

• Editorial par **B. Rapacchi**

• Le projet PLUME  
par **J.-L. Archimbaud**

• Le logiciel libre, un modèle  
issu de la recherche  
par **B. Lang**

# SÉCURITÉ INFORMATIQUE

numéro 60

juillet 2007

SÉCURITÉ DES SYSTÈMES D'INFORMATION

## éditorial

### Liberté et sécurité

Le caractère crucial pour le domaine des systèmes d'information de l'existence des logiciels libres est démontré par la diversité des acteurs de la filière : éditeurs de logiciels, communautés du libre, intégrateurs, sociétés de conseil, SSL – sociétés de services en logiciels libres –, intégrateurs, avocats, économistes. Ce secteur, par sa maturité et son dynamisme, génère, en 2006, 300 millions de chiffre d'affaires en France et 2500 emplois directs. Les dernières élections présidentielles et législatives ont montré l'entrée de ces questions dans le débat politique, les candidats ayant eu à s'exprimer et à s'engager sur les problèmes liés aux brevets logiciels et aux droits d'auteurs.

Dans un dossier du *Monde informatique* du début de l'année, les directeurs des systèmes d'information « votent pour le libre ». Un des aspects positifs soulignés par cette enquête était une « assurance sécurité » offerte par le logiciel libre : « Ouvrir le code pour montrer la conception des logiciels garantit un bien meilleur niveau de sécurité. » Sans entrer dans le détail des discussions délicates menées dans le milieu du « libre » sur les différences entre « logiciel ouvert » et « logiciel libre », sur les diverses licences utilisées et leur impact sur l'utilisation du logiciel, les DSI voient dans ces logiciels une alternative aux boîtes noires parfois inquiétantes.

À l'heure où les établissements d'enseignement supérieur se voient offrir le choix entre une solution à base d'un ERP unanimement reconnu par la communauté internationale et une solution d'un consortium à base de logiciels ouverts, cette question de la sécurité offerte par ce type de logiciel est justement posée ; question encore plus d'actualité, alors que le CERT-A relève le nombre important d'incidents de sécurité liés aux sites web utilisant des briques « ouvertes » s'appuyant sur le langage PHP<sup>1</sup> et que l'UREC propose une synthèse sur les vulnérabilités des sites web liées à PHP<sup>2</sup>.

Dans ce numéro, Bernard Lang, de l'INRIA, vice-président de l'Association francophone des utilisateurs de Linux et des logiciels libres, qui vient de rejoindre le Conseil supérieur de la propriété littéraire et artistique, nous montre justement la relation entre SSI et logiciels libres et Jean-Luc Archimbaud présente les retombées du projet PLUME (Promouvoir les Logiciels Utiles, Maîtrisés et Économiques) pour la SSI. Ce numéro de *Sécurité informatique* est ainsi un moyen adéquat pour connaître les avancées menées au sein de cette communauté autour de la préoccupation de la sécurité des systèmes d'information.

**Bernard Rapacchi**

Directeur de l'Unité Réseaux du CNRS

1. <http://www.certa.ssi.gouv.fr/site/CERTA-2007-INF-002/CERTA-2007-INF-002.html>.

2. <http://www.urec.cnrs.fr/IMG/pdf/secu.corres.InfoT.ProjApp.failles-web-1.2.pdf>.

## Le projet PLUME

**Jean-Luc Archimbaud**

Directeur du projet, CNRS/UREC

*Le projet PLUME<sup>1</sup> a été lancé par l'UREC<sup>2</sup> mi-2006. Cette unité nationale de service du CNRS a, entre autres, un rôle d'animation technique des informaticiens de tous les laboratoires CNRS et donc une très bonne connaissance de l'informatique de terrain, de ses pratiques et de son évolution.*

### Constat

De très nombreux logiciels performants et stables sont disponibles actuellement. Ils sont peu coûteux, pour la plupart libres et gratuits, parfois partagés... PLUME les appelle logiciels économiques (le E de PLUME). Ils couvrent maintenant tout le spectre des applicatifs et sont parfois beaucoup plus innovants que les produits propriétaires.

