonction des résultats précédents du prochain exercice, etc. Tous ces questionnaires sont mis en œuvre via le navigateur.

Les exercices sont plus conséquents : l'étudiant écrit des programmes qui sont notés par des procédures automatiques [QC02, Que02] mises en œuvre localement et instantanément, cf. figure 1. Nous sommes intéressés à la fois par suivre pédagogiquement l'étudiant de manière à pouvoir commenter les programmes qu'il a commis, connaître son opiniâtreté (au fil de ses différents essais) mais nous sommes aussi intéressés à améliorer nos procédures de notation automatique et de nous assurer de leur pertinence. À titre d'exemple, nous avons aujourd'hui recueilli plus de 7000 réponses aux exercices proposés ce qui nous permettra d'étudier la validité de nos notateurs, la taxonomie des erreurs les plus courantes, etc.

Les traces sont accumulées sur le disque dur et sont acheminées au serveur de l'université lorsqu'une connexion à Internet est présente ; elles peuvent également être acheminées par disquette via les ordinateurs en libre service à l'université. Le sixième service dit d'acheminement des traces est l'un des plus techniques et des plus critiques du serveur de cédérom mais c'est sur lui que repose le suivi pédagogique de l'étudiant ainsi que l'analyse de la pertinence des notateurs automatiques de programmes.

Le comportement est simple : les traces sont enregistrées dans une base de données locale (Mckoi écrit en Java [Dow]) afin de pouvoir résister à des pannes impromptues (et ne perdre aucune trace) et de pouvoir supporter de longs moments sans connexion à Internet. Une tâche du serveur de cédérom envoie ces traces au serveur de l'université (où elles sont stockées dans une autre base de données) puis les élimine de la base locale. Les traces peuvent également être convoyées par disquette et lorsque son contenu est dans les bases de l'université, un mot de passe obtenu en retour permet d'éliminer ces mêmes traces de la base locale. Les traces sont lisibles par leurs auteurs sur le serveur de l'université.

Il y a toutefois des contraintes spécifiques à prendre en compte : *(i)* authentifier l'auteur des traces, *(ii)* interdire que des traces puissent être modifiées, *(iii)* interdire que des traces puissent être forgées.

Le premier problème est réglé par un système de certificats. Lorsque l'étudiant installe son cédérom, un couple de clés publique et privée est créé, un certificat auto-signé par l'étudiant et incorporant son mot de passe pour le serveur de l'université (ainsi que sa clé privée) est engendré et installé comme première trace. Lorsque la première trace est acheminée, la clé privée est effacée et chaque trace est cryptée à l'aide de la clé publique à partir de laquelle on ne peut reconstituer la clé privée désormais détenue par le serveur de l'université. Les traces sont donc des messages cryptés, ceci assure l'authentification de leur auteur et les rend non modifiables et non réemployables par d'autres étudiants indélicats ce qui résoud en même temps le second problème.

Le troisième problème ne peut, quant à lui, être résolu. Si le serveur de cédérom sait émettre des traces et qu'il tourne sur la machine d'un étudiant alors il peut être détourné pour synthétiser n'importe quelles traces. Il n'est possible que de rendre difficile ce procédé : le code est caché, il est séparé en deux moitiés dans des paquetages différents qu'il faut réassembler pour le rendre opérationnel, une partie est cryptée avec la clé publique du serveur de l'université, il utilise un contexte variable basé sur l'ensemble des traces émises et bien reçues, etc. Toutefois comme le source du serveur est distribué sous licence GPL et est présent sur le cédérom, c'est un véritable défi que de cacher ce procédé !

Si l'on fait fi du problème précédent, le mécanisme de traces permet, comme les traces sont datées et authentifiées, d'envisager des interrogations écrites, en temps limité, à domicile, associées à une remise des résultats avant une date limite. Des outils de détection de non-plagiat peuvent être utilement mis en œuvre côté université.

2.3 Couplage avec IDE

Nos cédéroms sont axés sur l'enseignement de langages de programmation. L'écriture de programmes par les étudiants requiert des outils qui ne sauraient être approchés, même de loin, par les navigateurs dont l'interface est par trop frustre. Ces environnements de développement ou IDE (pour *Integrated Development Environment*) doivent donc être couplés au serveur de cédérom afin que les programmes qu'ils ont permis aux étudiants d'écrire ou de tester puissent être notés. Nos expériences en C et en Scheme nous ont montré que l'on pouvait normaliser ces relations et ainsi les étendre à toutes sortes d'autres langages.

