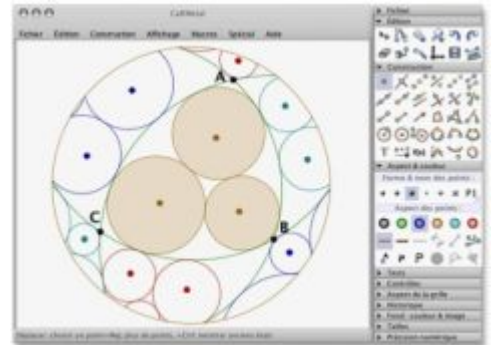


CaRMetal : Entretien avec Eric Hakenholz

[CaRMetal](#) est un excellent logiciel éducatif de géométrie dynamique. CaRMetal est un logiciel libre (sinon d'ailleurs il n'aurait jamais vu le jour puisqu'il est directement issu du code source d'un autre logiciel libre). CaRMetal est développé et maintenu par un professeur de mathématiques français fort expérimenté qui a des choses à dire.



Autant de bonnes raisons de rencontrer Eric Hakenholz pour un entretien témoignage riche et intéressant.

Remarque : Le Framablog poursuit ici sa série de mises en valeur et en lumière de projets éducatifs libres et de ceux qui les portent.

Entretien avec Eric Hakenholz

Eric bonjour, une rapide présentation personnelle ?

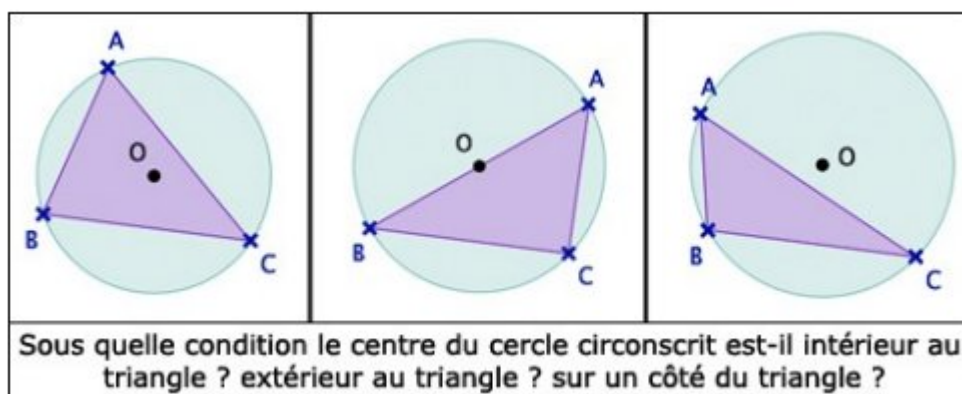
Je suis professeur de mathématiques depuis 1989. J'ai passé mes deux premières années d'enseignement dans une école française installée dans un lycée public australien (Melbourne). Grâce à ce poste d'enseignant, j'ai eu la chance de pouvoir travailler dès 89 dans un environnement très informatisé et d'être témoin de nombreuses pratiques pédagogiques liées à l'utilisation du réseau (les nombreux [Mac Plus](#) de l'établissement étaient tous interconnectés). J'ai programmé là-bas mon premier logiciel Mac : un traceur de courbes et de surfaces, dont l'interface paraîtrait aujourd'hui bien désuète.

J'ai passé ensuite douze années dans l'académie de la Réunion, où j'ai « milité » pour l'utilisation du logiciel [Cabri-Géomètre](#) par le biais de formations et de publications (sites, journal bimestriel [abraCAdaBRI](#), développement de l'utilitaire [Cabri File Exchange](#)). En 2004, en arrivant dans l'académie de Toulouse, j'ai cherché s'il existait un autre logiciel de géométrie dynamique, libre et gratuit cette fois-ci, qui puisse rivaliser avec la puissance et la pertinence de Cabri : j'ai découvert cette année-là le logiciel [C.a.R. \(Compass and Ruler\)](#).

Qu'est-ce que le logiciel CaRMetal ?

C'est un [logiciel de géométrie dynamique](#), un environnement qui permet notamment d'explorer de manière interactive les propriétés géométriques des figures. Dans ce type de logiciel, les utilisateurs peuvent effectuer des constructions géométriques et déplacer les objets sur lesquels se basent la construction.

Pendant ces déplacements les propriétés sont conservées, et de nombreuses constatations peuvent surgir à ce moment-là : le simple mouvement des figures permet de mettre en évidence de nombreuses propriétés géométriques qui seraient passées inaperçues dans un environnement papier-crayon.



Lorsque j'ai découvert le logiciel C.a.R. de René Grothmann en 2004, je me suis aperçu qu'il contenait de nombreuses fonctionnalités très originales, avec une gestion très pointue

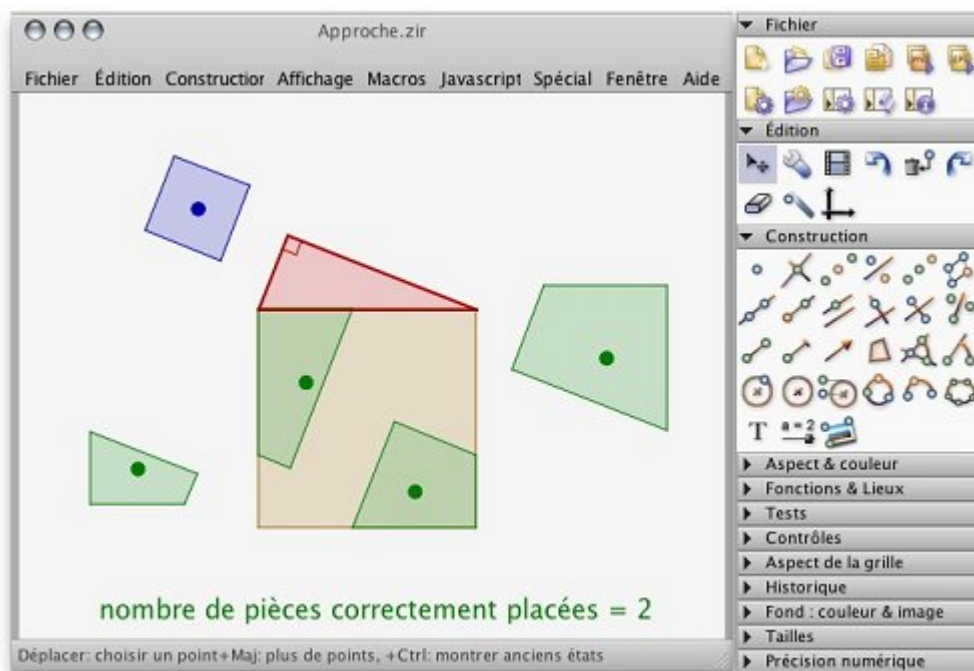
des macro-constructions (possibilité de se créer des outils à partir de ceux déjà existants). Cependant, malgré tout l'émerveillement que me procurait la découverte des possibilités de C.a.R., je gardais un oeil assez critique sur la façon d'accéder aux fonctionnalités, sur l'interface utilisateur.

CaRMetal au départ (février 2006) était juste un « TP » motivé par mon envie d'apprendre le langage java. Mes années de programmation en Pascal et en C m'ont amené petit à petit à vouloir changer pour un langage objet multi-plateforme, sur un code qui puisse être exécuté aussi bien sur GNU/Linux, Macintosh ou Windows. Les sources du logiciel C.a.R., écrits en Java par un programmeur clair, rigoureux et doué, ont rempli à merveille leur rôle de tuteur pendant mes deux premiers mois d'apprentissage. Le but que je m'étais fixé au départ était de reprendre les algorithmes mathématiques de C.a.R. et de les installer dans une interface neuve, intuitive, et toute en manipulation directe. Les choses sont allées plus vite que je ne le pensais, et les premiers résultats m'ont encouragé à changer d'optique : ce qui n'était qu'un TP est rapidement devenu un vrai projet, que j'ai décidé de publier sur Internet et de mettre à jour régulièrement.

On développe aujourd'hui des interfaces avec le soucis de donner une réponse immédiate à l'action de l'utilisateur : on s'aperçoit qu'il est presque toujours possible, avec des efforts de programmation supplémentaires, de supprimer toute situation bloquante (comme un dialogue avec boutons *ok* ou *appliquer*) ou toute situation qui ralentirait l'action (comme l'obligation systématique de dérouler un menu et de chercher l'item). Développée dans ce sens, avec le soucis d'une organisation claire et rationnelle des outils, une interface graphique permet de travailler d'une façon beaucoup plus fluide, avec un gain de temps très sensible, et une impression de liberté pour l'utilisateur qui s'en trouve nettement améliorée.

C'est avec l'objectif de produire un logiciel qui réponde un peu à ces critères que j'ai programmé CaRMetal : les seules boîtes à dialogues qui existent sont celles relatives à l'enregistrement et l'ouverture des fichiers. En phase de construction ou d'édition des objets, aucun menu n'est à dérouler, l'accès aux fonctionnalités est sans étape intermédiaire et les modifications apportées à l'aspect des objets sont visibles immédiatement dans la fenêtre de travail.

Jusqu'en décembre 2007, je suivais de très près l'évolution du logiciel C.a.R. en incluant dans CaRMetal, version après version, les modifications que René apportait à son « moteur mathématique ». Mais à partir de cette date j'ai souhaité mettre la main sous le capot algorithmique, ce qui nécessitait l'abandon du *copier-coller* de C.a.R. vers CaRMetal, et plusieurs dizaines de fonctionnalités qui me tenaient particulièrement à coeur ont ainsi pu voir le jour.



