

Avis sur l'utilisation de logiciels libres et formats ouverts à l'Université de Montréal

Groupe d'utilisateurs Linux de l'Université de Montréal



22 août 2007

Cette création est mise à disposition selon le contrat « Paternité – Partage des conditions initiales à l'identique 2.5 Canada », disponible en ligne <http://creativecommons.org/licenses/by-sa/2.5/ca/> ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

1 Introduction

L'idée d'un ensemble complet de logiciels partagés par tous et écrits par une communauté de développeurs a pris naissance en 1985 avec la publication par Richard Stallman du *Manifeste GNU* [21]. Selon Stallman, chacun devrait avoir la possibilité de partager librement les programmes qu'il apprécie avec d'autres personnes, ce que la plupart des licences de logiciels commerciaux interdisent.

Stallman a ainsi entrepris de développer un système d'exploitation « libre », c'est-à-dire que tous doivent pouvoir l'utiliser, mais aussi étudier son fonctionnement, le modifier et le redistribuer : ce système est appelé GNU¹, qui signifie simplement *GNU's not Unix*. Exercer ces libertés nécessite notamment d'avoir accès au code source, c'est-à-dire les documents lisibles par l'être humain à partir desquels une version définitive et exécutable est produite. C'est pourquoi on entend aussi parler de code source ouvert (*open source*)². En deux décennies, l'idée a fait son chemin dans de nombreux milieux et régions du monde, de sorte qu'une grande variété de logiciels libres de qualité est aujourd'hui disponible. De plus, l'accessibilité aux non-spécialistes a été travaillée si bien qu'il est envisageable pour le grand public d'utiliser ces logiciels.

De nombreuses administrations ont déjà étudié la viabilité de l'adoption du logiciel libre, par exemple le gouvernement du Québec [19] et le gouvernement du Canada [4].

Un grand nombre d'universités se sont aussi démarquées dans leur utilisation de logiciels libres. Par exemple, l'Université d'Australie-Occidentale a adopté une politique [11] encourageant l'utilisation des logiciels libres dans tous les champs d'application, l'Université du Littoral — Côte d'Opale (France) offre un master en ingénierie du logiciel libre. Plus près de nous, l'Université Laval propose un cours « logiciels libres et société » et l'Université de Sherbrooke a développé, il y a quelques années une distribution GNU/Linux québécoise nommée ÉduLinux. Il est donc surprenant que l'Université de Montréal ne se soit pas encore intéressée à la question.

Heureusement, certains départements utilisent déjà les logiciels libres depuis plusieurs années, par exemple le département d'informatique et de recherche opérationnelle. La plupart des membres de la communauté universitaire ont sans doute déjà entendu parler de Firefox, un fureteur libre. Cependant, on peut rapidement s'apercevoir que les connaissances sur les logiciels libres sont minimales : il reste beaucoup d'information à distribuer et de formations à donner.

C'est ce constat qui a poussé l'association des étudiants et étudiantes aux études supérieures du département d'informatique et de recherche opérationnelle à faire adopter l'orientation :

Que la FAÉCUM développe son discours sur les logiciels libres et les formats ouverts.

Dans ce document, nous présenterons dans un premier temps les formats ouverts permettant à des logiciels libres de communiquer entre eux et à l'utilisateur de choisir son logiciel en fonction de ses besoins. Dans un second temps, nous étudierons les logiciels libres plus en profondeur en examinant leurs avantages et les processus rendant possible leur existence. Nous envisagerons ensuite diverses pistes pour « donner en retour » à la communauté du libre, afin que l'université contribue à son essor.

¹désigné par GNU/Linux lorsqu'utilisé avec Linux comme système sous-jacent

²*Open source* et logiciel libre sont des idées distinctes : il est tout à fait possible d'écrire du logiciel propriétaire et de distribuer tout ou partie de son code source aux clients, en restreignant les libertés d'actions sur ce code source.

2 Formats ouverts

2.1 Format

Un *format de données* est un ensemble de règles servant à représenter sous forme informatique, c'est-à-dire avec une suite de nombres, un certain type de données, qu'il s'agisse de pages de texte, d'images, de vidéos ou autres. Les données sont la plupart du temps stockées dans des fichiers dont l'extension, c'est-à-dire les caractères suivant le point dans le nom du fichier, indique le format. Pour cette raison, il arrive souvent de dénoter un format de fichier par son extension plutôt que son nom, par exemple `.html` pour une page web en HTML.

Posséder un format de données commun permet à des programmes distincts de s'échanger des données, en passant par des fichiers respectant ce format commun.

2.2 Qu'est-ce qu'un format ouvert ?

Un format *ouvert* se distingue par le fait que ses particularités sont entièrement documentées, disponibles au grand public, et sans limitation d'utilisation.

La liberté qu'apportent les formats ouverts ne touche pas que les programmeurs ; elle permet à plus de logiciels d'être compatibles entre eux, augmentant les possibilités de communication entre utilisateurs de logiciels différents. Utiliser des formats ouverts évite également qu'un utilisateur voie sa liberté de choix de logiciel réduite par des craintes d'incompatibilité entre un nouveau programme et ses fichiers existants.

Quiconque peut librement étudier le format, et bâtir des programmes entièrement conformes, étant donné que toute l'information nécessaire sur le format est disponible. Par exemple, une entreprise qui choisirait d'intégrer le format OASIS OpenDocument Text (`.odt`) à son logiciel de traitement de texte peut s'assurer de le supporter au meilleur de ses capacités. Ceci **remet le choix entre les mains du consommateur**.

2.3 Pourquoi utiliser les formats ouverts ?

L'usage des formats propriétaires (ou « fermés ») s'accompagne généralement d'une volonté, de la part des concepteurs de logiciels propriétaires, de se livrer à une « guerre de secrets » : il s'agit d'assurer que seuls leurs logiciels seront capables de relire les fichiers qu'ils ont créés, de manière à augmenter, pour le consommateur, le coût de transition vers un logiciel concurrent.

En plus d'empêcher la création d'une clientèle captive, les formats ouverts encouragent l'innovation de la part des développeurs, en axant la concurrence sur la qualité des programmes eux-mêmes plutôt que sur leur capacité de lire le format de données dominant.