La communauté enseignement supérieur et recherche était pionnière dans la pratique de ces logiciels libres, ainsi que dans la contribution à leurs développements mais, actuellement, elle perd toute cette avance. Par manque d'organisation, chaque informaticien, isolé ou dans un service informatique, ou utilisateur éclairé de la communauté, « réinvente la roue », alors que son homologue dans un autre laboratoire a fait le même travail. On appelle ici « utilisateur éclairé » le chercheur ou l'enseignant capable d'installer et d'utiliser un logiciel libre sur son poste sans support local. Quand il a besoin d'un logiciel (pour ses besoins propres, ceux de son service ou pour ses utilisateurs) qu'il pressent disponible en ligne, il utilise une méthode très artisanale : il contacte son voisin (« qu'est-ce que tu utilises ? »), un spécialiste du domaine (« qu'est-ce que tu recommandes ? »), interroge un serveur généraliste de type Framasoft, Wikipedia ou Google. S'il est téméraire, il pose une question dans une liste de diffusion... Puis il télécharge, installe, essaie... Parfois, le produit répond à son besoin et fonctionne. D'autres fois, non. Dans certains cas, il se lance dans un développement souvent partiellement redondant avec l'existant. Cela est refait des dizaines de milliers de fois, souvent pour les mêmes besoins. Le CNRS emploie 1 800 ingénieurs ou techniciens permanents en informatique et, si on cumule les postes dans les organismes de recherche et les universités, on arrive à environ 10 000 informaticiens sans compter les ..... suite page 2 ..... ➔

1. PLUME : Promouvoir les Logiciels Utiles, Maîtrisés et Économiques dans l'enseignement supérieur et la recherche.

<http://www.urec.cnrs.fr/jla>.

<http://www.urec.cnrs.fr/plume>.

2. UREC : Unité Réseaux du CNRS, <http://www.urec.cnrs.fr>.

— suite de la page 1 —

non-permanents et les utilisateurs éclairés. Si chacun fait le même travail pour trouver le bon agenda partagé ou l'anti-spyware ou... On peut imaginer le temps perdu au total. Les grosses entreprises ou certaines administrations ont déjà créé un service pour sélectionner et diffuser les logiciels libres intéressants en interne, notre communauté non.

En revanche, nous avons un atout extrêmement favorable. Dans notre communauté, de très nombreux personnels connaissent bien certains de ces produits, les utilisent régulièrement, en sont parfois des spécialistes. Cette expertise cumulée, dans des domaines très variés de l'informatique, est certainement globalement beaucoup plus riche que dans les grandes entreprises privées ou d'autres administrations. Mais ces compétences distribuées ne bénéficient pas aux autres.

## PLUME

L'idée simple de PLUME est, d'une part, de créer un comité stratégique interorganismes de concertation enseignement supérieur et recherche et, d'autre part, sur un plan plus technique, de mutualiser les compétences de terrain (serveur d'information, organisation pour faciliter le travail de choix et d'installation de logiciels pour la communauté). Les deux axes sont obligatoires et pourront progresser parallèlement. L'un bénéficiera de l'expertise de l'autre, l'autre pourra être piloté et soutenu par le premier.

## Sécurité

### Mais quel est le rapport avec la sécurité ?

Dans un logiciel libre, le code source est public, ce qui permet de vérifier si :

1. Il est globalement bien écrit. Cela est rassurant lorsque le programme offre des fonctions de sécurité vitales mais aussi pour la pérennité et l'évolution du produit.
2. Il fait correctement ce qu'il dit. Qu'est-ce qui prouve que les fonctions d'authentification, de chiffrement, de contrôle d'accès sont bien rendues ? C'est-à-dire sont fiables, avec des algorithmes reconnus et une bonne implémentation ? Dans les logiciels propriétaires, on ne peut que faire confiance à l'éditeur et on n'a géné-

ralement pas d'engagement de sa part en cas de défaillance de son logiciel. Le programme peut implémenter des algorithmes non validés, le développement avoir court-circuité les processus de tests et de validations et comporter des bugs évidents... En libre, tout est transparent et vérifiable. Les forums et listes de discussions ne manquent pas d'exemples de discussions autour de ces sujets.