En quoi se distingue-t-il des autres logiciels de la catégorie ? Je pense en particulier à Geogebra lui aussi libre et sous Java.

L'interface utilisateur de CaRMetal se distingue de celle des autres logiciels pour toutes les raisons évoquées

précédemment. Certains utilisateurs me disent aussi que, pour leurs élèves, les retours vidéos systématiques ainsi que le caractère graphique des palettes et de l'inspecteur rend pratiquement nul le temps de prise en main du logiciel. D'autres, en général des adultes habitués à des logiciels différents, disent que l'absence de menus-boutons classés par types d'objets les perturbent : je laisse donc le soin aux utilisateurs de se faire leur propre opinion !

Bien sûr il y a de nombreuses fonctionnalités originales qui ne se retrouvent que dans CaRMetal, mais dire uniquement ceci serait oublier un peu trop vite que c'est aussi le cas pour d'autres logiciels de géométrie dynamique, comme Geogebra (je pense notamment à sa récente fonctionnalité « tableur » qui ne se retrouve nulle part ailleurs, du moins dans une forme très proche de celle des tableurs).

Comparer les fonctionnalités est un exercice assez subjectif : chacun peut faire son propre tableau avec en ligne le nom des logiciels, en colonne ses propres fonctionnalités, et ainsi faire apparaître une belle et unique ligne entièrement cochée... Je préfère de loin m'intéresser à l'ergonomie et l'expérience utilisateur, mais je vais tout de même essayer de faire une liste non exhaustive de quelques fonctionnalités "exotiques" de CaRMetal :

- Les diaporamas : CaRMetal peut transformer automatiquement (en un seul clic comme disent les commerciaux) un dossier contenant des figures en un mini site web. En cliquant sur les liens de la page générée, on charge des applets qui correspondent à chacune des figures.
- Les [objets magnétiques](#) : Cela permet notamment de créer un point sur plusieurs objets ou un point tout simplement attiré par un ensemble d'objets, conformément à une zone d'attraction définie en pixels par l'utilisateur.

- Les [contrôles systèmes](#) : on peut inclure dans la figure des curseurs systèmes, des menus déroulants, des boîtes à cocher, des boîtes d'entrées numériques et des boutons poussoir.
- Le [JavaScript](#) : un éditeur graphique permet de créer des scripts utilisant l'intégralité du langage javascript, enrichi de commandes liées à la pratique de la géométrie dynamique. Le logiciel permet de lancer un script et de l'annuler dans une figure.
- Attribution automatique des noms de points : avec un tableau numérique interactif, il est assez gênant d'avoir à utiliser le clavier. Un dispositif d'attribution de noms permet de sélectionner à la volée et au stylet les noms qu'on veut donner aux objets.
- Le mode « tableau numérique interactif » ([TNI](#)) : un stylet sur un tableau interactif ne se comporte pas tout à fait comme une souris. On perd la notion de *mouse over* et par conséquent bon nombre d'informations que donne la souris lors du survol d'objets d'intérêts (retours vidéos). Va-t-on par exemple, en cliquant, créer un point à une intersection, ou va-t-on louper sa cible et créer un point libre à côté ? En *mode TNI*, un cliquer-glisser équivaut à un survol de souris, et un lâché de stylet correspond à un clic de souris. Dans ce mode, sur un tableau interactif, on garde le même confort visuel et la même précision que lorsqu'on construit à la souris.
- Les exercices : le formateur peut créer une construction-exercice avec comme objectif de faire construire un objet-cible (un orthocentre par exemple). L'élève doit effectuer la construction et s'il la réussit correctement, le message *bravo !* apparaît. Dans ce type de figure c'est le logiciel lui-même qui s'assure que le travail a été effectué sans erreur.

CaRMetal dérive du logiciel C.a.R. (Compass and Ruler) développé par un professeur allemand. C'est parce que ce dernier était un logiciel libre que vous avez pu récupérer les sources et le modifier pour l'adapter à vos besoin. Que cela vous inspire-t-il ?

Certains informaticiens qui vivent en free-lance de leurs productions n'ont pas envie de produire des algorithmes ouverts, de peur que ce qu'ils programment ne soit immédiatement repris et vendu par d'autres. De peur finalement que le bénéfice de leurs travaux, pour lesquels ils ont dépensé beaucoup d'énergie et monopolisé beaucoup de connaissances, aillent dans une autre poche que la leur. Même si je ne fais pas partie de ce monde-là, je peux comprendre cette démarche de protection, par fermeture, dans des secteurs pointus dominés par les majors, les requins et la course au profit. Je préfère de très loin cette démarche cohérente à celle de nombreuses boîtes informatiques entièrement habillées chez Microsoft, et qui d'un autre côté ne jurent que par l'open source, en utilisant abondamment ce *label* dans le seul but de faire du dollars à grands coups de copier-coller. Concernant l'open source, les choses ne sont donc pas si simples et si manichéennes...

C.a.R. est un logiciel libre, et pas seulement open source : il est sous licence GNU-GPL et par conséquent CaRMetal l'est aussi. René Grothmann, le concepteur de C.a.R., a développé son logiciel dans une démarche d'ouverture et de partage : le véritable esprit du libre est-là, dans un comportement plus éthique et moral que purement technique. Reprendre les sources de René, les adapter suivant mes préférences, ajouter de nouvelles fonctionnalités m'a permis de créer un autre logiciel, qui a son tour peut être repris, et amélioré dans le même esprit.

L'idée du libre me fait penser au chanteur folk américain [Woody Guthrie](#), qui, prenant le contre-pied des copyrights déjà en vogue à son époque, écrivait ceci en 1930 dans un recueil

de chansons : *This song is Copyrighted in U.S., under Seal of Copyright #154085, for a period of 28 years, and anybody caught singin' it without our permission, will be mighty good friends of ourn, cause we don't give a dern. Publish it. Write it. Sing it. Swing to it. Yodel it. We wrote it, that's all we wanted to do.*

Rares sont les enseignants développeurs de logiciels libres, êtes-vous reconnu et soutenu par l'Institution ? Est-ce important pour vous ?

N'ayant jamais aimé jouer les commerciaux, je n'ai jamais rien demandé à personne pendant ces quatre années de développement. C'est peut-être la raison pour laquelle je n'ai jamais eu aucun soutien, aucune reconnaissance de la part de ma hiérarchie institutionnelle ici, dans l'académie de Toulouse.

Lorsque je lis par exemple des présentations de CaRMetal [comme celle mise en ligne par l'académie de Poitier](#), cela me fait bien évidemment plaisir, et je ne cacherai pas qu'une petite reconnaissance du travail effectué, par ma hiérarchie toulousaine, me paraîtrait bien sympathique. Développer un logiciel comme celui-ci prend beaucoup de temps, et le temps n'est pas extensible... Pour la petite histoire, en dehors de mes 19h00 de cours hebdomadaire, il m'a été reproché l'an dernier de ne pas avoir voulu m'investir et participer à la vie informatique de mon établissement, comme personne ressource. Si l'absence de reconnaissance de ma hiérarchie locale n'est pour moi qu'un petit regret, ce genre de reproche-là, par contre, fait plutôt du mal.

Hors institution par contre, l'encouragement est très présent, avec des articles récents comme celui [publié sur le portail des IREMs](#) (Institut de Recherche sur l'Enseignement des Mathématiques), ou par exemple les [travaux réalisés par l'IREM de La Réunion](#). L'[IREM de Toulouse m'a contacté très tôt](#), en 2007, pour savoir si je voulais intervenir dans ses stages

« Enseigner les maths au collège » et c'est avec grand plaisir que j'ai accepté, et que je donne dans ce cadre-là quelques heures de formation par an. Tout ce qui est écrit aussi au sujet de CaRMetal dans le monde du logiciel libre me donne bien sûr l'envie de continuer !

Que pensez-vous du label RIP et de l'initiative SIALLE visant à promouvoir le logiciel libre dans l'éducation ?

Sans même parler des critères d'attribution de ce label [RIP](#) (Reconnu d'Intérêt Pédagogique), on peut se demander si les enseignants ont vraiment besoin qu'on leur indique quelles ressources sont *intéressantes* pour eux, avec comme corollaire possible que les autres ne le sont pas. Une ressource pédagogique n'étant pas tout à fait comparable à un poulet sous cellophane, j'ai toujours été assez réservé sur la nécessité de labelliser quoi que ce soit dans notre métier d'enseignant. Pour bénéficier de ce label, il faut le demander, et ce n'est peut-être pas dans la culture de tous les développeurs libres du monde éducatif. Cependant, je n'irai pas jusqu'à affirmer que c'est pour cela que presque aucun logiciel libre n'est classé RIP !

Je n'ai jamais demandé le label RIP pour CaRMetal, qui ne fait donc pas partie de la liste : on constate par contre que cela convient tout à fait aux sociétés privées de e-learning qui cherchent à se faire estampiller leurs animations flash, et qui, par les temps qui courent, y arrivent plutôt bien.

Je vois par contre d'un tout autre oeil [l'initiative SIALLE](#), qui montre que l'idée du libre fait aussi son chemin dans l'Education Nationale, même si tout cela reste très timide. C'est pour cette raison-là que lorsqu'on m'a contacté il y a deux ans pour faire partie de cette base de logiciels, j'ai donné mon accord immédiatement. Je regrette un peu l'aspect rigide de la base de données, avec des classements par étoiles qui ne sont issus en général que d'un ou deux avis, mais il

reste que cette initiative a le mérite d'exister et qu'il faut l'encourager, et faire en sorte qu'elle prenne plus d'importance.