2.3.1 Choix contre *vendor lock-in*

Lorsque des formats fermés sont employés, les utilisateurs sont obligés d'acheter les mêmes logiciels que ceux qu'utilisent leurs collègues de travail, leurs professeurs ou leur famille pour s'assurer d'une parfaite compatibilité. De plus, une entreprise qui déterminerait qu'une alternative logicielle convient mieux à ses besoins devrait faire face à la crainte d'être incapable de lire tous les documents fournis par d'autres entreprises, voire ses propres documents archivés.

D'un autre côté, si des formats ouverts sont utilisés, les utilisateurs peuvent communiquer sans difficulté avec des logiciels différents capables d'utiliser les mêmes formats. De cette façon, chacun choisira le logiciel qui est le plus **adapté à ses besoins**, en termes de coût, de fonctionnalités, etc.

Les formats de fichiers utilisés par la suite Microsoft Office illustrent bien une situation d'enfermement propriétaire. Les formats de traitement de texte, de chiffrier et de présentation `.doc`, `.xls` et `.ppt` sont propriétaires, ce qui empêche l'écriture de programmes compatibles.

En pratique, il existe des suites bureautiques compatibles, mais elles ont opéré par *rétro-ingénierie* [29], c'est-à-dire en essayant de « deviner » la structure du format par essais et erreurs. C'est hélas un procédé lent, coûteux, et qui ne peut assurer une compatibilité complète et correcte. C'est néanmoins ce que sont

forcés de faire la grande majorité des logiciels, qu'ils soient libres ou propriétaires, pour pouvoir manipuler des formats fermés.

Au contraire, si des formats ouverts étaient utilisés par tous, chacun serait libre de choisir le logiciel qu'il préfère (Microsoft Office ou autre) à cause de ses qualités propres, et non par peur de ne pouvoir lire des formats secrets.

2.3.2 Indépendance et longévité

Comme n'importe qui est en mesure de concevoir des logiciels conformes à un format ouvert, **l'utilisation de ce formats ouverts à des fins d'archivage est un choix judicieux**. Il est implicitement garanti qu'il y aura toujours des logiciels en mesure de le manipuler adéquatement, même si le format a été lancé par une compagnie défunte.

2.3.3 Formats ouverts pour le secteur public

Transparence et intégrité sont des mots qui devraient fondamentalement être associés au secteur de l'éducation, et au gouvernement en général. Transparence dans le sens où l'information devrait être disponible au grand public sans discrimination ; intégrité dans le sens où l'information devrait être accessible en tout temps et à toute époque (tant que le support physique des archives existe).

Parmi les nombreuses raisons pour lesquelles les formats ouverts devraient être promus et utilisés par ces derniers, il y a :

- la nécessité des organisations gouvernementales d'être indépendantes d'une entreprise particulière pour à la création et à l'échange de documents ;
- l'assurance de pouvoir accéder aisément à ces données, même dans 100 ans (c'est impossible à assurer avec un format propriétaire, qui change régulièrement selon le bon vouloir de son créateur) ;
- garantir aux citoyens l'accès éternel à ces documents ;
- ne pas exercer de discrimination monétaire : par exemple, publier des documents dans le format d'une suite bureautique propriétaire restreint leur lecture aux personnes qui sont en mesure de payer le logiciel correspondant.

2.4 Quelques exemples

Voici une liste non exhaustive de formats ouverts populaires.

OASIS OpenDocument [14] (`.odt`, `.ods`, `.odp`, etc.) : formats de documents de bureautique permettant de représenter du texte, des tableaux de nombres, des diaporamas, des diagrammes, etc. Ces formats s'appuient sur plusieurs autres normes telles que XML, SVG, MathML, etc. Ils pourraient remplacer les formats fermés `.doc`, `.xls`, `.ppt`, etc. Ces formats sont supportés par OpenOffice, Google Docs & Spreadsheets, KWord, Abiword, Gnumeric, ajaxWrite, Ichitaro, TextEdit, Zoho Writer, etc.

OGG Vorbis [5] (`.ogg`) : format de compression audio avec pertes libre de brevets. La compression audio « avec pertes » permet d'éliminer une partie de l'information audio que l'oreille humaine ne peut (généralement) pas entendre, afin d'obtenir un échantillon nécessitant moins d'espace de stockage. OGG Vorbis offre une qualité sonore supérieure à son concurrent non libre le MP3 à des niveaux de compression équivalents.

OGG Theora [6] (`.ogg`, `.ogm`) : format vidéo libre de brevets qui sert à représenter les images d'une vidéo de façon compacte. Un fichier vidéo au format Theora peut aussi comporter une trame sonore dans le format Vorbis. Theora est comparable au format non libre MPEG-4.

Free Lossless Audio Codec (`.flac`) : codec audio sans perte de qualité sonore permettant de stocker des enregistrements en utilisant moins d'espace qu'un format non compressé. À savoir que la compression FLAC est « sans pertes », c'est-à-dire qu'il n'y a aucune dégradation de la qualité sonore (comparativement aux codecs « avec pertes » comme le OGG Vorbis).

Scalable Vector Graphics [27] (.svg) : un format d'images vectorielles supporté par les navigateurs Web récents, à l'exception d'Internet Explorer 7 qui ne le prend complètement pas en charge. Les images vectorielles sont des images interprétées par des calculs mathématiques. Elles possèdent l'avantage d'une qualité optimale, peu importe la taille, mais peuvent nécessiter une puissance de calcul plus grande dans le cas d'images photoréalistes. SVG permet aussi de représenter des animations en plus des images statiques.

Portable Network Graphics [26] (.png) : un format d'images matricielles « non destructeur » (sans perte de qualité). Un format matriciel décrit une image comme un tableau à deux dimensions associant une couleur à chaque position. Les couleurs peuvent être spécifiées par des triplets indiquant la quantité de rouge, de vert et de bleu, ou tirées d'une palette fixe. Il est également possible d'associer un niveau d'opacité distinct à chaque point de l'image. Contrairement à SVG, PNG, étant un format d'images matricielles, est à taille fixe et dépend de la résolution d'écran ou d'impression.

Portable Document Format [9] (.pdf) : format ayant l'avantage de préserver la mise en forme d'un document (images, polices de texte, objets graphiques, espaces, etc), peu importe l'application ou la plateforme utilisée pour le lire. Un désavantage de cette préservation rigide de la mise en page est qu'il est difficile pour l'utilisateur de modifier des documents PDF.