Les adjonctions visibles à apporter à l'IDE sont simples : *(i)* Un bouton ou une entrée dans un menu est ajouté à l'IDE pour sélectionner un exercice, afficher l'énoncé, positionner les bibliothèques nécessaires pour sa compilation ou exécution. *(ii)* Un bouton ou une entrée dans un menu est ajouté à l'IDE pour envoyer le programme de la fenêtre d'édition au serveur à des fins de notation automatique. Le résultat est une note et une série d'annotations en relation avec le texte du programme.

Les échanges sont implantés comme des services Web, ils sont conditionnés sous forme de documents XML valides vis-à-vis de grammaires normalisées.

- Sélectionner un exercice depuis l'IDE invoque le serveur de cédérom qui, au terme d'un dialogue plus ou moins complexe, détermine l'exercice approprié et renvoie ces informations à l'IDE. Les informations déterminent l'identifiant de l'exercice, le langage à utiliser (qui déterminera le notateur), l'URL de l'énoncé, le squelette de programme à compléter.
- Envoyer un programme pour notation indique le langage dans lequel est écrit celui-ci, son texte, la question de l'exercice auquel il est sensé répondre. Le retour est une note et une suite d'annotations en relation avec le texte du programme que l'IDE pourra rendre en illuminant les zones annotées.

Des IDE génériques, multi-langages et portables tels qu'Eclipse existent aujourd'hui. Un greffon pour Eclipse afin de le coupler à notre serveur de cédérom est en cours de développement.

L'intérêt de cette interface sous forme de services Web est de complètement découpler les aspects d'interface dans les IDE des aspects liés à la notation des programmes produits. Il est également possible de greffer ou de mettre à jour, relativement simplement, de nouveaux notateurs pour d'autres langages de programmation.

3 Déploiement

Souvent négligé, le déploiement est un problème d'une grande complexité. Plusieurs déploiements sont possibles :

- **sur la machine personnelle de l'étudiant**. L'installation passe par l'acquisition d'un espace sur le disque dur où seront stockés l'état du serveur (et sa base de données de traces), les données de mises à jour du cédérom, les clés de chiffrement ou déchiffrement, etc.
  Il est possible de copier le cédérom sur disque dur afin d'améliorer la vitesse d'accès. Les mises à jour sont alors faites directement dans cette copie. Cela permet aussi à certains étudiants de réindexer le cédérom afin de trouver rapidement les pages qui contiennent tel ou tel mot.
- **sur le portable personnel de l'étudiant**. 25% des étudiants de licence ont un portable comme seconde machine personnelle. Un mécanisme de synchronisation entre ces deux machines devrait alors être assuré afin de pouvoir travailler alternativement sur l'une ou l'autre. Ce mécanisme est à l'étude.
- **sur les ordinateurs de travaux pratiques de l'université**. Sur ces machines, les étudiants travaillent en binôme, le binôme est authentifié par le système d'exploitation (ici Linux). Le serveur de cédérom est automatiquement lancé sauf que le contenu du cédérom (à jour) est distant sur quelque serveur de l'université. Les étudiants sont invités au premier TP à s'enregistrer en binôme afin que leurs traces soient correctement et automatiquement acheminées. Les droits d'accès aux pages que les membres du binôme gagnent en TP peuvent leur être transmis sur leur machine personnelle.
- **sur les ordinateurs en libre service de l'université**. Sur ces machines, aucune donnée propre à un étudiant ne doit subsister après son passage. Il lance le serveur de cédérom et s'authentifie auprès de lui, il retrouve alors son état tel que maintenu par le serveur de l'université et ses droits associés. Le contenu du cédérom est local ou distant mais à jour.

L'élément central de ces déploiements est le serveur de l'université. Ce dernier contient la copie à jour du cédérom ainsi que la suite des différences entre la version courante et les versions précédentes de manière à assurer sa part du service de mise à jour. Il consolide les traces reçues des étudiants et, pour chacun, entretient son état à jour, état que l'étudiant pourra utiliser pour réinstaller sa machine personnelle ou temporairement sur une machine en libre service. Ce dernier aspect couvre le besoin de nomadisme.