En mathématiques, il y a l'incontournable et très libre association Sésamath. Participez-vous, de près ou de loin, à ses travaux ?

Non, et c'est un de mes grands regrets... J'ai participé il y a trois ans, avec l'équipe de développeurs de mathenpoche, à une réunion qui s'était fixée pour objectif de jeter les bases du Labomep à venir. J'aurai dû continuer à collaborer sur ce projet avec ces collègues sympas et talentueux, mais je suis resté "en solo" sur CaRMetal, en ne traitant que l'infinie liste des choses à faire pour ce logiciel... On n'a qu'une vie et c'est dommage ! Je suis par contre, dans ma pratique pédagogique, un utilisateur très assidu de toutes les productions de cette association : les manuels, les démos, Mathenpoche, Sesaprofs, eBep's etc.

Sésamath est un formidable ballon d'oxygène au milieu d'un monde de brutes. Je ne prendrai qu'un seul exemple, que je souhaite détailler : celui des manuels numériques. A l'occasion des nouveaux programmes de sixième estampillés « manuels numériques », les enseignants ont reçu en fin d'année dernière de nombreux spécimens, avec du Bordas, du Nathan, du Hachette et du Hatier comme s'il en pleuvait. Rien de très nouveau dans cette avalanche de papier glacé, mise à part la possibilité de vidéo-projeter le manuel. Il n'y a souvent aucune interactivité véritable dans les documents proposés : juste du texte, des images, et l'idée véhiculée qu'avec tout ça, on allait enfin pouvoir se sentir *dans le vent*.

Les choses se compliquent lorsque l'enseignant veut installer pour lui et pour sa classe les manuels numériques en question :

- On commence par la case tiroir-caisse avec en moyenne 5,50€ par élève, si on a bien sûr déjà payé les 20€ par élève du manuel papier. Ceux qui souhaitent étudier de près les grilles de tarification des éditeurs, peuvent cliquer [ici](#) pour visualiser l'exemple de l'offre de la maison Hachette.
- On part sur un site portail (KNE) pour télécharger son manuel, avec les codes d'accès achetés par l'établissement.
- On installe une application nommée delivery (logiciel utilisé pour les ventes de magazines en ligne) qui a pour fonction de télécharger, lire, et cadenasser la ressource par l'utilisation d'un format propriétaire (.dly).

Pour un enseignant donné, le manuel ne s'installe qu'une seule fois, avec à la limite une autre installation possible sur une clé USB. Si cela vous rappelle quelque chose, c'est normal : il s'agit bien de [DRMs](#) appliquées à l'Education Nationale. J'allais oublier de signaler que l'installation de ces manuels numériques, quand on y arrive (de nombreux collègues rencontrent de grosses difficultés techniques pendant cette phase), ne peut se faire que sur windows ou mac. Amis linuxiens, bonjour. Les collègues qui ont opté pour le manuel sesamath 6e (mais aussi les autres !) peuvent aller librement sur le site, téléchargent sans code d'accès ce qu'ils veulent, d'où ils veulent, sous format pdf ou OpenOffice.org, le modifient comme ils le souhaitent, utilisent les démos interactives des chapitres, le site Mathenpoche, ainsi que tous les formidables outils produits par cette association.

Sésamath, c'est le choix de la pertinence et de la sérénité, très loin des contraintes, des contorsions techniques et des interdits que veulent nous imposer les marchands du temple par le biais de leurs relais institutionnels.

La situation du logiciel libre à l'école : excellente, bonne,

peut mieux faire ?

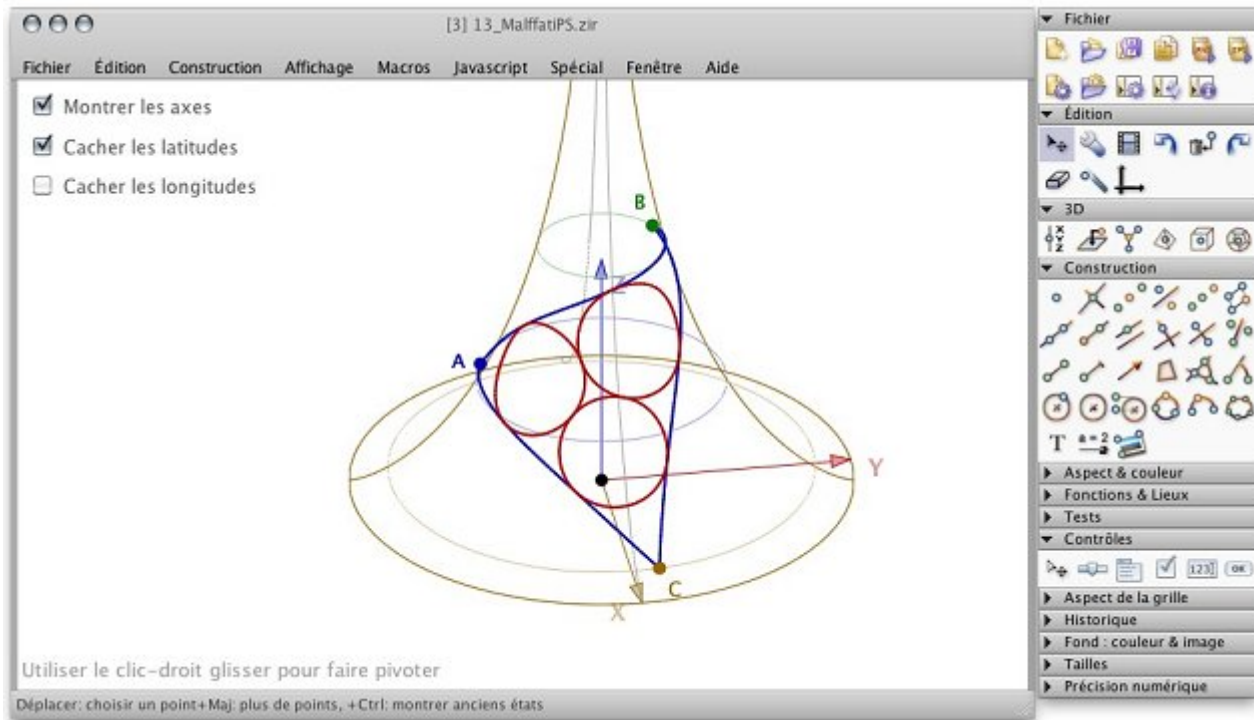
Peut beaucoup mieux faire bien entendu... J'attends impatiemment que l'institution se mette enfin à étudier sérieusement les alternatives libres, mais avec des exemples récents comme celui des manuels numériques, on peut se dire qu'il y a encore beaucoup de travail à faire de ce côté-là.

Il serait pourtant possible d'imaginer des établissements entièrement équipés de clients linux sur les réseaux pédagogiques : l'offre logicielle, ainsi que la simplicité d'installation et d'utilisation des nouvelles distributions le permettrait dès maintenant. Il manque juste, pour se faire, une évolution des mentalités des acteurs du monde éducatif, et je ne parle pas que de l'institution : beaucoup de collègues, qui ne se sentent pas en confiance avec l'outil informatique, sont encore très attachés au système windows qui leur a été livré en standard avec leur machine. Je peux comprendre cela et je ne critique pas : je pense juste que la situation du logiciel libre à l'école va sûrement évoluer, mais pas aussi rapidement qu'on pourrait le croire...

Que manque-t-il à CarMetal selon vous et quels sont les améliorations que vous aimeriez voir figurer dans les prochaines versions ?

CaRMetal est un logiciel de géométrie plane qui simule assez bien la 3D, mais il lui manque certaines fonctionnalités pour qu'on puisse directement construire et manipuler des objets dans l'espace : c'est sur cette question-là que je travaille en ce moment, et une version qui véritablement 3D devrait voir le jour d'ici décembre.

Une autre amélioration en vue : le multi-fenêtrage de l'application devrait être remplacé, dans un avenir proche, par une navigation par onglets qui permettra un accès plus simple aux figures.



Ou toute autre version ultérieure de la licence...

Ce billet est un peu technique, au sens juridique du terme, mais la problématique abordée est loin d'être inintéressante (et n'étant pas des spécialistes comme [Veni Vidi Libri](#), nous vous remercions par avance des précisions, compléments et corrections que vous apporterez dans les commentaires).



Il évoque les risques potentiels encourus par une création (logicielle, culturelle, etc.) à adopter une licence portant la mention « ou toute autre version ultérieure » (de la-dite licence). Dit autrement, il s'agit de la confiance que vous,

auteurs ou utilisateurs, accordez à ceux qui rédigent la licence choisie, puisque vous vous engagez ainsi non seulement au moment présent mais également dans le futur, avec une licence susceptible d'être modifiée au cours du temps. Ce que vous créez ou utilisez aujourd'hui peut donc ne pas suivre exactement les mêmes règles demain.

Prenons l'exemple de la plus célèbre des [licences de logiciels libres](#), la [GNU General Public License](#) (ou GNU GPL). Voici ce que l'on peut lire dans la [traduction non officielle](#) du paragraphe 9 de la [version 2](#) datant de 1991 :

La Free Software Foundation se réserve le droit de publier périodiquement des mises à jour ou de nouvelles versions de la Licence. Rédigées dans le même esprit que la présente version, elles seront cependant susceptibles d'en modifier certains détails à mesure que de nouveaux problèmes se font jour.