3. Il ne fait pas autre chose que ce qu'il dit. Tout système étant maintenant connecté à Internet, il est très simple pour un éditeur de rajouter une petite verrue invisible qui lui permettra de recueillir des informations sur le poste et son contenu sans que l'utilisateur le sache (cela s'est déjà pratiqué, même par certains grands éditeurs), ou qui lui permettra de prendre la main à distance sur ce poste. Ce ne sont que deux exemples de tout ce que peut cacher un code de programme non public. Il y en a d'autres. Ce n'est pas faire preuve de paranoïa que de refuser l'angélisme. Dans l'hypercompétition économique actuelle et les rivalités exacerbées d'intérêt entre les États, notre dépendance vis-à-vis des éditeurs de logiciels justifie pleinement certaines craintes, surtout quand on sait que des actions de ce type, dans un but commercial ou géostratégique, sont couramment pratiquées.

### Le logiciel libre a aussi d'autres avantages

D'abord, il permet d'être indépendant vis-à-vis des éditeurs commerciaux car, en informatique, changer de produits ou de matériel est toujours une opération coûteuse. Ainsi est-on facilement à leur merci, sans parler des conséquences de défaillances diverses (arrêt du produit ou fermeture, rachat ou changement de politique de l'entreprise editrice). À l'inverse des produits commerciaux, les logiciels libres utilisent de préférence des formats ouverts, ce qui les rend plus facile à changer et assure une meilleure pérennité des données.

Ensuite, comme « les sources » sont disponibles, il est facile d'ajouter des modules pour assurer des fonctions particulières ou même d'en remplacer certains pour des raisons de sécurité (par exemple un algorithme de chiffrement). Enfin, les dernières recommandations du comité interministériel sur la sécurité des sys-

tèmes d'information est de favoriser les logiciels libres. S'il est vrai que la France n'a pas les moyens d'être totalement indépendante au niveau informatique en créant tout elle-même, au moins les logiciels libres sont-ils un levier pour maîtriser cette dépendance.

## Deux projets

Revenons à PLUME. Le projet stratégique va mettre en place un comité de concertation interorganismes de l'enseignement supérieur et de la recherche. Son objectif sera de promouvoir officiellement ces logiciels libres (en évitant tout intégrisme), de lancer des préconisations et des actions majeures de manière concertée (comme une migration à la suite bureautique [OpenOffice.org](http://OpenOffice.org) par exemple), de regrouper les différentes initiatives dans ce sens. Il est en phase d'étude et de concertation, mais apparaît de plus en plus nécessaire. On ne peut pas envisager, par exemple, que le CNRS fasse le choix de [OpenOffice.org](http://OpenOffice.org) sans concertation avec les autres EPST et les universités ([OpenOffice.org](http://OpenOffice.org) n'est qu'un exemple possible, pas obligatoirement le premier objectif).

Le second projet technique est la mise en place d'un serveur de référence de fiches descriptives de logiciels UME de PLUME : Utiles, Maîtrisés et Économiques (libres et assimilés) pour les trois grandes plateformes (MacOS, famille Unix, Windows). Une partie « objets communicants » est envisagée. Associés à ces fiches, d'autres documents sur le sujet (comparatifs, cours, annonces de séminaires...) sont créés ou pointés. Chaque fiche est rédigée par un informaticien ou un utilisateur éclairé, universitaire ou chercheur d'un EPST, qui maîtrise bien le logiciel (le M de PLUME). Ces logiciels sont utilisés en production dans un laboratoire ou une université, donc utiles (le U de PLUME). Les fiches ne sont pas un résumé de l'aide ou du mode d'emploi du logiciel, elles sont conçues pour :

- être des éléments de synthèse (fonctions principales et annexes),
- donner les moyens de retrouver de l'information (pointeurs vers la documentation, les listes de discussion...),
- offrir un historique des différentes étapes de l'évolution du logiciel et des différentes collaborations,
- donner un état du déploiement dans la communauté,

— suite page 3 —

... suite de la page 2

- faire un comparatif avec des produits similaires.