Voir aussi :

- http://fr.wikipedia.org/wiki/Format_ouvert
- <http://www.openformats.org/fr6>
- http://fr.wikipedia.org/wiki/Alternatives_ouvertes_aux_formats_fermés

2.5 Pourquoi pas « XPS » ou « Office OpenXML » de Microsoft Office 2007 ?

Dans un contexte de formats ouverts, il faut toutefois se mettre en garde contre les « faux » formats ouverts. XPS (XML Paper Specification) et Office OpenXML sont des formats mis en avant par Microsoft dans le but de raffermir son emprise sur les formats que nous utilisons dans la vie courante.

Les raisons de fuir le nouveau format OOXML (Office Open XML, à ne pas confondre avec OpenDocument ou OpenOffice) sont nombreuses, au point où elles pourraient faire l'objet d'un argumentaire entier. Pour des raisons de simplicité et de concision, nous ne mentionnerons que quelques points saillants sous forme simplifiée, et nous nous attarderons uniquement sur le format OOXML.

2.5.1 OOXML n'est pas réellement ouvert

Contrairement au format OpenDocument (qui a été ratifié comme le standard ISO/IEC 26300:2006), le format Office OpenXML semble n'être qu'une tactique publicitaire de Microsoft pour tenter de conserver ses contrats gouvernementaux en se proclamant « ouvert » tout en s'assurant d'être les seuls à être réellement en mesure d'utiliser le format à sa pleine capacité.

Notamment, de nombreuses « fonctions » ne sont que des références non documentées vers des produits Microsoft antérieurs. Dans le quatrième volume de la documentation d'OOXML [22], on retrouve :

- autoSpaceLikeWord95 (spécifier les espaces « comme Word 95 », section 2.15.3.6) ;
- lineWrapLikeWord6 (retours à la ligne « comme Word 6 », section 2.15.3.31) ;
- useWord2002TableStyleRules (tableaux « comme Word 2002 », section 2.15.3.6) ;
- useWord97LineBreakRules (« retours à la ligne de Word 97 », section 2.15.3.6).

Ces fonctions ne font donc que dire au lecteur « vous saurez ce que ça fait si vous détenez la clé du fonctionnement de Microsoft Word 6/95/97/2002/etc. ». Évidemment, personne sauf Microsoft ne sait comment ça marche ! Ce genre de « fausse documentation » est un exemple flagrant du type d'« ouverture » dont Microsoft fait preuve.

Ainsi, certaines fonctionnalités, au lieu d'être expliquées comme les autres dans la documentation du format OOXML, ont une mention « définie par l'application », ce qui veut dire que leur fonctionnement est entièrement dépendant du logiciel qui lit le format. En d'autres mots, c'est comme dire à quelqu'un « c'est très simple de fabriquer une station spatiale, regarde, tu prends une station, et de l'espace » !



FIG. 1 – Pour implanter correctement la spécification Office Open XML dans son logiciel, il faut passer au travers de 6039 pages de documentation.

2.5.2 6039 pages

La « spécification » (description du fonctionnement du format) d'OOXML est répartie sur plusieurs livres, totalisant plus de six mille pages, ce qui est inutilement complexe pour un simple format de documents de bureautique.

Pour comparer, la spécification entière du format OASIS OpenDocument (ISO/IEC 26300:2006) tient en 722 pages.

2.5.3 Autres critiques et références additionnelles

- OOXML a été conçu pour le passé, et non l'avenir [1]
- Les 6039 pages de la spécification sont impossibles à évaluer dans un délai raisonnable [7]
- Le format de dates décimal est utilisé et ignore le standard ISO 8601 [20]
- Omissions et contradictions dans la spécification OOXML [15]
- Format de numérotation ignorant les standards ISO 10646 et W3C XSLT [15]
- Microsoft réinvente la roue au lieu d'utiliser des standards existants (comme MathML, SVG, les formats de numérotation ISO 10646 et les dates ISO 8601 mentionnés ci-haut)
- Objections diverses au format OOXML [15]
- When is a standard not a standard? [12]
- Critiques du format OOXML sur Wikipédia [28]

2.6 Propositions pour l'usage de formats ouverts

Proposition 1 : *Que l'Université de Montréal et la FAÉCUM offrent en format ouvert tous les documents destinés au public, étudiants, professeurs et autres personnes affiliées à l'université. Dans les cas où des formats propriétaires doivent tout de même être distribués pour des raisons de transition, qu'elles s'assurent d'offrir également les documents en format ouvert (si applicable) sur un pied d'égalité.*

Proposition 2 : *Que l'Université de Montréal et la FAÉCUM informent les destinataires de la raison pour laquelle des formats ouverts sont utilisés pour la distribution de documents.*

Proposition 3 : *Que l'Université de Montréal et la FAÉCUM fassent la promotion de l'utilisation de logiciels favorisant les formats ouverts et l'interopérabilité.*

Proposition 4 : *Que l'Université de Montréal encourage les étudiants et professeurs à utiliser des formats ouverts lors de la réalisation de travaux.*

3 Logiciels libres

3.1 Qu'est-ce que c'est ?

De façon générale, un *logiciel* est un ensemble de données, de programmes et de documents destinés à accomplir une certaine tâche. Il existe des logiciels pour traiter du texte, gérer des tableaux de données, dessiner par ordinateur, etc. Plusieurs de ces logiciels sont développés par des entreprises qui les rendent accessibles aux utilisateurs sous licence et moyennant un paiement. D'autres logiciels sont disponibles gratuitement pour le grand public.

Il existe plusieurs types de logiciels, dont les applications, les utilitaires et les systèmes d'exploitation. Une *application* permet de manipuler ou de produire des documents. Par exemple, cela inclut les traitements de texte, les tableurs, les logiciels de dessin, etc. Un *utilitaire* permet quant à lui de manipuler des données et d'aider à la maintenance de l'ordinateur. Par exemple, un logiciel permettant de compresser des fichiers ou de réparer des documents endommagés peut être considéré comme un utilitaire au même titre qu'un défragmenteur de disque. Un *système d'exploitation* fournit finalement un environnement dans lequel les logiciels peuvent cohabiter et s'exécuter.

Il existe quatre catégories de logiciels en fonction de la licence sous laquelle ils sont distribués.