Chaque version possède un numéro distinct. Si le Programme précise un numéro de version de cette Licence et « toute version ultérieure », vous avez le choix de suivre les termes et conditions de cette version ou de toute autre version plus récente publiée par la Free Software Foundation. Si le Programme ne spécifie aucun numéro de version, Vous pouvez alors choisir l'une quelconque des versions publiées par la Free Software Foundation.

Donc si vous placez votre logiciel sous GNU GPL sans préciser la version ou en version 2 avec la mention « ou toute version ultérieure », alors « vous avez le choix de suivre les termes et conditions de cette version ou de toute autre version plus récente publiée par la Free Software Foundation », version plus récente dont on nous assure, et c'est là que ça peut coïncider, qu'elle sera « rédigée dans le même esprit ».

Or, justement, dans l'intervalle, la GNU GPL [est passée en version 3](#) en 2007 (sa version actuelle). Extrait de la FAQ de

cette nouvelle version, à la question [Pourquoi les programmes doivent-ils se référer à « la version 3 de la GPL ou toute version ultérieure » ?](#) :

De temps en temps, après quelques années, il nous arrive de modifier la GPL—quelquefois simplement pour éclaircir un point, quelquefois pour autoriser certaines utilisations alors qu'elles ne l'étaient pas auparavant, et quelquefois pour renforcer une exigence. (Les deux derniers changements datent de 2007 et 1991). L'utilisation de ce « pointeur indirect » dans chaque programme nous permet de changer les conditions de distribution de l'ensemble des logiciels GNU, lorsque nous modifions la GPL.

Si aucun programme ne contenait ce pointeur indirect, nous serions forcés de longuement discuter du changement avec de très nombreux détenteurs de copyright, ce qui serait virtuellement impossible. En pratique, la probabilité d'aboutir à un mode de distribution unifié pour les logiciels GNU serait nulle.

Supposez qu'un programme dise « Version 3 de la GPL ou toute version ultérieure » et qu'une nouvelle version de la GPL soit publiée. Si la nouvelle version de la GPL donne une permission supplémentaire, cette permission est immédiatement disponible pour tous les utilisateurs du programme. Mais, si la nouvelle version de la GPL comporte une exigence plus restrictive, cela ne limitera pas l'utilisation de la version courante du programme, car ce dernier peut toujours être utilisé sous GPL Version 3. Lorsqu'un programme dit « Version 3 de la GPL ou toute version ultérieure », les utilisateurs seront toujours autorisés à l'utiliser, et même à le modifier, selon les termes de la GPL Version 3—même après que d'autres versions de la GPL auront été rendues disponibles.

Si une exigence plus forte dans une nouvelle version de la GPL n'est pas obligatoire pour les logiciels existants, à quoi sert-elle ? Lorsque la version 4 de la GPL est

disponible, les développeurs de la plupart des programmes sous GPL distribueront les versions suivantes de leur programme en spécifiant « Version 4 de la GPL ou toute version ultérieure ». Les utilisateurs devront se conformer aux clauses plus restrictives de la GPL Version 4, pour les versions suivantes de ce programme.

Les développeurs ne sont toutefois pas dans l'obligation d'agir ainsi ; ils peuvent continuer à autoriser l'utilisation de la version précédente de la GPL, si c'est leur souhait.

Le fait qu'une licence évolue n'est en rien critiquable puisque depuis 1991 il s'en est passé des choses dans le monde numérique (à commencer par l'avènement d'Internet). De plus cette nouvelle version a été [le fruit d'un long processus de concertation publique](#) (voir la lettre de Chris Frey plus bas), la FSF ayant le souci de donner confiance et ne pas donner l'impression qu'on lui signait un chèque en blanc pour les « versions ultérieures » de ses licences.

Il n'empêche que ces modifications « rédigées dans le même esprit » sont laissées à la libre appréciation de chacun. C'est ainsi que Linus Torvalds, conformément au dernier paragraphe de notre extrait ci-dessus, n'a toujours pas passé le [noyau Linux](#) en GPL v3 et [préfère rester en GPL v2](#), au grand dam de la FSF.

Mais prenons désormais l'exemple d'une autre licence de la FSF, la [GNU Free Documentation License](#) (ou GNU FDL). À l'origine cette licence avait été pensée pour la documentation de logiciels libres mais elle a été adoptée, faute d'existence d'autres licences mieux adaptées à l'époque, par le *poids lourd de la culture libre* que constitue l'encyclopédie Wikipédia.

Or, depuis, une licence clairement mieux adaptée est apparue :

la licence Creative Commons By-Sa^[1]. Ces deux licences n'étant pas compatibles, que pouvait-on faire ? C'est justement la réponse à cette question que nous avons évoqué dans notre récent billet dédié : [Wikipédia en route vers la licence Creative Commons By-Sa](#). Ce billet explique pourquoi et comment la FSF a accepté de modifier sa licence GNU FDL (version intermédiaire 1.3) pour que Wikipédia puisse légalement passer sous Creative Commons, passage justement rendu possible parce que la licence de Wikipédia est la GNU FDL sans mention de version (acceptant donc implicitement « toute version ultérieure » de la licence).

Le problème c'est que contrairement au passage de la GNU GPL de la version 2 à la version 3, il n'y a pas eu de concertation publique et de long processus de maturation transparent et « démocratique ». De plus les modifications apportées ont clairement été rédigées dans l'unique but de satisfaire la [Wikimedia Foundation](#) (et Creative Commons) puisqu'elles ne s'adressent qu'aux wikis créés avant novembre 2008 qui ont jusqu'en juin 2009 pour se décider à migrer. Une mise à jour « sur mesure » en quelque sorte.

Aussi nobles soient les intentions des uns et des autres, il y a tout de même ici un peu d'arbitraire et d'opacité aussi bien sur le fond que dans la forme. C'est ce que critique Chris Frey dans cette « lettre ouverte à Richard Stallman » que nous avons choisi de traduire et vous proposer ci-dessous, accompagnée par la réponse toute aussi ouverte de Richard Stallman.

Lettre ouverte à Richard Stallman

[Open Letter to Richard Stallman](#)

Chris Frey – 4 novembre 2008 – Advogato.org

(Traduction Framalang : Olivier, Siltaar et Don Rico)

Cher M. Stallman,

Je vous écris pour exprimer la déception que j'ai ressentie à l'égard de la Free Software Foundation par rapport à la publication récente de la version 1.3 de la GNU Free Documentation License .

Cette nouvelle version introduit à la section 11 une nouvelle clause qui, selon la FAQ, accorde aux wikis le droit de modifier la licence de certains contenus pour les faire passer de la licence GFDL 1.3 à la licence CC-BY-SA 3.0, pour les ajouts datant d'avant le 1er novembre 2008. Ils peuvent appliquer ce changement de licence jusqu'au 1er août 2009.

Cela constitue à mon sens une erreur morale importante et un abus de confiance. Même si cette nouvelle clause n'est pas dangereuse, elle n'en reste pas moins mauvaise.

Il y a de nombreuses années, j'ai lu des critiques visant les recommandations quant à l'application de la licence GNU GPL aux logiciels. On critiquait la formule « version 2 of the License, or (at your option) any later version » (*NdT : version 2 de la licence, ou, selon votre choix, toute version ultérieure*) avançant l'argument que cette formulation conférait trop de pouvoir à la FSF, et que cela revenait à signer un chèque en blanc permettant le détournement futur de l'œuvre licenciée.

Ces critiques ont à mes yeux perdu de leur sens après que j'ai été témoin de la transparence et de l'attention extrême portée à la rédaction de la nouvelle GPL version 3. Tous les commentaires furent enregistrés sur un site Web et débattus de manière tout à fait transparente. La rédaction a été longue et méticuleuse, réalisée en prenant en compte l'opinion des utilisateurs de la licence, et dans l'objectif de servir l'intérêt des logiciels libres. C'était magnifique.

Je commençais à comprendre les avantages d'une licence ouverte. J'avais compris les intentions de la FSF. Ils ne fléchissaient pas concernant les principes des logiciels

libres, mais au contraire se montraient toujours plus fermes. Seule la FSF pouvant modifier la GPL, et étant donné le sérieux de sa méthode pour la mettre à jour, je me suis dit que je pouvais lui accorder ma confiance.

Hélas, ce changement apparemment précipité apporté à la GFDL vient tout gâcher. Cette modification va à l'encontre de la clause tacite selon laquelle c'est l'auteur qui choisit la licence pour son travail, mais elle introduit en outre un nouvel acteur. Non seulement la FSF peut changer les termes de la GFDL, mais voilà que la Creative Commons Corporation peut à présent modifier les termes qui régissent des travaux plus anciens auparavant placés sous GFDL.

Fort heureusement, cette situation ne durera que jusqu'au 1er août 2009. Mais les anciens wikis placés sous GFDL ont désormais une double licence.

Peut-être ai-je vécu coupé du monde légal, mais je n'ai pas non plus vu d'appel à contribution pour cette révision. Si je l'ai manqué, je regrette de n'avoir pu donner mon avis plus tôt. Si la période de discussion était plus courte, voire inexistante, c'est une faille dans le processus de mise à jour de la GFDL, et j'espère que la FSF y remédiera.

