L'apport du logiciel libre et gratuit dans l'enseignement supérieur

< Manuel Siabato >

ESAV École Supérieur d'AudioVisuel
Université Toulouse II Le Mirail et Université Montpellier III Paul Valéry
msiabato@gmail.com

DOI:10.3166/RIN.3.321-340

< RÉSUMÉ >

Le logiciel libre et gratuit a un rôle important à jouer dans l'enseignement supérieur, bien qu'il soit peut-être nécessaire de remettre en question certaines positions chez les enseignants pour mieux apprécier ses bénéfices. Pour commencer, l'enseignant ne devrait plus présenter les outils propriétaires comme les seules solutions viables. Les applications libres et gratuites ont beaucoup évolué depuis leur apparition et peuvent aujourd'hui rivaliser avec les applications propriétaires équivalentes. De plus, le métier d'enseignant devrait être avant tout de transmettre une connaissance et un concept, et non pas l'utilisation ou la maîtrise d'un outil. Les bénéfices d'enseigner les logiciels libres sont nombreux. Sans vouloir établir une liste, il est important de citer ceux qui semblent les plus intéressants. Stimuler la curiosité de l'étudiant, l'inciter à la participation, au partage et à la collaboration et tout particulièrement véhiculer la notion de concept et non pas seulement celle d'outil tout au long de son apprentissage.

< ABSTRACT >

Free Software has an important role in higher education, although it may be necessary to question some opinions to better appreciate its benefits. In first place, teachers should not submit proprietary tools as the only viable solution to their students. Free software has significantly evolved since its beginnings and can now compete with equivalent proprietary applications. Secondly, teaching should be primarily to transmit knowledge and concepts and not just to use a tool. Using free software for teaching has many benefits. Without making a list it is important to include those that seem most interesting . Stimulate students curiosity, encourage participation, sharing and

collaboration, and particularly transmit the notion of concept and not just tool use throughout their learning.

< MOTS-CLÉS >

Logiciel libre, enseignement supérieur.

< KEYWORDS >

Free software, higer education.

1. Introduction

Nombreuses sont les raisons qui ont éloigné le logiciel libre de l'enseignement supérieur. Sans vouloir rentrer dans trop d'explications, car le but de ce texte n'est pas d'expliquer les raisons de cette absence mais plutôt les bienfaits de leur utilisation, il faut dire que tout commence par l'apparition du concept de logiciel propriétaire, soutenu largement par Bill Gates et l'entreprise dont il est cofondateur, Microsoft[1]. Ensuite les problématiques deviendront propres aux systèmes universitaires de chaque pays et plus concrètement aux choix de chaque enseignant. Pour mieux comprendre l'esprit du logiciel libre et les enjeux liés à son utilisation, il est important de citer les libertés que doit proposer ce type de logiciel aux utilisateurs. Richard Stallman, fondateur de la Fondation pour le logiciel libre (*Free Software Foundation, FSF)* et du concept même de logiciel libre, explique de manière très claire ces libertés[2].

> *Un programme est un logiciel libre si vous, en tant qu'utilisateur de ce programme, avez les quatre libertés essentielles :*
>
> *1- la liberté d'exécuter le programme, pour tous les usages (liberté 0) ;*
>
> *2- la liberté d'étudier le fonctionnement du programme, et de le modifier pour qu'il effectue vos tâches informatiques comme vous le souhaitez (liberté 1) ; l'accès au code source est une condition nécessaire ;*

1. Voir « An open Letter to Hobbyists »
http://fr.wikipedia.org/wiki/An_Open_Letter_to_Hobbyists
2. https://www.gnu.org/philosophy/free-sw.fr.html

3- la liberté de redistribuer des copies, donc d'aider votre voisin (liberté 2) ;

4- la liberté de redistribuer des copies de vos versions modifiées (liberté 3) ; en faisant cela, vous donnez à toute la communauté une possibilité de profiter de vos changements ; l'accès au code source est une condition nécessaire.

Quant à la place du logiciel libre et gratuit dans l'enseignement supérieur, on ne peut qu'espérer qu'un jour la tendance puisse être inversée. Car bien que la France avec l'Accord cadre[3] AFUL[4]–MENRT[5] de 1998 ou le Québec avec l'« Avis sur l'utilisation de logiciels libres et formats ouverts à l'Université de Montréal[6] » proposé par la FAÉCUM[7] en 2007 incitent à leur utilisation, aujourd'hui, un changement évident au sein de l'enseignement supérieur se fait toujours attendre. Les universités, les établissements de recherche ou les grandes écoles françaises ne prennent toujours pas de position officielle ferme par rapport à ce sujet (Romier et Archimbaud, 2009) et il faudra attendre encore quelques années avant que la loi 2013-660[8] relative à l'enseignement supérieur et à la recherche votée et adoptée en juillet en 2013 par le Parlement français porte ses fruits.