Commercial Pour utiliser un logiciel commercial, le client doit en faire l'achat. Le fournisseur d'un tel logiciel offre des informations générales à propos de son produit, des vidéos montrant ses fonctions de base mais aucune version d'essai. Microsoft Windows et Microsoft Office sont des exemples de logiciels commerciaux.

Partagiciel Aussi appelé *shareware* en anglais, un partagiciel est un logiciel qui peut être librement distribué mais qui n'offre qu'un nombre limité de fonctionnalités ou ne fonctionne que pendant un temps limité. L'utilisateur d'un partagiciel a la possibilité, après la période d'essai, d'acheter la version complète du logiciel. Paint Shop Pro de Corel et WinZip sont des exemples de partagiciels.

Gratuciel Aussi appelé *freeware* en anglais, le gratuciel est offert gratuitement sous certaines conditions, par exemple pour usage personnel seulement. L'anti-virus AVG Free de Grisoft est un exemple de gratuciel.

Libre Aussi appelé *free* en anglais, le logiciel libre, que nous allons étudier plus en profondeur ici, peut être utilisé, modifié et distribué sans restriction. Comme nous allons le voir, c'est beaucoup plus qu'un logiciel gratuit !

Plus précisément, un logiciel *libre* répond aux quatre caractéristiques suivantes.

1. Il peut être utilisé par toute personne ou tout organisme, sans restriction. En particulier, un logiciel libre peut être employé autant pour un usage personnel que commercial.
2. L'utilisateur d'un logiciel libre dispose du droit de le distribuer à toute autre personne ou tout autre organisme.
3. Le code source du logiciel est disponible pour tout utilisateur désirant le consulter. Habituellement, le code pourra être téléchargé depuis Internet.
4. Tout utilisateur peut se procurer et modifier le code source pour ensuite distribuer des versions modifiées du logiciel. Idéalement, les modifications seront bien entendu propagées dans le code source original du logiciel.

Cette définition s'inspire de celle donnée par Roberto Di Cosmo [3] lors de sa conférence sur les logiciels libres donnée en avril 2007, au Département d'Informatique et de Recherche Opérationnelle (DIRO) de l'Université de Montréal. Parmi les logiciels libres figurent des applications, des utilitaires et même des systèmes d'exploitation.

3.2 Avantages des logiciels libres

Le bénéfice le plus évident des logiciels libres est sans nul doute leur coût minime de mise en place et d'utilisation. Il suffit en effet de télécharger le produit souhaité et de l'utiliser, sans devoir payer de redevance

à son fabricant ou se soumettre à un contrat de licence long et restrictif. Par contre, les avantages des logiciels libres vont bien au-delà de la simple réduction des coûts [18] : une plus grande robustesse, une meilleure sécurité, un soutien technique simplifié, etc.

Robustesse Il est très courant d'observer des anomalies de fonctionnement dans son logiciel préféré. L'utilisateur est la plupart du temps impuissant devant ce phénomène, car seul le fabricant peut modifier le logiciel mais consacre habituellement le gros de ses ressources au développement de versions futures de ses logiciels. Acheter la prochaine version devient alors la seule solution disponible pour l'utilisateur si le problème ne peut pas être contourné. Puisque le code source d'un logiciel libre peut être téléchargé et consulté, plusieurs personnes sont susceptibles de l'examiner. Cela permet de mieux cerner les bogues et ainsi d'améliorer la robustesse du produit. L'équipe de développeurs se trouve ainsi élargie par rapport à un logiciel propriétaire. Cette accessibilité du code source est sans nul doute à l'origine de la grande robustesse du système d'exploitation GNU/Linux qui s'exécute de nos jours sur un grand nombre de serveurs hébergeant des sites web importants sur Internet, par exemple celui de Google.

Sécurité Souvent, des failles de sécurité permettent à des personnes malveillantes de s'introduire sur un serveur et d'en perturber le fonctionnement, par exemple en modifiant des pages web hébergées, en surchargeant volontairement la machine ou en manipulant des données sensibles (statut d'utilisateurs, comptabilité, etc.). Une faille de sécurité permet aussi à un virus de s'incruster dans un ordinateur et de causer divers problèmes pour l'utilisateur, allant du simple agacement à de lourdes pertes de données. Dans un logiciel libre, de telles failles sont plus facilement visibles puisque n'importe qui peut consulter le code source. Cela semble à première vue faciliter la vie aux personnes malveillantes, mais il n'en est rien puisque les failles détectées peuvent être corrigées par les développeurs du logiciel et que les utilisateurs peuvent mettre les logiciels à jour sans contraintes de licence. En bref, l'accessibilité du code source fait en sorte que la sécurité doit se fonder sur des hypothèses plus fortes que le secret, ce qui accroît la sécurité.

Pérennité du logiciel Si une entreprise cesse de développer un logiciel dont elle détient jalousement le code source, plus personne ne peut poursuivre le développement de ce logiciel ; conséquemment, le produit « meurt ». L'utilisateur est alors contraint de changer de logiciel si la version actuelle n'est pas compatible avec sa nouvelle configuration, si la licence interdit de continuer à l'utiliser ou s'il a besoin d'une fonctionnalité qui est absente. Par opposition, l'utilisateur peut continuer à employer un logiciel libre, même si ce dernier n'est plus développé par son fabricant. Aucune restriction légale ne lui empêche de le faire, mais des problèmes de compatibilité peuvent bien entendu survenir. Par contre, un logiciel libre ne « meurt » pas ; il continue toujours d'exister et son développement peut être repris par toute personne désireuse de le faire, y compris un fervent utilisateur.

Liberté de mise à jour L'utilisateur peut être contraint d'adopter les nouvelles versions de son logiciel favori et de faire évoluer son matériel en conséquence, ce qui occasionne évidemment des coûts. Celui qui ne se soumet pas à cette contrainte d'évolution subit des désagréments comme l'impossibilité d'ouvrir des documents produits par des nouvelles versions, la cessation de mises à jour de sécurité, etc. Pire encore, les organismes qui paient des coûts annuels pour renouveler des licences peuvent même être forcés légalement de faire la mise à jour si le concepteur décide de ne plus renouveler leur contrat de licence. D'un autre côté, si la version actuelle de son logiciel libre préféré lui convient parfaitement, l'utilisateur n'est pas obligé, lorsqu'une nouvelle version voit le jour, de l'adopter. Mais celui désireux de suivre l'évolution de son logiciel peut le faire sans autre coût que le téléchargement et l'installation de mises à jour. Ainsi, avec les logiciels libres, l'utilisateur met son produit à jour **au besoin**, il est **libre** de le faire à sa convenance.