Concrètement, cette requalification de licence importe peu. Le cadre de ce qui peut être soumis à une requalification de licence est bien délimité, et ces deux licences ont en gros la même fonction. Certains auteurs ont peut-être préféré la GFDL à la licence CC-BY-SA pour des raisons spécifiques, l'une d'entre elles étant le fait que la licence doit être jointe à l'œuvre. Cette condition de la GFDL est à la fois une caractéristique et un défaut de conception. Mais ce choix, qui auparavant revenait à l'auteur, lui est à présent retiré.

Au final, les conséquences sont sûrement minimales. En effet, beaucoup voient en ce changement un grand pas en avant. Je pense que c'est simplement la réputation de la FSF qui s'en

trouve affectée. Quel message cela adresse-t-il aux utilisateurs d'autres licences de la FSF ? Est-on sûr que la FSF ne permettra pas la modification des licences de nos œuvres au profit de licences non-FSF si l'on opte pour la clause « version ultérieure » ? Dans le passé je ne me serais même pas posé la question. À présent, j'ai un doute.

J'ai hâte de connaître votre sentiment à ce propos et, si vous le permettez, je publierai une copie conforme de votre réponse là où je publie cette lettre.

Je vous remercie de votre attention.

Cordialement,

Chris Frey

Réponse ouverte à Chris Frey à propos de la GFDL 1.3

[An open response to Chris Frey regarding GFDL 1.3](#)

Richard Stallman – 3 décembre 2008 – FSF.org

(Traduction Framalang : Olivier, Siltaar et Don Rico)

Cher M. Frey,

Votre lettre concernant les modifications récentes apportées à la GFDL, qui permettent aux administrateurs de certains wikis sous licences GFDL de modifier la licence de leur contenu sous licence Creative Commons BY-SA, soulève des questions cruciales vis-à-vis des modifications, de la manière dont elles ont été conduites et de leurs implications quant à la gestion par la FSF de nos licences à l'avenir.

De mon point de vue, et c'est aussi celui du conseil de la FSF, cette modification de la licence est totalement en accord avec nos valeurs, notre éthique et nos engagements, et c'est une nouvelle démonstration que la FSF mérite toujours votre

confiance. Les licences comportant la clause « ou toute version ultérieure » nous permettent de donner de nouvelles permissions qui répondent aux besoins de la communauté et de contrer les nouvelles menaces qui pèsent contre la liberté des utilisateurs.

Cette option de changement de licence introduite par la GFDL 1.3 est complètement en accord avec l'esprit et les objectifs de la GFDL. Certains sites Web pourront ainsi passer de la GFDL à une autre licence libre, une autre licence qui diffère sous certains aspects mais qui est très proche globalement de la GFDL. Cette disposition permet à ces sites de rendre leur licence compatible avec de vastes collections de contenu sous licence libre avec lesquelles ils veulent coopérer.

L'impact de ce changement est limité car l'option de changement de licence ne s'applique qu'à un nombre de cas limité : les wikis, telle Wikipédia, qui ne font pas usage de certaines clauses particulières de la GFDL, comme les sections invariantes (*NdT : contenu qui n'est modifiable que par l'auteur lui-même*) et les textes de couverture, qui n'ont pas d'équivalent dans la licence CC-BY-SA.

Nous avons offert aux wikis une option à durée limitée dans le temps leur permettant de passer sous CC-BY-SA les contenus issus de la contribution du public. La fondation Wikimedia a initié un débat public quant à la décision à prendre : les participants de Wikipédia peuvent donner leur avis pour ou contre le changement. Nous n'avons joué aucun rôle dans cette décision, nous avons simplement donné à Wikipédia l'opportunité de la prendre.

Ce changement ne s'est pas fait dans la précipitation. La FSF s'est entretenue pendant plus d'un an avec la Wikimedia Foundation, Creative Commons et le Software Freedom Law Center pour planifier ce changement.

Ces discussions sont restées secrètes pour une raison que, je

l'espère, vous approuverez : limiter l'applicabilité de cette option et éviter l'éventualité d'un changement de licence massif d'autres contenus placés sous licence GFDL. Cette option a été pensée uniquement pour les wikis bénéficiant d'une collaboration publique. Nous ne souhaitons pas que tout le monde puisse changer la licence des œuvres couvertes par la GFDL en les « blanchissant » au travers de Wikipédia ou d'autres sites similaires avant la date butoir.

Si un wiki exerce son option de changement de licence, cela implique qu'il accorde sa confiance aux Creative Commons plutôt qu'à la Free Software Foundation pour ce qui est des modifications futures de la licence. En théorie, on pourrait y voir une source d'inquiétude, mais je pense que nous pouvons faire confiance à Creative Commons pour continuer sa mission à l'occasion des mises à jours de ses licences. Des millions d'utilisateurs font déjà confiance aux Creative Commons à ce sujet et je pense que nous pouvons faire de même.

Vous avez qualifié notre modification de licence de « trahison », mais si vous examinez ce que nous avons entrepris, vous verrez que nous sommes restés fidèles à nos principes.

Nous n'avons jamais affirmé que nous ne changerions pas nos licences, ou que nous ne ferions pas de modifications comme celle-ci. Nous nous sommes en revanche engagés à effectuer ces changements en respectant l'esprit de nos licences, et en garantissant les objectifs de ces licences. C'est ce que nous avons fait avec la GFDL 1.3 et, à plus grande échelle, avec la GPLv3, et c'est ce que nous continuerons à faire.

Vous avez entièrement raison d'attendre de la FSF une éthique irréprochable. J'espère que les modifications apportées à la licence GFDL vous convaincront que nous respectons cette éthique.

Cordialement,

Richard Stallman

Président de la Free Software Foundation

Notes

[1] Crédit photo : [Cambodia4kidsorg](#) (Creative Commons By)

Comment Eben Moglen a rencontré Richard Stallman

Tranche d'Histoire du logiciel libre...

Aux premiers temps de l'informatique, (presque) tous les logiciels étaient libres (sans même le savoir). Puis vient le temps de la propriétérisation du code qui obligea certains, comme [Richard Stallman](#), à protéger la liberté des logiciels ou plus précisément la liberté des utilisateurs de logiciels.

Cette protection serait d'autant plus forte qu'elle serait sans faille vis-à-vis de la loi. Et c'est ainsi que le monde des [hackers](#) fit connaissance avec celui des juristes pour enfanter de licences qui font bien plus qu'accompagner les logiciels libres puisqu'elles participent à leur définition même.

Or, l'une des rencontres les plus fécondes entre le juridique et l'informatique (libre) est très certainement celle d'[Eben Moglen](#) avec Richard Stallman. Et c'est pourquoi nous avons jugé intéressant de traduire^[1] la retranscription d'une interview qu'Eben Morglen a donné à Joe Barr de [Linux.com](#) en juin dernier pour en sous-titrer la vidéo^[2].

« Nous sommes des nains juchés sur les épaules des géants.

Stallman était un géant, je me suis juché sur ses épaules et j'ai vu le monde. »

Vous trouverez la version au format libre Ogg de l'interview sur [Linux.com](#). Si vous souhaitez lui ajouter le sous-titrage en voici le [fichier SRT](#).

[Eben Moglen: How I discovered Free Software and met RMS \(video\)](#)

Comme Stallman, John Gilmore et d'autres de ma génération, on peut dire que j'ai été impliqué dans le logiciel libre quand j'étais enfant car les logiciels étaient libres pour eux. J'ai commencé à 14 ans comme développeur d'applications APL pour Scientific Time Sharing Corporation (STSC) en 1973. J'ai travaillé à la conception et à la mise en oeuvre d'applications APL pour STSC et pour Xerox dans les années 70 et, en 79, j'ai été travailler pour IBM au laboratoire de Santa Theresa où j'ai modifié les interpréteurs APL pour IBM. J'ai travaillé sur l'APL et APL2. J'ai écrit une bonne partie du premier compilateur pascal d'IBM.

La manière dont nous travaillions était basée, après tout, sur le partage du code avec les personnes (clients) qui utilisaient les ordinateurs sur le terrain. Ils nous aidaient à concevoir, mettre en œuvre, améliorer et modifier les choses (code). Ils avaient les sources et quand ils émettaient un APAR (*NdT : Authorized Program Analysis Report, dans la nomenclature d'IBM ce terme désigne un problème officiellement reconnu et diagnostiqué par le centre de support IBM*), ils envoyaient un patch.

Ainsi, dans un sens, nous vivions dans un environnement où le logiciel était encore libre. Bien sur, nous avons des principes de propriété mais, en 1979, quand la commission "CONTU" terminait sa réflexion sur le logiciel libre, ces principes de propriété n'étaient pas encore bien compris et

dépendaient à la fois de tout le monde et de personne.

Pouvait-on y attacher une propriété intellectuelle? AT&T et IBM n'étaient pas d'accord. Quelle partie du code pouvait-on protéger par un brevet ? Quasi rien. Pouvait-on le considérer comme un secret industriel ? Et bien non, ce code permettait seulement de différencier des ordinateurs onéreux.

Ainsi, d'une certaine manière, le monde dans lequel nous vivions présupposait une liberté à bricoler (le code). J'ai principalement travaillé sur des langages interprétés où les codes source et objet sont confondus. Il fallait partager le code. J'ai travaillé sur des produits distribués sur des environnements 370 (*NdT : IBM mainframe*) où le client s'attendait à recevoir le code source et s'il avait le code source de VM (*NdT : OS*) et que quelqu'un lui fournissait du MVS (*NdT : OS*) dans un langage appelé PLS pour lequel il n'avait pas de compilateur, il se plaignait ; parce qu'il s'attendait à pouvoir compiler le produit sur le site. Il s'agissait tout de même d'un super-ordinateur de plusieurs millions de dollars, qui aurait osé lui dire qu'il n'avait pas le droit de compiler le logiciel sur sa machine ?