Il est pertinent de remarquer que les logiciels propriétaires sont souvent enseignés en cours comme le seul choix possible et non pas comme une possibilité parmi d'autres. L'enseignant, dans un souci de proposer aux étudiants les outils les plus utilisés dans le monde du travail, oublie qu'il y a toujours des alternatives libres et gratuites et que son rôle est avant tout de transmettre des concepts. Les entreprises, de leur côté, exigent très souvent des salariés de travailler exclusivement avec des logiciels fournis par elles pour éviter tout problème de compatibilité et de performance dans leur chaîne de production. Sans pouvoir exiger des entreprises de laisser ses employés choisir librement

3. https://aful.org/gdt/educ/accord-cadre-aful-menrt
4. Association Francophone des Utilisateurs de Linux et des Logiciels Libres
5. Ministère de l'Éducation Nationale, de la Recherche et de la Technologie
6. https://www.zotero.org/dmercier/items/itemKey/BV62XFKQ
7. Fédération des Associations Étudiantes du Campus de l'Université de Montréal
8. Article L123-4-1 : Le service public de l'enseignement supérieur met à disposition des usagers des services et des ressources pédagogies numériques. Les logiciels libres sont utilisés en priorité.

les outils qu'ils utilisent, comment l'enseignant peut-il briser ce « cercle vicieux » qui l'incite à présenter les logiciels propriétaires aux étudiants comme le seul choix possible pour s'insérer facilement dans le monde du travail ? Pourquoi l'enseignant devrait-il proposer des outils libres et gratuits aux étudiants ? Est-ce que ces outils vont inciter les étudiants à l'innovation tout en éveillant leur créativité ?

2. L'éthique du design interactif

Le rôle principal de l'enseignement supérieur devrait rester celui de transmettre les connaissances d'un métier par la théorie et la pratique, tout en proposant aux étudiants des outils qui rendent plus facile leur insertion au monde du travail. Les stages en entreprise et les simulations de projets personnels appliqués sont une première étape vers l'embauche. Mais la pédagogie active dans l'enseignement supérieur ne doit pas se limiter à l'étude de cas, à la réalisation de projets ou à la résolution de problèmes et, comme l'explique Denis Lemaître (2007), il s'agit aussi d'un « passage de valeurs "modernes" à des valeurs "postmodernes" », c'est-à-dire d'une adaptation à la réalité de notre quotidien. En quelques mots, ce qui devrait aussi intéresser l'enseignant est de transmettre une déontologie par rapport au métier qu'il enseigne. Bien que le terme *déontologie* ne soit pas le plus adapté pour de nombreux domaines, on peut tout de même souhaiter la transmission d'une certaine éthique et/ou code moral dans l'enseignement.

En partant de ce raisonnement, il n'est pas déplacé de se demander quelle serait l'éthique du design interactif. Quel comportement le designer doit-il suivre au long de son parcours professionnel pour rester fidèle aux principes de sa profession ? Comment un enseignant pourrait-il transmettre cette hypothétique déontologie professionnelle du design interactif ?

2.1. L'évolution d'un métier

Le design industriel illustre très bien les changements profonds des métiers du design dans la postmodernité. Aujourd'hui, contrairement

aux débuts de la production industrielle, il existe une forte tendance à proposer des produits qui soient en accord avec la protection de l'environnement. Dans un passé récent, ce qui comptait était l'innovation par de nouvelles formes, de nouveaux matériaux et de nouvelles technologies en se souciant peu des concepts comme le recyclage, l'économie de matière première ou les processus de production. En proposant des matériaux biodégradables ou issus du recyclage, des produits économiques en matières premières ou des processus de production moins polluants, le designer industriel peut s'approcher d'une éthique professionnelle dans son domaine. Mais n'oublions pas l'objectif du design industriel : résoudre des problématiques liées à l'utilisation confortable et agréable d'un produit créant un besoin chez le consommateur et, en conséquence, conduisant à la vente par des arguments sur l'esthétique et le confort. Par rapport au respect de l'environnement, l'éthique professionnelle du designer n'est pas appliquée à la lettre. La concurrence et la course au profit du monde actuel font que ce souhait utopiste n'est pas une priorité. En étant réaliste, la possibilité d'appliquer une telle éthique échappe à la profession quand il s'agit de convaincre un patron soucieux d'augmenter ses marges et de rendre des comptes aux actionnaires. L'éthique du design est alors contrainte par une exigence de production industrielle.

Quant au design interactif, les processus de production ne peuvent pas être envisagés par le designer lui-même. Cependant, si on veut imaginer quelle serait une éthique professionnelle, on peut suggérer que le designer se devrait de respecter les droits d'auteur, la propriété intellectuelle, tout comme de proposer des applications dont le but est d'accomplir correctement la tâche qui a motivé leur création. Par rapport aux droits d'auteur et à la propriété intellectuelle, le designer peut proposer des applications qui ne sont pas des copies conformes de ce qui existe déjà. Il serait souhaitable que l'innovation et la créativité soient la base de toute création en évitant toute appropriation, involontaire ou pas, d'une partie ou de l'intégralité des applications qu'il propose. Il ne s'agit pas de se fermer au monde en se privant de s'inspirer de ce qui existe déjà et de ce qui fonctionne, mais de respecter les droits de ceux qui ont développé les idées et les concepts en amont. Par rapport à ce qu'il développe, il est censé proposer des applications