Égalité des chances en éducation Bon nombre d'étudiants se voient contraints de faire l'achat de logiciels pour leurs cours, parfois à coûts réduits si des ententes préalables ont été établies entre l'université et les fabricants. Ceux qui n'ont malgré tout pas les moyens de se procurer les logiciels doivent travailler uniquement à l'université, installer des copies piratées des logiciels requis sur leurs ordinateurs personnels ou

utiliser d'anciennes versions à leur disposition. D'un autre côté, quand la formation exige principalement des logiciels libres, toute personne disposant chez elle d'un ordinateur peut télécharger et installer les logiciels dont elle a besoin pour sa formation ; l'université a également le droit de distribuer directement ces logiciels aux étudiants. Cela évite des coûts pour les étudiants, mais aussi pour l'université qui n'a pas à établir d'entente avec les fournisseurs pour la distribution de logiciels aux étudiants. Cela libère aussi des postes de travail à l'université puisque davantage d'étudiants peuvent travailler chez eux ou avec leur ordinateur portable.

Soutien technique Une grande partie du travail du personnel de soutien technique consiste à administrer les serveurs et à résoudre divers problèmes techniques. Leur tâche se voit alourdie par l'impossibilité de savoir exactement comment fonctionnent les produits, la nécessité de trouver des parades à des bogues qui ne seront corrigés que quelques années plus tard, dans des versions futures des logiciels, et la nécessité de supporter de multiples versions des logiciels. Avec tout ce travail, il ne reste pas beaucoup de temps aux techniciens pour aider les usagers. D'un autre côté, la robustesse des logiciels libres diminue les coûts de maintenance, permettant aux techniciens de se consacrer à des tâches plus intéressantes telles que l'apprentissage de nouvelles technologies et l'assistance aux employés et aux étudiants. De plus, les techniciens ont moins besoin de traiter les détails spécifiques de chaque version d'un même produit. En cas de problème technique avec une ancienne version, une mise à jour vers la plus récente version, toujours possible sans coût, permet soit de résoudre la difficulté, soit de donner aux techniciens les outils pour la traiter.

3.3 Philosophie du Libre

Beaucoup de gens se demandent comment « vit » un logiciel libre. Il existe en fait plusieurs processus de développement pour un tel produit. Dans le mode le plus simple, un programmeur amateur construit un logiciel pour résoudre un problème précis et décide de le mettre à la disposition de tous les utilisateurs en le publiant sur Internet. Le développement d'un tel logiciel peut par exemple avoir lieu dans le cadre d'un projet de recherche universitaire. Même si la personne qui a construit la première version du logiciel décide de ne plus travailler sur ce dernier, une autre personne ou un organisme désireux de faire évoluer le logiciel peut le télécharger et le modifier à sa guise. Le système d'exploitation GNU/Linux a par exemple débuté de cette façon.

D'autres logiciels libres proviennent de produits commerciaux que leurs concepteurs ont rendus disponibles pour la communauté. Par exemple, le navigateur web Mozilla Firefox est un descendant du navigateur Netscape, mais il a beaucoup évolué depuis que le code source original est devenu libre. Une entreprise rendant un logiciel libre de cette façon se réserve habituellement le droit d'en faire des versions non libres plus étendues, mais la version libre demeurera toujours disponible.

Maintenant, comment un informaticien peut-il faire de l'argent avec un logiciel libre ? À première vue, cela semble difficile. Par contre, supposons qu'une entreprise éprouve un besoin particulier qu'aucun produit sur le marché ne puisse combler. Cette entreprise engagera alors un informaticien pour développer une solution. Cet informaticien pourrait fort bien partir d'un logiciel libre existant, lui ajouter la fonctionnalité requise et rendre cette fonctionnalité disponible pour qu'elle soit intégrée à la version officielle du logiciel. L'entreprise qui a engagé l'informaticien a, dans ce cas, non seulement comblé son besoin initial, mais a en plus permis à la communauté entière de bénéficier de la solution. Ceci contribue à enrichir le logiciel en question et encourage d'autres entreprises à procéder de même pour développer de nouvelles fonctionnalités.

3.4 Exemples de logiciels libres

Il existe un très grand nombre de logiciels dont voici quelques exemples. Tous les logiciels présentés ici comportent l'avantage de la portabilité, c'est-à-dire qu'ils sont disponibles autant pour Microsoft Windows, Apple Macintosh que GNU/Linux.

Mozilla Firefox [13] Navigateur web semblable à Microsoft Internet Explorer doté de plusieurs fonctionnalités intéressantes telles que la navigation par onglets et le blocage des fenêtres *pop-up*. Contrairement

à Microsoft Internet Explorer, ce navigateur ne prend pas en charge ActiveX qui est une source importante de problèmes de sécurité. Son apparence peut être personnalisée par des thèmes et un système de composantes enfichables permet d'y ajouter une foule de fonctionnalités.

Mozilla Thunderbird [13] Client de messagerie électronique semblable à Microsoft Outlook Express. Comme son complément Firefox, Mozilla Thunderbird procure une grande sécurité, permettant par exemple de filtrer les messages indésirables, de ne pas afficher les images des courriers électroniques et dispose lui aussi d'un système d'extensions permettant de greffer des nouvelles fonctionnalités.

OpenOffice.Org [16] Suite bureautique comportant un traitement de texte, un tableur, un concepteur de diaporamas, etc., et utilisant un format de documents ouvert appelé OASIS OPENDOCUMENT (voir section 2.4), mais il peut également lire et écrire des fichiers dans le format de Microsoft Office 95/97/2000/XP/2003.

The GIMP [23] Logiciel d'édition d'images permettant entre autres la retouche de photographies. The Gimp prend en charge plusieurs formats d'images courants et possède des fonctionnalités rivalisant avec le logiciel commercial PhotoShop de la firme Adobe.

Inkscape [10] Logiciel de dessin vectoriel utilisant principalement le format SVG (voir section 2.4).

7-Zip [17] Logiciel de compression de données permettant de stocker des fichiers dans des archives au format 7z. 7-Zip prend aussi en charge le format ZIP plus courant ainsi que la décompression d'archives dans d'autres formats tels que RAR, CAB, etc.