D'une certaine façon, je dirais que j'ai toujours vécu dans le logiciel Libre. Ce toujours a néanmoins connu une pause.

Alors que je travaillais chez IBM en 1979, on m'a demandé de tester et de faire un rapport interne sur un machin nommé LISA ; le dernier gadget de Apple pour faire entrer la technologie de Xerox PARC (*NdT: Palo Alto Research Center*) dans le monde de Steve Jobs. Le LISA était une sorte d'ordinateur Pre-Macintosh ; j'ai donc écrit mon rapport interne sur cette machine et le contenu de ce rapport était le suivant : C'est une catastrophe. Cette machine incarne la fin du langage en relation avec l'ordinateur, c'est l'interface de l'homme des cavernes : tu vises et tu grognes. Si on résume l'interaction homme machine à viser et grogner, on écarte le rôle du langage dans l'évolution de l'esprit humain et de sa

conscience. Le langage est ce qui nous rend plus intelligent, si nous n'utilisons pas le langage pour communiquer avec les machines alors ni nos cerveaux ni ceux des machines ne s'épanouiront comme ils devraient le faire. Cet argument eu peu de poids chez IBM et il en eu encore moins dans le monde en général au fil du temps.

Je devins moins enthousiaste devant la perspective de programmer dans ce monde car j'étais mordu de langage de programmation et le langage n'était plus ce qui était en vogue. Je suis donc parti et j'ai obtenu une licence de droit et un Doctorat en histoire et je suis devenu Historien du droit et j'ai fait d'autres travaux. J'ai débuté avec un emploi en tant qu'assistant Juge (*NdT: Law Clerk*) pour Weinfeld à NY, j'ai été assistant de Thurgood Marshal, puis j'ai commencé à me demander de quelle manière on pouvait rendre le monde plus juste !

J'ai eu ensuite un boulot d'enseignant dans une excellente Université en tant que Historien du droit. J'y ai fait ma thèse de Doctorat et écrit quelques articles d'histoire. Je m'intéressais à la signification à long terme de l'information dans la société humaine. Puis j'ai voulu coder car je suis un codeur compulsif et aussi parce que j'avais un PC à ma disposition qui, certes n'était pas la machine de onze millions de dollars à laquelle j'étais habitué, mais c'était un ordinateur qui pouvait servir à deux trois petites choses. Je n'aimais pas beaucoup DOS mais je n'ai jamais utilisé Windows qui était La Chose mauvaise pour les ordinateurs. Je n'allais pas utiliser quelque chose que je considérais comme la pire des choses. Je savais ce que X windows était, mais qui désirait utiliser ça, vous savez... le cerveau etc... Je suis donc passé chez Coherent lorsque la compagnie de Mark Williams créa un Unix estropié à 99\$. Je l'ai essayé et j'ai commencé à l'utiliser avec les Outils du projet GNU puis ensuite j'ai utilisé les outils GNU sous DOS. J'utilisais DJGPP, puis le compilateur C de Delorie pour porter UNIX sur le DOS afin

d'utiliser EMACS sur ma machine DOS car tout comme Stallman j'avais une grande dévotion pour EMACS.

Donc quelque part le logiciel libre a toujours été présent mais l'essentiel de ma vie était non technologique. En 1991 je décidais que je savais ce qu'il fallait faire pour commencer à travailler pour la liberté au 21 siècle: La cryptographie à clé publique était la première chose à implémenter. Nous en avions besoin pour deux raisons: garder les secrets à l'abri du gouvernement et faire du commerce électronique. Donc j'ai commencé à m'intéresser à la question, à chercher un moyen de faire de la cryptographie pour casser les règles du gouvernement sur le chiffrement de données. En Juillet 1991 j'ai vu un programme appelé Pretty Good Privacy (PGP) publié sur un forum. J'ai récupéré l'archive zip, j'ai lu le manuel de l'utilisateur et j'ai lu le code source car celui-ci était fourni, puis j'ai écrit un email non sollicité à l'auteur Phil Zimmerman qui n'avait jamais entendu parler de moi et je lui ai dit: « Bravo, tu vas changer le monde. Tu vas aussi ne pas tarder à être dans un merde noire, lorsque ça te sera tombé dessus je pourrais t'aider. Voici qui je suis, voilà ce que j'ai fait et voici ce que je sais, quand tu auras des soucis appelle moi. » Dix jours plus tard le FBI frappait à sa porte et les ennuis commencèrent.

Donc j'ai décidé de travailler comme bénévole dans un groupe local de défense ; nous étions quelques personnes à prendre sur notre temps libre pour essayer d'empêcher le gouvernement fédéral d'accuser Zimmerman de violation de la loi sur le trafic d'armes. Notre but était d'aller aussi loin que possible avec cette affaire jusqu'au cœelig;ur des choses, afin de mettre les contradictions au grand jour. Alors que je travaillais sur l'affaire Zimmerman j'ai passé du temps avec John Markov du Times et au cours d'une interview je lui ai exposé mon idée selon laquelle le droit de parler le PGP (communiquer avec PGP) était le pendant numérique du droit de parler le Navajo. Markov fit paraître ceci dans le Times et ça

devint une maxime que de nombreuses personnes utilisèrent comme signature dans leurs emails pendant quelques mois. Stallman vit la couverture de Markov sur le Times et il m'écrivit. Il me dit « j'ai un problème légal/juridique personnel et j'ai besoin d'aide; il me semble que tu es l'homme qu'il me faut. » Je lui ai répondu: « J'utilise Emacs tous les jours et il faudra du temps pour que tu épuises ton crédit d'aide juridique gratuite!.» il m'a demandé de le faire et j'ai fait ce qu'il fallait faire pour lui.

J'ai réalisé qu'il était la source même d'informations sur ce que je devais faire. J'avais fait ce qu'il me semblait important de faire au sujet de la cryptographie et je voyais que le problème était sur le point d'être réglé, mais je ne savais pas quoi faire ensuite pour apporter la Liberté technologique au 21ème siècle. C'était en automne 1993 et j'ai réalisé que toute personne qui avait un souci concernant la Liberté Technologique ne connaissait qu'une seule adresse mail : rms AT gnu.org. Si RMS me transférait tout ses messages nécessitant l'intervention d'un juriste, je serais assez rapidement mis au courant de ce qu'il y avait à faire en ce bas monde. RMS avait la meilleure prospective stratégique qui soit.

Je me suis donc assis sur ses épaules pour quelques années, faisant tout le travail qu'il considérait comme important et me tenant au courant de tout ce que les gens lui écrivaient. A la fin, je lui ai dit « tu as besoin d'un conseiller juridique » et il a dit « bien sûr ! » et j'ai commencé à faire le travail qu'il y avait à faire. C'était juste du travail que je faisais sur mon temps libre, j'étais toujours un historien du droit, personne parmi mes collègues académiques n'avait la moindre idée de ce dont il s'agissait, tout le monde savait que je racontais que le logiciel libre allait conquérir le monde et il me répétaient « oui oui, c'est formidable, super, merci beaucoup, à bientôt » et ça en restait là. Mais je savais où nous allions et surtout j'avais

compris que Stallman en personne était la plus haute des montagnes et qu'en étant assis sur ses épaules on voyait bien plus loin.

Newton et consorts jusqu'à Bernard de Chartres avaient raison. Nous sommes des nains juchés sur les épaules des géants et c'est ce qui s'est passé : Stallman était un géant, je me suis juché sur ses épaules et j'ai vu le monde. Donc d'une certaine manière on pourrait dire 1993, ou 1995, ce qu'on peut dire en tout cas c'est que j'en ai fait de plus en plus plus car il y avait de plus en plus de travail. Mais je ne pouvais pas me multiplier par neuf ! Et puis d'un coup l'espace temps à gonflé et on s'est tous rendu compte que c'était arrivé.

En gros, la réponse est que j'étais là avant le Big Bang et le temps n'existait pas encore. Beaucoup de personnes ont commencé à s'y référer seulement après que tout cela ait vraiment commencé. Mon point de vue est que tout cela entre dans la continuité de quelque chose qui a commencé il y a bien longtemps, c'est le renversement d'une singularité dans le déroulement du temps. Microsoft a, un temps, réussi à faire croire que le logiciel pouvait être un produit. Maintenant ce n'est que rarement un produit. L'information technologique précisant la façon dont nous et nos cerveaux numériques existent, ce n'est pas un produit, c'est une culture, c'est l'empreinte d'un être humain en interaction avec les autres. C'est comme la littérature, ça ne peut être un produit.

Donc, nous sommes en train de découvrir qu'il s'agit d'une culture engendrée par des communautés; nous aurions pu nous en rendre compte en 1965 où en 1970. C'était difficile à voir en 1990 mais c'est devenu évident (rires) en 2006. Pour moi il s'agit plus, d'un point de vue historique, de mettre un terme à une confusion temporaire plutôt que de parler d'un mystérieux et étrange point de départ qui aurait surgit d'on ne sait où.

Notes

[1] Grand merci à Ripat, Ziouplaboum et Olivier pour la traduction.

[2] Grand merci à Xavier Marchegay pour le sous-titrage.

L'iPhone n'est pas encore sous GPLv3, étonnant non ?

Il fallait être particulièrement rétif aux nouvelles technologies, genre mon père, pour ne pas avoir été au courant vendredi dernier de la sortie simultanée de l'[iPhone](#) (hou, hou, hou) et de [la version 3 de la licence GNU GPL](#) (clap, clap, clap).