4 Conclusion

Cet article a présenté les grandes fonctionnalités attendues d'un serveur de cédérom transformant un support passif en un système interactif, extensible, robuste permettant d'établir un suivi pédagogique étroit quoique asynchrone. Les fonctionnalités discutées ont été expérimentées progressivement ces quatre dernières années sur des cédéroms distribués aux étudiants de DEUG MIAS et de licence d'informatique de l'UPMC.

Certaines fonctionnalités répondant à de nouveaux besoins, notamment d'interrogations écrites à distance, ont également été introduites mais n'ont pas encore subi l'épreuve des étudiants.

L'état de nos expérimentations nous a ainsi conduit à proposer une architecture, un protocole de coopération entre serveur de cédérom et serveur de l'université, une standardisation du protocole de notation entre IDE et serveur de cédérom. Ces propositions ne couvrent que les enseignements d'informatique où l'usage d'un langage de programmation est prédominant en matière de travaux pratiques. Ces enseignements ne se limitent pas aux enseignements de programmation avec tel ou tel langage mais couvre les enseignements où l'emploi de langages est important. Ces enseignements représentent environ la moitié des enseignements des trois années du cycle L d'informatique.

Des informations additionnelles sur ces cédéroms sont disponibles en

`http://www.infop6.jussieu.fr/cederoms/`

Un grand merci à tous mes collègues et notamment Anne Brygoo, qui se sont attachés à me faire écrire précisément ce que recouvrait cette notion de serveur de cédérom.

Références

[BDM+02] Anne Brygoo, Titou Durand, Pascal Manoury, Christian Queinnec, and Michèle Soria. Un cédérom pour scheme — chacun son entraineur, un entraineur pour tous. In *TICE 2002 – Technologies de l'Information et de la Communication dans les Enseignements d'Ingénieurs et dans l'industrie*, pages 223–231, Lyon (France), November 2002. Institut National des Sciences Appliquées de Lyon.

[Dow] Tobias Downer. Mckoi : An open source java sql database system. Diehl & Associates, Inc. http ://mckoi.com/database/.

[QC02] Christian Queinnec and Emmanuel Chailloux. Une expérience de notation en masse. In *TICE 2002 – Technologies de l'Information et de la Communication dans les Enseignements d'Ingénieurs et dans l'industrie – Conférences ateliers*, pages 403–404, Lyon (France), November 2002. Institut National des Sciences Appliquées de Lyon. version complète disponible en `http ://www.infop6.jussieu.fr/licence/2001/cct/cfsreport.ps.gz`.

[QG01a] Christian Queinnec and Hélène Giroire. Pages dynamiques composées par filtrage une mise en œuvre sur cédérom. In Cyrille Desmoulins, Monique Grandbastien, and Jean-Marc Labat, editors, *EIAO'2001 – Environnements Interactifs d'Apprentissage avec Ordinateur*, volume 8 of *Sciences et Techniques Éducatives*, pages 190–191, Cité des Sciences de La Villette, Paris (France), May 2001. Hermès Science Publications. version longue en [QG01b].

[QG01b] Christian Queinnec and Hélène Giroire. Pages dynamiques composées par filtrage une mise en œuvre sur cédérom. Technical Report 6, LIP6, May 2001.

[Que00a] Christian Queinnec. Enseignement du langage C à l'aide d'un cédérom et d'un site – Architecture logicielle. In *Colloque international – Technologie de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie – TICE 2000*, pages 93–102, Troyes (France), October 2000. CNED.

[Que00b] Christian Queinnec. The influence of browsers on evaluators or, continuations to program web servers. In *ICFP '2000 – International Conference on Functional Programming*, pages 23–33, Montréal (Canada), September 2000.

[Que02] Christian Queinnec. A library for quizzes. In Olin Shivers, editor, *Scheme 2002 — Proceedings of the Third Workshop on Scheme and Functional Programming*, volume Georgia Tech, Technical Report GIT-CC-02-48, pages 1–7, Pittsburgh (Pennsylvania, USA), October 2002.

[The] The Apache Jakarta Project. Tomcat. http ://jakarta.apache.org/.