Proposition 5 : *Que l'Université de Montréal installe les logiciels Mozilla Firefox, Mozilla Thunderbird et OpenOffice.Org sur tous ses postes informatiques afin que tout étudiant ou employé ait la possibilité de les découvrir.*

Proposition 6 : *Que l'Université de Montréal favorise la distribution massive aux étudiants de disques compacts regroupant des logiciels libres. Des disques tels que The Open CD [24] ou des distributions de GNU/Linux pourraient par exemple être offerts à faible coût dans les librairies de l'Université et aux comptoirs multiservices de la FAÉCUM.*

Évidemment, le système d'exploitation GNU/Linux est un autre exemple important de logiciel libre. Ce système permet d'accomplir à peu près les mêmes tâches (ou parfois plus) que son homologue Microsoft Windows tout en étant affranchi des contraintes de licences. Son architecture le rend moins vulnérable aux virus, et l'utilisateur a accès à une multitude d'interfaces semblables à celles de Microsoft Windows, Apple Mac OS et autres (selon le choix de l'utilisateur). L'installation de GNU/Linux a été grandement simplifiée ces dernières années et des logiciels sont disponibles pour exploiter presque tous les types de fichiers, même certains formats propriétaires.

Nous croyons qu'une personne utilisant son ordinateur pour des fonctions de base telles que naviguer sur Internet, consulter sa messagerie électronique et se servir d'un traitement de texte pourrait sans trop de difficulté migrer vers GNU/Linux.

Proposition 7 : *Que l'Université de Montréal incite les étudiants à découvrir GNU/Linux et à l'utiliser. Ceci peut se faire entre autres en installant GNU/Linux sur quelques postes de l'université, en offrant aux étudiants des distributions de GNU/Linux sur disques compacts, en offrant du soutien technique pour ce système, etc.*

3.5 Exemples de succès du libre

Plusieurs écoles et gouvernements ont migré avec succès vers des logiciels libres, permettant dans bien des cas une réduction des coûts, mais aussi l'accroissement de la fiabilité des systèmes. Par exemple, dans la commission scolaire de Kamloops, en Colombie-Britannique, les écoles primaires et secondaires ont troqué leurs solutions logicielles à base de Novell pour un système fondé sur GNU/Linux [2].

Initialement, ils ont rencontré de la résistance de la part des professeurs qui ne souhaitaient pas adapter leurs cours aux nouveaux logiciels et des techniciens formés pour les logiciels propriétaires. Par contre, lorsque

les techniciens ont été formés pour administrer les logiciels libres, ils ont vu leur charge de travail diminuer en raison de la réduction du travail de maintenance nécessaire et ont pu venir en aide aux professeurs.

Lors de sa conférence au DIRO, en avril 2007, Roberto Di Cosmo a mentionné l'implantation de OpenOffice.Org au sein de ministères français, un projet qui fut fructueux. Microsoft ne voulait plus renouveler leur contrat de licence annuel pour Office 97 si bien qu'ils devaient passer à la version 2003. Cela occasionnait des coûts de formation en plus de la licence. Au lieu de former les employés à Office 2003, ils les ont formés à OpenOffice.Org, ce qui a donné des résultats fructueux. Dans le cas du ministère des Finances, l'objectif n'était pas d'économiser de l'argent, et pourtant, ils en ont économisé malgré tout !

3.6 Le problème de la formation

La nécessité de former les employés est un argument fréquemment utilisé contre la migration vers une nouvelle solution logicielle. Par exemple, plusieurs personnes pensent que passer de Microsoft Office à OpenOffice.Org nécessitera un long apprentissage, que la formation à OpenOffice.Org sera aussi longue que celle à Microsoft Office et qu'elle sera coûteuse. Il n'en est rien, car les mêmes concepts généraux reviennent d'un logiciel à l'autre. L'utilisateur devra certes prendre quelques minutes pour s'y retrouver, mais il réussira sans trop de mal à faire fonctionner le nouveau logiciel.

Nous croyons que passer de Microsoft Internet Explorer à Mozilla Firefox ou de Microsoft Office vers OpenOffice.Org est à peu près aussi facile que migrer vers une nouvelle version du produit de la même entreprise.

Idéalement, il faudrait aider les étudiants et les employés à se libérer de ce mythe selon lequel il est nécessaire d'apprendre et utiliser les logiciels propriétaires puisqu'ils sont employés partout dans l'industrie. La formation ne devrait pas cibler un logiciel particulier, mais plutôt des concepts généraux : comment utiliser un traitement de texte, comment maîtriser un tableur, etc. ? Le logiciel employé pour la formation ne devrait être qu'un exemple appuyant le concept, pas le concept lui-même.

Par contre, il ne faut pas tomber dans l'autre extrême en remplaçant tous les logiciels propriétaires par des logiciels libres du jour au lendemain. L'idée ici ne consiste pas à bannir ces produits commerciaux mais plutôt à en faire une solution parmi tant d'autres plutôt que *la* seule et unique solution. En d'autres mots, l'utilisateur choisissant un logiciel propriétaire doit le faire uniquement si le produit satisfait ses besoins.

Proposition 8 : *Que l'Université de Montréal installe plusieurs logiciels différents pour le traitement de texte, la navigation sur Internet, le courrier électronique, etc. sur les postes à la disposition du personnel et des étudiants.*

Toute personne devrait pouvoir choisir le logiciel qu'elle désire employer pour accomplir une tâche donnée. À première vue, cela peut causer des problèmes de communication entre usagers, mais les formats de fichiers ouverts (voir section précédente) résolvent la question.

Proposition 9 : *Que l'Université de Montréal diversifie les systèmes d'exploitation installés sur ses postes de travail. L'Université devrait offrir aussi bien des postes Microsoft Windows que des postes Apple Macintosh et GNU/Linux.*

Proposition 10 : *Que l'Université de Montréal offre aux étudiants de la formation pour les logiciels libres afin de leur faire découvrir qu'ils peuvent facilement migrer vers ces derniers.*

4 Rétrocession à la communauté du libre

4.1 Pourquoi redonner à la communauté ?

Au cours des sections précédentes, nous avons exposé plusieurs bénéfices que pouvaient apporter à l'Université de Montréal les logiciels libres et les formats ouverts. Or cette collaboration peut aller dans les deux sens. L'Université de Montréal peut effectivement apporter beaucoup à la communauté du libre, comme nous le verrons un peu plus loin. Cette aide aura pour effet de mettre l'Université de Montréal sur la carte du monde des logiciels libres, ce qui est évidemment très positif pour sa visibilité et sa réputation.