Ce serveur fait connaître et assure la promotion de ces logiciels et mutualise les compétences des personnels internes. En complément, d'autres actions vont émerger telles que la création de groupes de travail pour tester des produits, la mise en place de formations...

Le projet, lancé depuis fin 2006, se déroule en trois phases:

- Jusqu'à l'automne 2007: mise en place d'une maquette, définition du projet, de la plate-forme définitive et de l'organisation nécessaire.
- Pendant un an (jusqu'à l'automne 2008): mise en production réelle de la plate-forme et construction de l'organisation.
- Fin 2008-mi 2009: passage d'un mode projet à un service permanent avec transfert de l'exploitation et de l'organisation vers une structure pérenne d'exploitation.

Ce projet a une vocation évidemment multipartenaires. Il intéresse déjà fortement d'autres EPST et universités. C'est ce qui nous a conduits à le poursuivre avec un état d'avancement en phase avec les prévisions. Le CNRS via l'UREC n'est que l'initiateur. Une structure officielle de type GIS, institut ou autre, regroupant les partenaires, sera mise en place fin 2007 pour le pilotage.

## Le projet technique aujourd'hui

Tout est en ligne sur

<http://www.urec.cnrs.fr/plume>.

Le format des fiches (avec les champs précis), les processus de relecture, la structuration du serveur Web et l'organisation humaine sont définis. Mi-juin, un serveur maquette est ouvert (l'URL précédente) avec actuellement 45 fiches en ligne rédigées par 30 contributeurs, 8 en relecture et 77 propositions spontanées en attente. Les rubriques «informatique pour les services informatiques» (par opposition à informatique pour poste personnel), maths et travail coopératif sont officiellement ouvertes, avec des responsables de rubriques désignés (mais volontaires). IST-Documentation, Bio-Informatique et Sécurité SI sont en cours de création.

Après la rédaction d'un cahier des charges fonctionnel, une étude des pro-

duits (libres) disponibles aujourd'hui (plus de 200 qui répondaient aux besoins de base), une sélection de 15 d'entre eux et de 5 pour des tests, l'outil central de la plate-forme définitive a été choisi: Drupal<sup>3</sup>. Il intégrera des fonctions de publication Web (avec recherche par mots-clés...) mais aussi la gestion de flux d'informations et de travail coopératif (pour les propositions de correction de fiches, la rédaction collaborative de fiches...). Une première version sera mise en place en octobre pour remplacer le serveur maquette actuel.

Les différentes familles d'utilisateurs de la plate-forme sont définies. Selon l'appartenance, les clients n'auront pas les mêmes informations et les mêmes actions possibles:

- Les visiteurs, non authentifiés, n'auront accès qu'aux documents validés et stables avec éventuellement la possibilité de mettre des commentaires. Certaines informations seront masquées (l'adresse électronique du contributeur par exemple).
- Les lecteurs de la communauté enseignement supérieur et recherche (avec une authentification très légère à déterminer: nom de domaine?) pourront avoir accès à des documents complémentaires tels que des comparatifs. Ils pourront aussi faire des suggestions de fiches, des propositions de modifications plus facilement avec un accès plus direct aux auteurs
- Les contributeurs au projet PLUME qui ont déjà rédigé ou relu une fiche (avec une authentification plus forte, peut-être OpenID avec une liaison automatique avec les IGC CNRS et autres, les fédérations d'identités si possible) pourront améliorer les fiches existantes, créer des fiches de manière collaborative, avoir accès à des fiches bêta (produits en test), contacter directement l'auteur...
- Les responsables de rubriques pourront gérer complètement les fiches et les processus (relecture...) ainsi que les documents associés aux fiches.
- Le rédacteur en chef (fonction pouvant être partagée) veillera à la cohésion de l'ensemble.

Le but est d'offrir le maximum de souplesse, d'interactions possibles et de force de proposition, tel que le permet Wikipedia, mais sans avoir à gérer toutes les perturbations (questions basiques, avis

sans intérêt...) que peut engendrer une ouverture totale. Avec la discrimination décrite ci-dessus, on pourra positionner le curseur des droits plus ou moins ouverts selon le comportement des visiteurs.