simples et faciles d'accès, où l'ergonomie de l'interface permette un accès rapide à toutes les options de l'application. Il ne s'agit pas seulement d'une question de simplification de l'interface dans un but esthétique, mais d'éviter que l'utilisateur gaspille un temps de connexion qui pourrait lui être facturé. Le designer devrait proposer des applications qui ne poussent pas à l'achat systématique et compulsif pour améliorer un avatar ou les options de base du produit acheté. Bien qu'il s'agit d'une manière de produire de l'argent avec l'application, ce qui est souvent le but premier de créer une application, le designer peut aussi proposer d'autres moyens d'acquérir ces améliorations, comme des tâches ou des quêtes à accomplir. Par rapport aux applications vastes nécessitant de très grands volumes de travail pour les créer, il peut éviter de sous-traiter son travail à des prix qui frôlent l'exploitation. Dans un avenir proche ou lointain, on pourra peut-être se poser des questions quant aux choix stratégiques liés à la consommation d'énergie, soit par rapport à la vitesse de traitement des données ou selon le nombre de choix offerts par l'interface. Dans ce même avenir, le designer interactif devra sûrement se questionner par rapport aux limites d'intrusion de l'interface dans le champ visuel de l'utilisateur et jusqu'où la machine pourra prendre des décisions à la place du cerveau. Par exemple pour les applications d'assistance aux tâches complexes, comme la conduite automatisée de véhicules terrestres.

Envisager une éthique professionnelle pour le design interactif nécessite une réflexion approfondie et des avis provenant de multiples disciplines. Il faut avant tout se demander quel est son but et quelles sont ses limites. Car malgré la difficulté pour définir ces paramètres, on peut peut-être s'appuyer sur la base que le design interactif gravite autour de l'intangible et du concept, ce qui rend d'autant plus important son regard envers les droits d'auteur et la propriété intellectuelle.

2.2. Assumer des choix

Pour revenir au métier d'enseignant et aux priorités qui guident sa pratique d'enseignement, il semble pertinent de proposer une réflexion autour de ses habitudes et de ce qu'il propose aux étudiants. Bien que ce texte se positionne clairement en faveur des logiciels libres et gratuits, l'intention n'est pas de faire un chantage moral autour d'un système de

valeurs mais de proposer des chemins alternatifs tout en insistant sur l'absence de ces chemins au sein de nombreuses universités. Pour être plus direct, on doit se demander si l'enseignant ne devient pas un attaché commercial des grandes compagnies de logiciels. En proposant exclusivement des logiciels propriétaires dans ses cours, en ventant les dernières fonctionnalités de ces produits, l'enseignant semble oublier que son rôle premier est de transmettre des savoirs et des concepts liés au métier qu'il enseigne. Peut-être s'agit-il d'un effort supplémentaire pour l'enseignant que d'apprendre un nouveau logiciel pour l'enseigner ou de réinstaller tout le parc informatique d'une salle de cours, mais – si on est en accord avec cette affirmation – n'est-ce pas cela ce qui arrive à chaque nouvelle mise à jour du logiciel qu'il utilise couramment ? Si l'enseignant est le seul responsable du contenu proposé dans ses cours, ne serait-il pas correct alors qu'il remette constamment en question ses méthodes et ses outils de travail dans un monde qui bouge à la vitesse de la technologie ?

Les enseignants ont souvent la liberté dans le choix des logiciels à enseigner. Après avoir cité quelques-unes des raisons qui expliquent le choix récurent du logiciel propriétaire dans les salles de cours, comme l'utilisation massive dans le monde du travail, il semble judicieux de s'intéresser aux raisons qui pourraient faire changer d'avis un enseignant par rapport à l'utilisation du logiciel libre et gratuit dans ses cours.

3. Le logiciel libre comme défi pour l'étudiant

Dans cette partie, on souhaite développer l'idée que le logiciel libre et gratuit permet une approche plus riche de la connaissance que les logiciels propriétaires. L'idée n'est pas de démontrer que le logiciel libre est meilleur que le logiciel propriétaire, car il n'existe pas de comparaison possible entre les deux.

3.1. Stimuler la curiosité

Prenons l'exemple d'une application propriétaire qui crée les gabarits et les comportements des pages web. Il faut savoir que

l'affichage homogène sur l'ensemble des navigateurs des différents systèmes d'exploitation ou des téléphones portables *smartphones* reste l'une des plus grandes problématiques en matière de conception web. Bien qu'il existe des normes communes, comme le HTML4 et sa révision majeure le HTML5 (dont la spécification devrait être finalisée en 2014[9]), tous les navigateurs ne liront pas de la même manière le même code et certains pourront même interpréter incorrectement le gabarit. Quand l'étudiant utilise un logiciel propriétaire pour créer un bouton dans son projet d'interface, il doit juste cliquer sur l'icône et positionner le bouton en question à l'endroit souhaité. Si l'étudiant envisage que ce bouton enregistre des informations sur le nombre de clics ou d'autres renseignements sur l'activité de la page, il n'a qu'à cocher des cases sur les options de l'outil, si elles lui sont proposées. Il ne sait pas ce qui se passe quant à la programmation du site et ignore comment le logiciel va rédiger le code propre au bouton de son choix. On peut dire que la question posée est : jusqu'où doit aller l'enseignement de la technique dans les filières du design ? Doit-on former des étudiants de design interactif aux langages de balisage et/ou aux langages de programmation ? Si on envisage le même questionnement depuis un autre point de vue, le logiciel libre permet à l'étudiant d'explorer le code à sa source pour tenter de comprendre, voire de modifier ce qui est écrit, alors que l'entreprise qui a conçu le logiciel propriétaire n'a aucun intérêt à dévoiler ses processus. Il ne s'agit pas de former un programmeur, ou un expert en langages informatiques mais de laisser le choix à l'utilisateur d'aller aussi loin qu'il le souhaite dans la compréhension de l'outil.