L'occasion de traduire^[1] à la volée^[2] le corps de l'intéressant communiqué de la [FSF](#) qui concatène judicieusement les deux événements^[3].



L'iPhone enferme les utilisateurs, la

GPLv3 les libère.

[iPhone restricts users, GPLv3 frees them](#)

BOSTON, Massachusetts, USA-Jeudi 28 Juin 2007

Vendredi, 29 juin, tout le monde aux Etats-Unis n'attendra pas dans les files d'attente pour acheter un iPhone à 500\$. En fait, des centaines de milliers d'aficionados du numérique autour du globe ne seront pas du tout dans les files, car le 29 juin marque la sortie de la version 3 de la GNU General Public License (GPL). La version 2 de la GPL gouverne la plus grande part des logiciels libres qui est en train de remodeler radicalement l'industrie et de menacer le modèle technologique propriétaire représenté par l'iPhone.

L'auteur de la GPL est le Professeur Richard M. Stallman, président et fondateur de la Fondation pour le Logiciel Libre (Free Software Foundation ou FSF), et créateur du projet GNU. Avec sa première révision de la licence en seize ans, la version 3 de la GPL s'attaque aux plus récentes tentatives pour ôter la liberté du logiciel libre – et notamment, la version 3 attaque la "Tivoization" – et ceci pourrait être un problème pour Apple et son iPhone.

Maintenant, de la Chine à l'Inde, du Venezuela au Brésil, des Tivos aux téléphones cellulaires : le logiciel libre est partout, et il est en train de bâtir lentement un mouvement mondial d'utilisateurs demandant à avoir le contrôle sur leurs ordinateurs et les équipements électroniques qu'ils possèdent.

Tivoization et l'iPhone ?

La "Tivoization" est un terme forgé par la FSF pour décrire des équipements qui sont construits avec du logiciel libre, mais qui utilisent des mesures techniques qui empêchent l'utilisateur de faire des modifications au logiciel – une liberté fondamentale des utilisateurs du logiciel libre – et une attaque sur le logiciel libre que la GPLv3 pourra arrêter.

L'iPhone laisse les gens perplexes : contient-il du logiciel sous licence GPL ? Quel impact la GPLv3 aura sur les perspectives à long terme pour des équipements comme l'iPhone qui sont construits pour frustrer leurs propriétaires ?

Peter Brown, directeur exécutif de la FSF a dit, "Demain, Steve Jobs et Apple sortent un produit criblé de logiciel propriétaire et de restrictions numériques : criblé / invalide, parce qu'un équipement qui n'est pas sous le contrôle de son propriétaire, fonctionne contre les intérêts de son propriétaire. Nous savons qu' Apple a bâti son système d'exploitation, OS X, et son navigateur internet Safari, en utilisant des travaux couverts par la GPL – il serait intéressant de voir dans quelle mesure l'iPhone utilise du logiciel sous GPL.

La version 3 de la GNU GPL sera publiée à 12:00 pm (EDT) – six heures avant la sortie de l'iPhone – mettant un terme à dix-huit mois de discussions et de commentaires publics, en révision de la plus populaire des licences du logiciel libre.

Notes

[1] Pour la traduction *Framalang* (alias ici Vincent et Daria) a encore frappé.

[2] Traduction non officielle.

[3] L'illustration est issue du site DefectiveByDesign.org.

Richard Stallman en grande

forme (conférence à l'ENST le 3 avril 2007)

Que ce soit en direct dans la salle ou en différé sur le net, je commence à avoir pas mal de *conférences générales sur le logiciel libre* de [Richard Stallman](#) au compteur. Il faut dire que le bonhomme pour apprécier la France y revient souvent et s'exprime dans un français plus que correct (espèce étrangère en voie de disparition ?).

A priori on a l'impression d'assister toujours à la même conférence. Et celle que nous vous présentons en vidéo ci-dessous donnée mardi 3 avril 2007 dernier à l'[ENST](#) (École nationale supérieure des télécommunications) n'échappe à la règle. Chaussures ôtées et plus beau tee-shirt exhibé, on se retrouve invariablement avec la même entame (dont je ne me lasse toujours pas) : « *Je puis expliquer le logiciel libre en trois mots : liberté, égalité, fraternité...* » (sous vos applaudissements). Idem pour la conclusion du reste (dont je commence à me lasser) : « *...J'ai des autocollants, des pins et des porte-clé à vendre* ». Sacré Richard !

Et pourtant, variation sur le même thème, elles ont toutes un petit quelque chose qui les distingue des précédentes. Voici ce qu'écrivait récemment un spectateur de cette conférence sur la liste de diffusion de l'[APRIL](#). Je recopie d'autant plus volontiers cet extrait qu'il exprime bien ce que je voulais dire.

J'ai entendu RMS plusieurs fois depuis 1998. C'est toujours la même histoire mais elle est à chaque fois amendée, corrigée, complétée, modifiée, avec de nouvelles références et des éléments d'actualité...

Mardi soir il a présenté cette histoire sous un jour très nouveau et original, qui me semble intéressant pour

sensibiliser le grand public (...) et lui faire comprendre concrètement l'intérêt du LL pour lui. C'était construit, tout était utile/indispensable, la boucle était bouclée. Tout cela à partir des mêmes grandes lignes de réflexion remontant à 1983/4.

Vous voulez un exemple d'élément d'actualité de la conférence de l'ENST ? Point d'impatience, il suffit d'attendre... la deuxième phrase. Ce qui donne : « *Je puis expliquer le logiciel libre en trois mots : liberté, égalité, fraternité. Trois principes que le gouvernement actuel de la France ne respecte plus...* » No comment !

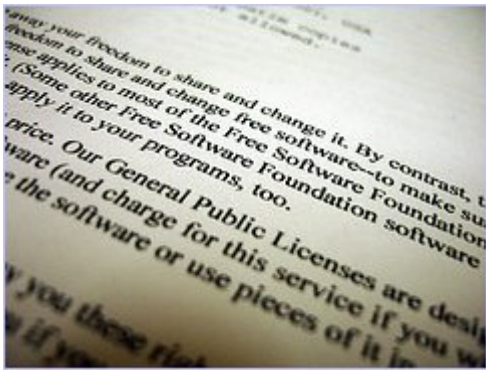
Toujours est-il qu'effectivement mardi soir et pour une heure environ c'était du grand Stallman (ou RMS pour les intimes). Définition et historique du logiciel libre (ou LL pour les intimes), différence avec le logiciel *privateur* (terme qu'il semble désormais privilégier à logiciel propriétaire), du danger des DRM et des brevets, différence entre son approche et celle de Linus Torvalds, différence entre *Logiciel Libre* et *Open Source* (quand l'un parle d'éthique et de liberté l'autre insiste plutôt sur rentabilité, efficacité et rentabilité), pourquoi il faut dire GNU/Linux et non Linux tout court, un point sur l'emploi, un autre sur l'éducation (et sa mission morale et sociale), un clin d'œil aux droits de l'homme... tout y était.

Pédagogie et vulgarisation de haute volée, une conférence que je vous invite à voir et à faire passer à vos proches qui seraient susceptibles d'être intéressés par un sujet qui dépasse en fait aujourd'hui le stricte cadre du logiciel libre pour ne pas être loin de proposer un véritable projet de société.



-> La [vidéo](#) au format webm

Stallman, Torvalds, Novell et la licence GPL v3



La plus célèbre et la plus diffusée de toutes les licences libres, la [GNU General Public License](#), est sur le point de sortir sa troisième version (la GPL v3). Ce sera un événement majeur dans la sphère plus si petite que cela du logiciel libre. Il faut dire que la version précédente, c'est-à-dire la version officielle actuelle pour quelques jours encore, remonte à 1991, Cette remarquable longévité témoigne de son efficacité et de sa stabilité mais il fallait bien lui apporter quelques révisions car dans l'intervalle, le développement d'internet, les [DRM](#), les [brevets logiciels](#), le cas exemplaire de [TiVo](#), [l'accord Novell Microsoft](#)... de l'eau avait coulé sous les ponts du monde numérique et la nouvelle donne impliquait aussi de nouvelles menaces.

Nous sommes désormais dans la dernière ligne droite d'un long processus de maturation discussion qui aura connu ses difficultés et ses controverses. Ainsi, en septembre 2006, lors de la mise en ligne du deuxième brouillon de cette troisième version, [Linus Torvalds en personne menaçait de ne pas faire passer le noyau Linux sous GPL v3 mais de rester sous GPL v2](#) ce qui aurait inévitablement engendré une crise au sein de la communauté.

Le troisième et dernier brouillon de la GPL v3 (faut suivre !) [vient de sortir](#) et, bonne nouvelle, il semble qu'il ait [plutôt rassuré Torvalds](#). Un article de [Linux.com](#) par Bruce Byfield fait le point en résumant les avis et positions de Richard Stallman (en quelque sorte le papa de la GNU GPL), de Linus Torvalds (en quelque sorte le papa de GNU/Linux) et d'un représentant de Novell (le papa de ses enfants).

Traduction (n'hésitez pas à en proposer des améliorations dans les commentaires).