4.2 Hébergement de serveurs miroirs

Comme plusieurs logiciels libres sont téléchargés par une grande quantité d'utilisateurs, qu'ils sont souvent de taille assez importante et qu'il est important de conserver les anciennes versions de chaque logiciel pour des raisons de rétro-compatibilité, il est nécessaire d'avoir une grande quantité d'espace de stockage et de bande passante pour les héberger. Comme la plupart des hôtes de ces logiciels le font bénévolement, ils ne disposent pas nécessairement de ressources suffisantes pour héberger seuls de façon efficace une quantité importante de logiciels libres. C'est pourquoi il est souvent nécessaire d'héberger ces logiciels simultanément sur plusieurs serveurs, appelés serveurs miroirs, situés un peu partout dans le monde. Comme chaque utilisateur désirant télécharger un logiciel donné peut choisir le serveur duquel il le télécharge, les serveurs ne sont pas surchargés, d'où l'importance des serveurs miroirs.

Les serveurs miroirs étant un élément essentiel à la distribution des logiciels libres, une contribution possible de l'Université de Montréal à la communauté du libre serait la mise en place de serveurs miroirs pour l'hébergement de divers projets libres. La présence de tels serveurs à Montréal permettrait un accès plus rapide au contenu pour les utilisateurs de l'Université de Montréal en particulier et du Québec en général, qui doivent dans certains cas télécharger leurs logiciels libres de serveurs surchargés ou éloignés.

Plusieurs universités canadiennes offrent déjà un service de serveurs miroirs, notamment l'Université d'Alberta [25] qui héberge tous les projets du projet GNU ou l'Université de Sherbrooke, qui héberge plusieurs distributions GNU/Linux. La gestion de ces serveurs peut être effectuée par l'Université, comme c'est le cas à l'Université d'Alberta, ou par le groupe d'usagers GNU/Linux local (ici, le GULUM), comme c'est le cas à l'Université de Sherbrooke (dans leur cas, le GULUS [8])

Proposition 11 : *Que l'Université de Montréal mette en place des serveurs dédiés à l'hébergement de projets libres afin de faciliter leur téléchargement par les étudiants et les membres de la communauté du libre.*

4.3 Services aux étudiants

Un grand service que pourrait rendre l'Université de Montréal à sa communauté du libre serait de mettre en place un service de soutien technique visant les logiciels libres utilisés par les étudiants. La présence d'un tel service pourrait rassurer certains étudiants et les encourager à adopter eux aussi le libre, en plus d'être une valeur ajoutée au séjour à l'Université de Montréal, ce qui pourrait attirer certains candidats.

En plus d'offrir du soutien technique, ce service pourrait également faire la promotion des formats ouverts et des logiciels libres dans le but de favoriser leur adoption par les étudiants, les professeurs et le personnel de l'Université.

Proposition 12 : *Que l'Université de Montréal mette en place un service de soutien technique et de promotion des formats ouverts et des logiciels libres, à la disposition des étudiants, des professeurs et du personnel de l'Université.*

Il pourrait également être intéressant d'offrir un espace d'hébergement pour les projets libres des étudiants et des chercheurs de l'université. Les responsables de ces projets n'auraient ainsi plus à se casser la tête pour trouver de l'hébergement. La présence à l'université de cet hébergement pourrait de plus donner une plus

grande visibilité et un aspect officiel à ces projets qui, s'ils ont du succès, contribueraient à la réputation de l'Université de Montréal, en plus d'encourager les responsables de ces projets à les distribuer sous licence libre.

Proposition 13 : *Que l'Université de Montréal propose de l'espace pour héberger les projets libres de ses étudiants et de ses chercheurs et encourage ces derniers à les distribuer sous licence libre.*

4.4 Développement interne

Plusieurs des besoins de l'Université de Montréal sont partagés par les autres universités. Certains des outils développés ici, ou pour ici par des sous-traitants, pourraient donc résoudre les problèmes qui occupent une autre université. En publiant ces outils sous des licences libres, ces institutions n'auraient ainsi pas besoin de réinventer la roue, et les améliorations qu'elles pourraient apporter aux outils pourraient être récupérées par l'Université de Montréal, qui se retrouverait ainsi avec un meilleur système, et ce, sans avoir à déboursier quoi que ce soit.

De plus, en rendant libres ses outils (son excellent système de gestion de dossiers étudiants, par exemple), l'Université de Montréal deviendrait un chef de file en matière de collaboration interuniversitaire, ce qui est évidemment positif pour sa notoriété.

Proposition 14 : *Que l'Université de Montréal distribue les projets informatiques développés à l'interne ou sous-traités sous des licences libres.*

5 Collaborateurs

L'écriture de ce document n'aurait pas été possible sans l'aide de ces personnes :

Prénom	Nom	Occupation	Programme
Marc	Julien	Étudiant	Maîtrise en informatique
Eric	Buist	Étudiant	Doctorat en informatique
Jean-François	Fortin Tam	Étudiant	Baccalauréat en psychologie
Pascal	Lamblin	Étudiant	Doctorat en informatique
Vincent	St-Amour	Étudiant	Baccalauréat en informatique
Gabriel	Filion	Étudiant	Baccalauréat en informatique