Pour un exemple concret, comparons un logiciel d'édition de sites web comme Dreamweaver et un possible équivalent représenté par un CMS[10] comme Wordpress[11]. Avec Dreamweaver, il sera toujours possible de relire comment le logiciel traduit la commande de création du bouton, mais l'utilisateur ne pourra pas lui indiquer ce qu'il souhaite comme code pour le bouton. Quand le travail se fait avec des logiciels

9. http://fr.wikipedia.org/wiki/Hypertext_Markup_Language
10. Content Management System
11. Bien qu'il s'agisse de logiciels qui font des sites Internet de manière différente, on a souhaité comparer deux solutions ayant relativement la même popularité chez les utilisateurs.

libres comme Wordpress, l'utilisateur a toujours le choix de l'extension pour faire ce qui l'intéresse. Pour mieux comprendre le rôle d'une extension dans un logiciel, il faut imaginer que le logiciel tel qu'il est livré par l'éditeur, ne propose pas toutes les options dont pourrait avoir besoin un utilisateur. Il ne s'agit pas de voir le logiciel livré comme incomplet, mais plutôt comme susceptible d'être amélioré grâce à l'extension. L'éditeur ne peut pas envisager les besoins propres à chaque utilisateur mais peut autoriser l'ajout d'une amélioration par le biais d'une extension. Généralement l'extension est fournie par un éditeur tiers qui la propose directement aux utilisateurs. Quand l'extension est demandée par un grand nombre d'utilisateurs, l'éditeur peut prévoir de l'inclure dans la mise à jour du logiciel. Pour revenir à Wordpress, on peut installer plusieurs extensions qui font la même chose de manière différente, ainsi que changer le code de chaque extension selon les besoins puisque tout est ouvert. Bien que les extensions payantes chez Wordpress soient nombreuses, rien ne bloque l'accès au code et donc à la *customisation* de l'extension. Quant à Dreamweaver, de même que pour l'ensemble des logiciels propriétaires, l'extension installée ne donne pas accès à son code limitant toute *customisation*.

On trouve aussi des extensions gratuites dans les logiciels propriétaires mais elles proposent très souvent le service de manière temporaire à titre d'avant-goût de l'extension en usage illimité. Outre la frustration que peut produire le fait d'avoir une extension qui ne marche plus du jour au lendemain, les formulaires et procédures à suivre pour réussir l'installation sont nombreux et découragent souvent l'étudiant à continuer. Dans le cas de Dreamweaver, il est nécessaire d'installer une application supplémentaire pour gérer ces extensions, ce qui peut compliquer davantage l'installation[12]. A contrario, sur Wordpress, l'interface propose une rubrique dédiée aux extensions qui, par le biais d'une recherche de mots clés, permet de rechercher pour ensuite installer l'extension qui est la mieux adaptée[13]. Avec une grande facilité d'installation, l'étudiant pourra tester très vite beaucoup plus d'extensions réalisant une même tâche et faire un choix mieux adapté à

12. http://www.dummies.com/how-to/content/installing-new-dreamweaver-extensions.html
13. http://www.wordpress-fr.net/faq/comment-installe-t-on-un-plugin-dans-wordpress-2/

ses besoins. Pour résumer en quelques mots, le logiciel libre offre beaucoup plus de possibilités pour tester un même processus, avec des procédures simples et pérennes. Pour avoir un accès équivalent sur un logiciel propriétaire, il est nécessaire d'acheter un SDK[14] ou Kit de développement de logiciel (quand il en existe un). Le prix de la licence SDK reste généralement inaccessible pour un enseignant, pour un étudiant ou un particulier. Par rapport aux extensions, bien qu'on puisse installer plusieurs versions payantes avec les mêmes fonctionnalités sur le logiciel propriétaire, on ne peut pas vraiment savoir comment elles fonctionnent puisqu'elles sont fermées.

3.2. Inciter à la participation

Bien que les logiciels soient proposés au public après avoir été testés et validés comme stables et fiables, aucun logiciel n'est parfait et ne présente pas de bogues[15] dans son fonctionnement. Même le logiciel le plus abouti est susceptible d'être amélioré, tant par l'ergonomie de l'interface que par l'ajout ou perfectionnement de ses fonctionnalités. On peut remarquer plusieurs choses par rapport à l'attitude qu'adopte un étudiant face à la trouvaille d'un bogue (ce qui est rare) ou à son souhait de voir modifiée ou améliorée l'interface ou une fonctionnalité. Souvent la première réaction de l'étudiant face à ces problématiques est de qualifier le logiciel de « nul » et de perdre son intérêt, et ce, indépendamment qu'il s'agisse d'un logiciel libre ou propriétaire. Une fois que l'intérêt de l'étudiant est perdu, il est très difficile qu'il se remette au travail et qu'il s'intéresse, avant tout, aux concepts que l'enseignant veut transmettre. Rien ne garantit qu'il trouvera facilement la solution à son problème, ce qui peut rapidement le décourager. Souvent l'enseignant n'a pas les réponses à tous les problèmes liés au fonctionnement et/ou à l'interface du logiciel et il se voit obligé de proposer à l'étudiant de trouver lui-même des solutions sur Internet. Nombreux sont les forums spécialisés qui proposent des tutoriels et qui donnent des conseils pour résoudre des bogues, mais il faut savoir que ces sites existent grâce à la générosité des utilisateurs, car l'éditeur du logiciel propriétaire, malgré qu'il propose souvent des aides à l'intérieur