[Stallman, Torvalds, and Novell comment on GPLv3](#)

Jeudi 29 mars 2007 – Bruce Byfield

Les commentaires à propos de la troisième version de la GNU General Public License (GPLv3), sortie hier, continuent d'affluer. Jusque là nous avons parlé avec le fondateur de la Free Software Foundation : Richard M. Stallman, le créateur de Linux : Linus Torvalds et le directeur des relations publiques de Novell : Bruce Lowry. Leurs réactions apportent de nouvelles perspectives et laissent entrevoir un possible premier pas vers un consensus. Ensemble, ils mettent en avant les points qui monopoliseront sûrement les discussions à propos de cette version dans les jours à venir.

Richard Stallman

Richard Stallman n'a pas pris part à la consultation autour de la GPLv3. A la place, il a quitté son poste de consultant au Software Freedom Law Center pour se concentrer sur les problèmes qui avaient été soulevés. Le processus, qui a duré presque deux ans pour lui, a représenté "pas mal de travail" dit-il. Cependant il ajoute "Je ne vois pas comment on aurait pu l'éviter. Beaucoup de problèmes que nous n'avions pas anticipés ont été soulevés".

Pour Stallman, la GPLv3 fait partie de l'évolution constante de la licence pour empêcher les tendances technologiques et légales de nuire aux principes du logiciel libre. "Avec la

GPLv1 je voyais deux moyens pour quelqu'un de rendre un logiciel libre vraiment propriétaire" dit-il. "L'un était par l'ajout de termes supplémentaires à la licence, l'autre moyen était de ne pas publier la source. Donc la GPLv1 a rendu ces deux choses impossibles. Ensuite, en 1990 j'en ai trouvé un autre : les détenteurs de brevet pouvaient faire pression sur les développeurs et leur imposer des conditions plus restrictives sur l'utilisation du logiciel. Alors la GPLv2 a ajouté la Section 7, selon laquelle soit vous distribuez votre produit avec toutes les libertés liées à la GPL soit vous ne le distribuez pas, peu important les conditions qui vous sont imposées. Actuellement nous avons découvert deux autres manières pour rendre un logiciel libre vraiment propriétaire : l'un d'entre eux est la TiVoisation (NdT : en rapport avec les enregistreurs numériques TiVo aux Etats-Unis) et l'autre est l'accord Novell-Microsoft, donc nous essayons de bloquer les deux. Et à chaque fois que nous découvrirons ce qui constitue une menace pour les libertés des utilisateurs nous essaierons de la bloquer."

Stallman dit que le dernier énoncé couvrant l'accord Novell-Microsoft a été achevé moins de cinq jours avant la sortie de la troisième version. L'astuce, dit-il, était d'écrire le langage dans la Section 11 afin de ne pas écarter de tels accords comme les licences partagées entre plusieurs sociétés qui incluent un engagement mutuel sur les brevets ou un arrangement avec ce qu'il appelle les "trolls des brevets", ces sociétés qui obtiennent des brevets et font leur beurre sur des litiges ou par des menaces de procès. A contre cœur il a accepté d'ajouter une ultime phrase qui consacrerait l'accord Novell-Microsoft, mais il dit "J'espère qu'on en arrivera pas là". Le maintien de cette phrase dépend, d'après lui, des commentaires de la communauté sur cette version.

Après le lancement de la GPLv3, Stallman prévoit une révision de la GNU Free Documentation License et de l'Affero GPL, une licence créée pour les logiciels proposés comme services Web.

Malgré les efforts d'internationalisation de la GPLv3, il ne s'attend pas à voir de traductions officielles de la licence dans d'autres langues. "Ça serait bien d'avoir des versions officielles", dit-il, "mais c'est aussi très risqué. Je suis très peu enclin à prendre ce risque."

A propos de l'opposition d'opinion entre l'open source et les logiciels libres, que la GPLv3 a souvent accentués, Stallman dit qu'ils font tous les deux partie de la communauté du logiciel libre. Cependant, en se référant aux défenseurs de l'open source, il dit : "Tout le monde n'accorde pas le même prix à la liberté. Quand on donne la liberté aux gens de choisir leur propre opinion, ils ne tomberont pas tous d'accord." En même temps, Stallman dit qu'il pense qu'un consensus sur l'adoption de la GPLv3 est essentiel, car "les projets sous GPLv3 protègent les utilisateurs de nouvelles menaces contre leur liberté. Si la GPLv3 n'est pas largement adoptée, ou si de nombreux programmes restent sous GPLv2, alors ces programmes seront vulnérables à de nouveaux types d'attaques. Par exemple, si Linus (Torvalds) ne se convertit pas à la GPLv3, alors les utilisateurs de Linux seront vulnérables à la TiVoisation. C'est un problème important.

"Quand le but d'un programme est de vous limiter, le rendre plus puissant et fiable dans ce qu'il fait est pire. C'est donc une erreur de dire que l'obtention de logiciels plus puissants et fiables est un but" comme le maintient le point de vue de l'open source. Au contraire, "le but du mouvement des logiciels libres est de vous donner le contrôle du logiciel que vous utilisez. Ensuite, si vous voulez le rendre plus puissant vous pouvez y travailler." Malgré cela, Stallman garde quelques espoirs. Il remarque, par exemple, que les défenseurs de l'open source ne partagent pas tous le même avis. Il montre Sun Microsystems comme exemple d'une société dominée par la pensée open source et qui envisage d'embrasser la GPLv3 pour des raisons qui lui sont propres.

Linus Torvalds

Quand la deuxième version de la GPLv3 a été publiée, Linus Torvalds a été l'un de ses critiques les plus francs. Bien qu'il insiste sur le fait qu'il ne donne qu'une opinion rapide sur la GPLv3, et qu'il pourrait changer d'avis à mesure qu'il l'étudie plus en détail, sa première réponse à la troisième version donne un accord nuancé.

"Est-elle meilleure ?" demande Torvalds rhétoriquement. "Et de loin ! Mais elle a été limitée de manière qui la rendent plus saine au moins. Je devrai y réfléchir. Le langage employé est plus clair et meilleur que celui de la GPLv2 à bien des égards et de nombreuses zones que j'aurai qualifiées d'"absurdité évidente et complètement idiote" ont soit été améliorées soit complètement retirées."

Torvalds approuve la reformulation et les clarifications des termes ajoutés dans la Section 7, suggérant que la troisième version rende la double licence plus simple dans des cas particuliers. Il ajoute : "l'absence totale de nouvelles restrictions est un soulagement énorme et rend la licence bien plus utile".

"Je ne suis pas certain que l'accord Novell (avec Microsoft) méritait tant d'attention", dit Torvalds, en référence à la Section 11 de la version. Cependant, il dit : "Je pense effectivement que ce sujet (plutôt que l'hystérie autour des DRM) était potentiellement une bien meilleure incitation à écrire la GPLv3 de prime abord."

La Section 6 autour de l'anti-TiVoisation couvre déjà cette "hystérie des DRM". A propos du langage tenu dans cette section de la version actuelle, qui remplace une interdiction complète des technologies restrictives et les logiciels espions par l'obligation d'inclure leur code source : "Il prétend toujours contrôler non seulement le logiciel, mais aussi le matériel ou l'environnement sur lequel le logiciel

est installé. Je trouve ça odieux, mais le langage est bien meilleur, et ils (la FSF) semblent avoir réalisés que leurs anciennes versions étaient insensées (c'est-à-dire qu'ils disent clairement que si quelque chose est créé pour ne pas recevoir de mise à jour il n'y a pas de raison qu'il demande des 'informations d'installation' et des clés). Le fait de limiter les choses aux 'dispositifs pour utilisateurs' permet aussi de se débarrasser d'un grand nombre de problèmes idiots présents dans les versions antérieures de la GPLv3. "Sous cette nouvelle forme je pense que la GPLv3 devient au moins une alternative viable à la GPLv2. Je dois encore la lire en entier quelques fois et laisser les choses se décanter mais intérieurement, après l'avoir lue une première fois, je sens au moins que je n'ai plus cette impression de 'Je n'aurai jamais choisi cette licence si je devais commencer un projet.'"

Novell

Représentant Novell, Bruce Lowry a refusé de fournir une réaction détaillée à cette version, apparemment pour réserver les commentaires officiels pour la version finale. Cependant il indique rapidement que "Rien dans cette nouvelle version de la GPLv3 ne réduit la capacité de Novell à inclure des technologies sous licence GPLv3 dans SUSE Linux Enterprise, openSUSE et d'autres offres open source, ni maintenant ni dans le futur."

Lowry dit "Nous sommes fermement décidés à poursuivre le partenariat avec Microsoft et ce, comme nous l'avons toujours fait, en total accord avec les termes des licences des logiciels que nous proposons, logiciels sous licence GPLv3 compris. Si la version finale de la GPLv3 a un impact potentiel sur l'accord que nous avons avec Microsoft, nous réglerons ça avec Microsoft." Lowry décrit Novell comme un "soutien fort au logiciel libre et à l'open source" et comme un "donateur important pour un grand nombre de logiciels libres et de projets open source."

Un avant goût des mois à venir

Stallman, Torvalds et Novell attendent de voir ce qui va se passer et la version finale de la licence avec de s'engager pleinement. Cependant, leurs commentaires mettent en avant les problèmes autour desquels tourneront les discussions dans les trois prochains mois à mesure que la GPLv3 s'approche de sa version finale. Malgré des efforts évidents de la part de chacun pour éviter les conflits et malgré quelques signes encourageants, le consensus est encore loin d'être atteint, mais l'adhésion générale à la GPLv3 semble déjà plus probable qu'il y a six mois.

Bruce Byfield est un journaliste informatique qui écrit régulièrement pour Newsforge, Linux.com et IT Manager's Journal.