Références

- [1] Michael Brauer. ODF is designed for the future, not the past. http://blogs.sun.com/GullFOSS/entry/odf_is_designed_for_the, janvier 2007.
- [2] Bruce Byfield. Kamloops school district gets an education in free software. *Linux.com*, juin 2007. <http://www.linux.com/articles/62285>.
- [3] Roberto Di Cosmo. Page personnelle. <http://www.pps.jussieu.fr/~dicosmo/>, 2007.
- [4] Secrétariat du Conseil du Trésor du Canada. Logiciel libre. http://www.tbs-sct.gc.ca/fap-paf/oss-11/oss-11_f.asp.
- [5] Xiph foundation. Vorbis I specification. http://xiph.org/vorbis/doc/Vorbis_I_spec.html, 2004.
- [6] Xiph foundation. Theora I Specification. http://www.theora.org/doc/Theora_I_spec.pdf, 2006.
- [7] Rupert Goodwins. Six thousand pages, one month, no chance... <http://community.zdnet.co.uk/blog/0,1000000567,100048050-2000331777b,00.htm>, janvier 2007.
- [8] GULUS. <http://www.gulus.org>.
- [9] Adobe Systems Incorporated. PDF Reference, Sixth Edition, version 1.7. http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf, 2007.
- [10] Inkscape. Draw freely. <http://www.inkscape.org/>, 2007.
- [11] University of Western Australia IT Policy Office. Open source software policy & strategy. <http://www.uwa.edu.au/it/policy/existing/opens>, 2002.
- [12] Edward Macnaghten. When is a standard not a standard? <http://www.freesoftwaremagazine.com/node/2110>, mars 2007.
- [13] Mozilla Foundation. Mozilla Firefox, Mozilla Thunderbird. <http://www.mozilla.com/>, 2007.
- [14] OASIS. Open Document Format for Office Applications (OpenDocument) v1.0. <http://www.oasis-open.org/committees/download.php/12572/OpenDocument-v1.0-os.pdf>, 2005.
- [15] Objections to JTC-1 Fast-Track Processing of the Ecma 376 Specification v. 0.1. http://www.grokdoc.net/index.php/E00XML_objections, 2007.
- [16] OpenOffice.Org. Voir <http://www.openoffice.org/>, 2007.
- [17] Igor Pavlov. 7-Zip. <http://www.7-zip.org/>, 2007.
- [18] Jay Pfaffman. It's time to consider open source software. *LinuxInsider*, 2007. Voir <http://www.linuxinsider.com/story/linux-software/57759.html>.
- [19] Québec Services gouvernementaux. Les logiciels libres au gouvernement du québec. <http://www.logiciel-libre.gouv.qc.ca/>.
- [20] Joel Spolsky. My first BillG review. <http://www.joelonsoftware.com/items/2006/06/16.html>, juin 2006.
- [21] Richard Stallman. Le manifeste gnu. <http://www.gnu.org/gnu/manifesto.fr.html>, 1985.
- [22] Ecma TC45. Office Open XML Final Draft, Part 4 : Markup Language Reference, 2006.
- [23] The GIMP Team. GIMP — the GNU image manipulation program. <http://www.gimp.org/>, 2007.
- [24] TheOpenCD. <http://www.theopencd.org/>, 2007.
- [25] University of Alberta Digital Objects Repository. <http://www.ualberta.ca/AICT/UADORe/>.
- [26] W3C. Portable Network Graphics (PNG) Specification (Second Edition). <http://www.w3.org/TR/PNG/>, 2003.
- [27] W3C. Scalable Vector Graphics (SVG) 1.1 Specification. <http://www.w3.org/TR/SVG11/>, 2003.
- [28] Wikipedia. Office Open XML — Wikipedia, The Free Encyclopedia, 2007. [Online; accessed 27-July-2007, http://en.wikipedia.org/wiki/Office_Open_XML#Criticism].
- [29] Wikipédia. Rétro-ingénierie — Wikipédia, l'encyclopédie libre, 2007. [En ligne; Page disponible le 11 août 2007, <http://fr.wikipedia.org/wiki/Rétro-ingénierie>].

Résumé des propositions

Proposition 1	Que l'Université de Montréal et la FAÉCUM offrent en format ouvert tous les documents destinés au public, étudiants, professeurs et autres personnes affiliées à l'université. Dans les cas où des formats propriétaires doivent tout de même être distribués pour des raisons de transition, qu'elles s'assurent d'offrir également les documents en format ouvert (si applicable) sur un pied d'égalité.	6
Proposition 2	Que l'Université de Montréal et la FAÉCUM informent les destinataires de la raison pour laquelle des formats ouverts sont utilisés pour la distribution de documents.	6
Proposition 3	Que l'Université de Montréal et la FAÉCUM fassent la promotion de l'utilisation de logiciels favorisant les formats ouverts et l'interopérabilité.	6
Proposition 4	Que l'Université de Montréal encourage les étudiants et professeurs à utiliser des formats ouverts lors de la réalisation de travaux.	6
Proposition 5	Que l'Université de Montréal installe les logiciels Mozilla Firefox, Mozilla Thunderbird et OpenOffice.Org sur tous ses postes informatiques afin que tout étudiant ou employé ait la possibilité de les découvrir.	10
Proposition 6	Que l'Université de Montréal favorise la distribution massive aux étudiants de disques compacts regroupant des logiciels libres. Des disques tels que The Open CD [24] ou des distributions de GNU/Linux pourraient par exemple être offerts à faible coût dans les librairies de l'Université et aux comptoirs multiservices de la FAÉCUM.	10
Proposition 7	Que l'Université de Montréal incite les étudiants à découvrir GNU/Linux et à l'utiliser. Ceci peut se faire entre autres en installant GNU/Linux sur quelques postes de l'université, en offrant aux étudiants des distributions de GNU/Linux sur disques compacts, en offrant du soutien technique pour ce système, etc.	10
Proposition 8	Que l'Université de Montréal installe plusieurs logiciels différents pour le traitement de texte, la navigation sur Internet, le courrier électronique, etc. sur les postes à la disposition du personnel et des étudiants.	11
Proposition 9	Que l'Université de Montréal diversifie les systèmes d'exploitation installés sur ses postes de travail. L'Université devrait offrir aussi bien des postes Microsoft Windows que des postes Apple Macintosh et GNU/Linux.	11
Proposition 10	Que l'Université de Montréal offre aux étudiants de la formation pour les logiciels libres afin de leur faire découvrir qu'ils peuvent facilement migrer vers ces derniers.	11
Proposition 11	Que l'Université de Montréal mette en place des serveurs dédiés à l'hébergement de projets libres afin de faciliter leur téléchargement par les étudiants et les membres de la communauté du libre.	12
Proposition 12	Que l'Université de Montréal mette en place un service de soutien technique et de promotion des formats ouverts et des logiciels libres, à la disposition des étudiants, des professeurs et du personnel de l'Université.	12
Proposition 13	Que l'Université de Montréal propose de l'espace pour héberger les projets libres de ses étudiants et de ses chercheurs et encourage ces derniers à les distribuer sous licence libre.	13
Proposition 14	Que l'Université de Montréal distribue les projets informatiques développés à l'interne ou sous-traités sous des licences libres.	13