14. Software Development Kit
15. De l'anglais « bug » http://fr.wikipedia.org/wiki/Bug_%28informatique%29

même du logiciel, aura toujours du mal à prévoir le besoin spécifique de chaque utilisateur et leur service après-vente ne s'engage malheureusement pas à répondre à toutes les questions et encore moins à résoudre tous les problèmes liés aux améliorations du produit[16]. On doit saluer la contribution de ces bénévoles passionnés, car, bien qu'il s'agisse de logiciels propriétaires, leur logique répond souvent aux mêmes principes que celle des logiciels libres, c'est-à-dire l'accès et le partage de la connaissance. Si jamais une solution logicielle est demandée par rapport à une faille de programmation ou à une amélioration souhaitée, il faut se résigner à attendre la prochaine mise à jour du produit ou tout simplement attendre que les éditeurs s'intéressent à la problématique suggérée.

La spécificité du logiciel gratuit par rapport au logiciel propriétaire est l'existence d'une communauté de gens bénévoles, programmeurs ou pas, disposée à proposer des solutions, voire à créer des solutions pour de nombreuses problématiques. Quand un individu signale une faille du système logiciel ou propose une amélioration sur un logiciel libre, sa demande est généralement prise en compte dans la limite de sa pertinence, ce qui peut inciter un étudiant à participer à une communauté qui cherche à produire le meilleur logiciel et non pas à réaliser du profit. En s'intégrant à cette communauté, l'étudiant éveille en lui un esprit critique tout comme le sentiment de respect pour ses propres inquiétudes et questionnements. Un étudiant qui n'a pas peur de poser des questions est un étudiant qui deviendra plus sûr de lui-même et qui ne se contentera pas du silence corporatif comme réponse. Un étudiant en design interactif devrait être capable de trouver les solutions ou du moins de proposer des solutions possibles à sa problématique, sans besoin d'attendre qu'un éditeur s'intéresse à ses soucis.

3.3. Inciter au partage et à la collaboration

Le logiciel, par sa nature d'objet numérique, permet une grande facilité de reproduction et de redistribution. Ces facilités, vont à

16. http://www.njuris.com/Imp_Breve.aspx?IDBreve=233

l'encontre des lois de la propriété intellectuelle et poussent à la schizophrénie quand il s'agit du partage d'outils logiciels. Pour le dire autrement, les étudiants désirant apprendre des nouveaux outils propriétaires se voient souvent confrontés au dilemme du piratage informatique pour suivre correctement leur formation. Pour remédier à cette pratique courante dans tous les domaines, car nombreux sont les travailleurs indépendants ou petites entreprises qui utilisent des logiciels sans payer de licences, d'intensives campagnes de communication visent à dénoncer ce « fléau » et à culpabiliser les pirates fraudeurs en les comparant à des voleurs.

Stratégie de communication ou simple laisser-faire, la position des éditeurs n'est pas toujours très nette[17]. La schizophrénie mentionnée apparaît quand les enseignants se voient obligés de conseiller passivement aux étudiants de pirater leurs outils de travail afin de travailler chez eux[18] alors qu'ils les éduquent à faire respecter ces mêmes droits quand il s'agit de leurs créations personnelles. Un étudiant justifie le piratage informatique quand il s'agit de s'approprier un outil qui lui permettra de créer, voire de s'intégrer au monde du travail[19], mais il ne sera jamais d'accord quant à la cession de ses droits en matière de création originale. D'une part, il est poussé au piratage et, d'autre part, on lui apprend à faire respecter ses droits. Une double morale qui a sans doute des effets pervers dans la mentalité des étudiants. Il faut tout de même remarquer que certaines pratiques se banalisent[20], comme le plagiat ou l'omission des sources et qu'on ne peut pas conclure rapidement qu'il s'agit seulement de la facilité du copier-coller[21] [22].

17. http://www.uzine.net/article94.html
18. http://www.framasoft.net/article1648.html
19. http://www.framablog.org/index.php/post/2012/12/02/J-ai-prirat%C3%A9-Windows-et-ma-vie-a-chang%C3%A9
20. http://www.ladocumentationfrancaise.fr/var/storage/rapports-publics/124000295/0000.pdf
21. http://archeologie-copier-coller.com/?p=11563
22. http://www.institut-numerique.org/le-plagiat-lectronique-au-niveau-de-lenseignement-superieur-universitaire-public-etat-des-lieux-propositions-de-pistes-de-prvention-et-dtection-cas-de-luniversit-sultan-moulay-slimane-de-beni-mellal-52ce62856e9be

Du côté du logiciel libre, il n'existe pas de problème de gestion du piratage ou des licences piratées. Bien au contraire, ce qui intéresse le créateur du logiciel non propriétaire est justement qu'il soit copié et distribué le plus largement possible. De ce point de vue, le partage du logiciel ne représente pas une atteinte à la loi sur la propriété intellectuelle et permet à l'étudiant de garder une cohérence entre ce qu'on lui apprend sur ses droits et les pratiques qu'il a en relation aux droits d'autrui. Les étudiants qui utilisent des logiciels libres peuvent partager plus facilement leurs connaissances dans la mesure où ils ne sont pas bloqués par la manipulation du piratage, ni par les formats propriétaires. Quand un étudiant souhaite montrer à des non-initiés les logiciels propriétaires qu'il utilise ou, d'une manière plus générale, s'il souhaite partager les fichiers de son travail avec son entourage, il se voit obligé d'inciter au piratage ou de distribuer des versions piratées des logiciels, moyen le plus simple de permettre à ses copains de travailler sur son projet.

Il peut sembler utopiste de gagner sa vie en développant du code pour la communauté du logiciel libre, mais il ne faut pas croire que ce choix s'oppose à la génération de richesse. Il faut savoir que le logiciel libre peut aussi être vendu, que ce n'est pas mal de le faire et que la formation et la migration vers des solutions logicielles libres, pour les entreprises et le secteur privé, représentent un potentiel économique assez important. Sur le site Internet de la FSF, l'article intitulé « Vendre des logiciels libres »[23] explique très bien comment la production de richesse est compatible avec les logiciels libres :

> *[...] Les logiciels libres sont parfois distribués gratuitement, et parfois pour un prix conséquent. Un même programme est souvent disponible sous ces deux formes à partir de sources différentes. Le programme est libre en dépit de son prix, car les utilisateurs ont toute liberté dans son utilisation [...]*
>
> *[...] L'expression « vendre des logiciels » peut aussi induire en erreur*
>
> Stricto sensu, *« Vendre » signifie échanger des biens contre de l'argent. Vendre une copie d'un logiciel libre est légal, et nous encourageons cette pratique.*

23. http://www.gnu.org/philosophy/selling.fr.html

> *Cependant, quand les gens envisagent de « vendre des logiciels », ils imaginent habituellement le faire de la même manière que la plupart des entreprises : rendre le logiciel privateur plutôt que libre.*
>
> *Alors, à moins que vous ne soyez prêts à faire des distinctions précises, comme le fait cet article, nous vous suggérons d'éviter le terme « vendre des logiciels » et de choisir un autre vocabulaire à la place. Par exemple, vous pourriez dire « distribuer des logiciels libres contre rémunération » ce qui lève toute ambiguïté ». [...]*

Le logiciel libre vendu garde alors toutes les libertés propres au logiciel libre, et ce n'est pas parce qu'il y a eu un échange monétaire lié à la distribution du logiciel que l'utilisateur perdra les droits propres à la licence. Pour mieux comprendre comment un logiciel libre peut être vendu et générer de la richesse, prenons l'exemple des distributions GNU/Linux[24] OpenSUSE et SUSE Linux Entreprise[25]. Bien que les deux distributions soient proposées par le même éditeur (Novell), et que toutes les deux proposent de manière générale les mêmes logiciels, la version entreprise est payante. Cette version propose du service après-vente, un support plus long, une espérance de vie plus longue et elle cible spécifiquement le monde de l'entreprise.

3.4. Véhiculer la notion du concept dans la création

Pour revenir aux commentaires des étudiants sur les outils numériques, il est important de remarquer la confiance qu'ils ont dans la fiabilité et la performance des logiciels propriétaires, notamment par rapport aux produits de l'éditeur Adobe. Il s'agit évidemment de logiciels très performants et, pour certains, la référence absolue dans le domaine en tant que pionniers de l'industrie, comme c'est le cas pour Photoshop. Bien que l'utilisateur puisse glorifier l'interface, la performance ou les outils proposés par un logiciel, seule sa maîtrise de l'outil peut témoigner de ce qu'il dit. Pour expliquer autrement, l'utilisateur qui produit du mauvais travail avec un logiciel très performant[26], produit ce mauvais travail parce qu'il ne maîtrise pas les

24. http://fr.wikipedia.org/wiki/Distribution_linux
25. http://fr.wikipedia.org/wiki/SUSE
26. Voir http://www.psdisasters.com/

concepts de l'outil qu'il a choisi. Il est très important pour les enseignants de faire prendre conscience aux étudiants, que ce n'est pas le logiciel qui fait des choses exceptionnelles parce qu'il est bien conçu mais que c'est l'utilisateur derrière la machine, qui grâce à des concepts solides et clairs peut proposer des créations cohérentes, voire intéressantes. L'utilisateur qui maîtrise bien les concepts des outils qu'il utilise doit être en mesure d'utiliser n'importe quel logiciel pour parvenir à de bons résultats. Ce sont ces concepts que l'enseignant se doit de transmettre, s'il souhaite que ses étudiants puissent devenir autonomes et surtout pas dépendants d'un seul logiciel. L'enseignant ne devrait pas proposer un logiciel en cours pour ses qualités exceptionnelles, ou parce qu'il est convaincu que la dernière version est la meilleure du marché. La méthode pédagogique de l'enseignant ne devrait pas mettre en avant un logiciel par rapport à un autre, mais devrait proposer des choix, transmettre des concepts et partager la connaissance. Préparer un cours qui se limite à une seule option logicielle prend sûrement moins de temps mais peut produire chez l'étudiant le sentiment qu'il n'a rien à chercher de sa part, car tout lui serait donné, même l'outil qu'il va utiliser. C'est grâce aux concepts et non pas à la maîtrise d'un logiciel en particulier qu'on peut accomplir correctement une tâche. Manuel Castels (2001)[27] nous met en garde quant aux « défis de la société des réseaux » dans laquelle nous vivons aujourd'hui :

> *[…] Le troisième défi majeur, c'est l'éducation. Il faut parvenir à équiper chacun de nous – et d'abord chaque enfant – de la capacité à traiter l'information et à produire du savoir. Au sens le plus large, le plus fondamental, j'entends par éducation la capacité intellectuelle d'apprendre à apprendre tout au long de sa vie, à aller chercher des informations mises en mémoire sous forme numérique, à les recombiner, et à produire grâce à elles un savoir utile à l'objectif qu'on s'est fixé, quel qu'il soit. Cette définition simple remet totalement en cause le système d'éducation conçu à l'ère industrielle. Il n'est pas de restructuration plus essentielle que celle-là, et très rares sont les pays et les institutions qui la tentent vraiment. Parce qu'avant de commencer à changer de technologie, à reconstruire les écoles et recycler les maîtres, il faut une pédagogie*

27. Castels Manuel (2001). *La galaxie Internet*, Fayard, Paris, p. 336.

> *nouvelle, fondée sur l'interactivité, l'individualisation, qui développe une capacité autonome d'apprendre et de penser, tout en affermissant le caractère et en sécurisant la personnalité. Ce territoire-là n'est pas cartographié, il est* terra incognita*. [...]*

4. Conclusion

Pour conclure cet article, il est important de remarquer l'intérêt grandissant pour les logiciels libres dans l'enseignement supérieur. C'est peut-être la crise économique planétaire des dernières années qui a mis en valeur les qualités du logiciel non propriétaire. D'une part, les universités tout comme le monde du travail se rendent compte de l'impossibilité de suivre économiquement un renouvellement régulier et parfois injustifié du parc logiciel et matériel. D'autre part, le logiciel libre évolue constamment, ses producteurs sont guidés par le seul objectif de proposer le meilleur outil possible à l'opposé des promoteurs du logiciel propriétaire qui, par définition, cherchent en premier lieu le profit.

Le logiciel libre émerge dans un univers informatique contrôlé par l'appât du gain, une jungle de concurrence où les partisans de la cathédrale et du bazar s'affrontent (Raymond, 2001). Cette analogie présente le développement d'un logiciel propriétaire comme une cathédrale où l'importance du développeur dans la hiérarchie du projet définira l'évolution du code fermé, à l'opposé du bazar, propre aux logiciels libres où tous les participants peuvent contribuer sans distinction de leur statut puisqu'il n'y a pas de vraie hiérarchie[28]. Le logiciel libre n'est peut-être que l'évolution naturelle du monde informatique, le rejet frontal du brevet dans la transmission et le partage de la connaissance. On peut espérer que le logiciel libre reviendra un jour en force dans l'enseignement supérieur, comme aux débuts de l'informatique quand les chercheurs se partageaient le code pour faire marcher les imprimantes. Il ne faut pas oublier que c'est l'explication que donne Richard Stallman à sa motivation de proposer le concept du logiciel libre.

28. http://fr.wikipedia.org/wiki/La_Cath%C3%A9drale_et_le_Bazar

> *[…] Illustration de ce changement d'époque, l'anecdote de l'imprimante : celle du laboratoire tombait souvent en panne, mais les chercheurs avaient modifié son programme, grâce à l'accès au code source […] Un jour la société Xerox offre au laboratoire une imprimante laser. [...] Pas de bidouillage possible cette fois, faute de code source fourni avec l'appareil, et les chercheurs doivent se résoudre à patienter parfois une heure pour un document dont l'impression prendrait normalement dix minutes. [...]*
>
> *L'épisode de l'imprimante, comme les problèmes plus larges de logiciels devenus inutilisables au MIT, amènent Stallman à en tirer des conclusions : le logiciel propriétaire est à combattre, comme un scientifique qui garderait pour lui les résultats de ses recherches au lieu de les partager avec ses pairs pour les évaluer et les améliorer. Durant l'été 1983, le jeune chercheur projette de développer un système d'exploitation libre pour les ordinateurs. [...]* [29]

Peut-être que ce sera une prise de conscience de la part des enseignants quant à l'importance d'associer à leurs cours des outils non propriétaires, qui permettra au logiciel libre de jouer un rôle important dans l'enseignement supérieur. Les avantages d'un tel changement sont nombreux et seulement le temps permettra de mesurer l'étendue de ce choix. Pour l'instant, on peut espérer que les logiciels libres permettront aux étudiants de mieux s'adapter à un monde du travail de plus en plus compétitif, que grâce à eux ils auront de meilleures bases pour mieux comprendre des concepts appris en cours. À ce propos, Autodesk, une des plus grandes maisons d'édition d'outils numériques (3DSmax, Maya, Autocad) a récemment changé sa politique vis-à-vis des licences proposées aux enseignants et aux étudiants[30]. On peut désormais télécharger et installer des versions illimitées de leurs produits dans un cadre de production pédagogique. Une tentative de rester présents dans le marché de l'enseignement, mais cela ne résout pas le problème de la licence propriétaire des logiciels au moment où les étudiants deviennent professionnels.

---

29. Perline et Thierry Noisette (2004). *La bataille du logiciel libre*, *Dix clés pour comprendre*, La Découverte, Paris, p. 23-24.
30. http://www.autodesk.com/education/free-software

Cependant, les mentalités changent même chez les plus réticents qui commencent à voir le logiciel libre, non pas comme un problème, mais comme une solution possible au développement. C'est le cas pour Microsoft, qui explique dans l'introduction de son livre blanc « Microsoft et les logiciels Open Source[31] » de 2013 que :

> *L'opposition historique entre les logiciels Open Source (OSS) et les logiciels commerciaux (en source « propriétaire ») a évolué ver une approche nuancée, parfois complexe que certains avaient prévue et d'autres pas. [...] Le résultat qui a été plus difficile à prédire – et beaucoup plus difficile étant donné le débat souvent clivant –, est que nous vivons désormais dans un monde de « source mixte ». [...]*
>
> *Microsoft supporte l'Open source et le modèle Source Mixte à de multiples niveaux, reflétant ainsi l'engagement à soutenir les clients et développeurs. [...]*

Bien que les intentions de Microsoft semblent bienveillantes et que plus tard dans le texte il se présente comme un grand supporteur des projets Open Source, il ne faut pas oublier que Microsoft a acheté Novell par le biais du consortium Attachmate Consortium[32] en récupérant ainsi une partie de ses brevets[33] :

> *[...] L'entreprise participe également à des projets Open source, en suivant les normes et les licences des projets. L'an dernier, Microsoft a été cité par la fondation Linux comme l'une des 20 entreprises ayant contribué au noyau Linux[34]. Des commentateurs l'ont même qualifié de "principal contributeur" pour Linux[35] [...]*

D'un autre côté, l'article de Microsoft présente le *Cloud*[36] comme « [...] une force majeure dans le monde de l'informatique [...] » qui « [...] rend

31. download.microsoft.com/documents/France/openness/2013/Livre-blanc-Microsoft-logiciels-OpenSource.pdf
32. http://www.theinquirer.net/inquirer/news/1900065/microsoft-led-consortium-buys-novell-assets
33. http://linuxfr.org/news/novell-et-microsoft-main-dans-la-main
34. The Linux Foundation, Linux Kernel Development: How Fast it is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It at 10-11 (Mars 2012), http://go.linuxfoundation.org/who-writes-linux-2012.
35. Joab Jackson, Microsoft Counted as Key Linux Contributor - For Now, Anyway, InfoWorld (Apr. 3, 2012), http://www.infoworld.com/d/open-source-software/microsoft-counted-key-linux-contributor-now-anyway-190104.
36. http://fr.wikipedia.org/wiki/Cloud_computing

caduc le débat opposant logiciels OSS et logiciels commerciaux, puisqu'il masque aux utilisateurs finaux le code source sous-jacent et sa complexité. [...] ». Comme on peut le constater, Microsoft semble cacher son jeu derrière la bonne volonté et par rapport au *Cloud* la position de Richard Stallman ne s'est pas fait attendre[37] :

> *Le cloud computing est tout simplement un piège forçant plus de monde à verser dans la sécurité, à acheter des systèmes propriétaires qui leur coûteront de plus en plus cher au fil du temps […] L'une des raisons pour lesquelles vous ne devriez pas utiliser des applications web pour votre travail informatique est la perte de contrôle [...].*

Comme on peut constater, le débat entre logiciel libre et propriétaire est loin d'être fini, et l'avenir se jouera dans les nombreuses propositions d'un modèle qui s'essouffle sans pouvoir vraiment justifier le prix des licences auprès des usagers. Par rapport au logiciel libre dans l'enseignement supérieur, on peut espérer que son apport dans l'avenir ne va pas se limiter à une possibilité de plus parmi les outils proposés, mais qu'il apportera un vrai changement quant à la manière d'apprendre, d'enseigner et de partager la connaissance.

Bibliographie

Bauwens Michel (2011). Du design ouvert aux fabrications coopératives, *Libres savoirs. Les biens communs de la connaissance,* C&F éditions, Caen, p. 211-222.

Blondeau Olivier (2000). Genèse et subversion du capitalisme informationnel, LINUX et les logiciels libres : vers une nouvelle utopie concrète ?, *Libres enfants du savoir numérique*, Éditions de l'éclat, p. 171-195.

Grevet Patrice (2006/3). Une contradiction structurante dans la numérisation de l'enseignement supérieur*. Distances et savoirs,* vol. 4, n° 3, p. 333-364.

Guédon Jean-Claude (2011). Connaissance, réseaux et citoyenneté : pourquoi le libre accès ? *Libres savoirs. Les biens communs de la connaissance.* C&F éditions, Caen, p. 67-76.

37. http://www.generation-nt.com/stallman-cloud-computing-logiciel-proprietaire-actualite-163041.html

Le Crosnier Hervé (2011). Leçons d'émancipation : l'exemple du mouvement des logiciels libres. *Libres savoirs. Les biens communs de la connaissance,* C&F éditions, Caen, p. 175-191.

Lemaître Denis (2007). Le courant des « pédagogies actives » dans l'enseignement supérieur : une évolution postmoderne ?, *Recherches en Éducation n° 2*, Université de Nantes.

Levy Steven (2013). *L'éthique des hackers.* Globe, L'école des loisirs, Paris.

Moreau Didier (2007/2). L'éthique professionnelle des enseignants : Déontologie ou éthique appliquée de l'éducation ? *Les Sciences de L'éducation – Pour l'ère nouvelle,* vol. 40, p. 53-76.

Romier Geneviève, Archimbaud Jean-Luc (2009) La place des logiciels libres dans l'enseignement Supérieur et la Recherche, état des lieux à travers PLUME. Et que font les autres? *JRES 2009,* Nantes.

Raymond Eric (2001). *Cathedral and the Bazaar : Musings on Linux and Open Source by an accidental Revolutionnary.* O'Reilly Media, Beijing.