## Les spécificités et innovations du projet technique

Les clients visés par le projet PLUME sont les informaticiens, ingénieurs ou techniciens en informatique (environ 10000) et les utilisateurs éclairés. Cette population est, de plus, restreinte à l'enseignement supérieur et la recherche. Donc la cible du serveur PLUME, les clients dirait-on dans d'autres milieux, est très précise. Ce projet a aussi d'autres spécificités qui le différencient de services grand public comme Framasoft ou le portail logiciels libres de Wikipedia dont le principe est similaire.

**La première spécificité** est de répondre aux besoins de l'enseignement supérieur et de la recherche. Cela consiste, dans un premier temps, à éviter un prosélytisme systématique pour le libre, tout en tenant compte du rapport qualité prix. Si un produit commercial est très performant, peu cher et fiable, il faut l'utiliser. Dans un second temps, il répond aux besoins professionnels (on retrouve l'objectif de logiciels utiles). Donc, pas de jeux, peu d'outils multimédias ludiques seront référencés, par exemple. Cibler sur les besoins oriente aussi la classification des logiciels: par fonction informatique (bureautique, services Internet, logiciels de gestion...) mais également par métier ou activité de la recherche et de l'enseignement (maths, bio-informatique, chimie, électronique, statistiques, calcul numérique, enseignement à distance, conception de cours...). Un visiteur PLUME avec un profil métier de la recherche ou de l'enseignement supérieur devra retrouver très rapidement les logiciels qui peuvent l'intéresser. C'est une approche métier.

**Deuxièmement**, PLUME ne référence que ce qui est utilisé dans la communauté, ce qui est une preuve de l'utilité et de la qualité: ça fonctionne «en production». Le but n'est pas de faire un catalogue exhaustif des logiciels libres qui existent mais d'indiquer des logiciels de qualité et UME: *... suite page 4* ➤

3. [http://www.urec.cnrs.fr/IMG/pdf/LL.PLUME\\_Choix\\_Drupal.pdf](http://www.urec.cnrs.fr/IMG/pdf/LL.PLUME_Choix_Drupal.pdf).

— suite de la page 3 —

Utiles, Maîtrisés, Économiques pour le travail toujours, dans le contexte de l'enseignement supérieur et de recherche.

**Le troisième point** est de faire connaître les développements internes. En effet, de nombreux développements « libres » ont été faits par des chercheurs, des enseignants ou des ingénieurs sans que cela soit visible comme une production du laboratoire ou de l'université. Sans empiéter sur les services de valorisation, PLUME référence « de préférence » ces produits mais avec les mêmes critères UME. Il faut que le logiciel soit déjà diffusé (sur un Web, dans Sourceforge, etc.), ce qui veut dire que le développement est abouti et possède une documentation... Il faut aussi qu'il fasse la preuve de son intérêt en étant utilisé en production par plusieurs autres sites.

La caractéristique suivante est d'avoir un processus clair et public de sélection des produits, de relecture par au moins trois personnes et de mise à jour avec une périodicité inférieure à six mois. Ce dernier point est un défi fort mais crucial. Tout document (fiche ou autre) non relu et non revalidé sera effacé ou affiché avec un état « périmé ». Le monde du libre évolue très rapidement et, sans chercher à suivre toutes les versions qui sortent (proposer systématiquement la dernière version non testée en production est d'ailleurs fortement déconseillée), il faut être à jour.

Un autre point stratégique est de décentraliser les choix de logiciels et de rédacteurs de fiches dans des rubriques avec des animateurs-référents responsables par rubrique. Ces responsables de rubriques sont des personnes qui effectuent déjà en partie ce travail de sélection (dans des services informatiques pour des gros laboratoires, des réseaux de métiers, des universités), avec des compétences spécifiques. Là, ils peuvent le faire à un niveau national sans beaucoup de surcroît de travail. Ce sont des volontaires, avec l'accord de leur directeur. La mise en place de ces rubriques et des responsables associés se fait progressivement.

Une dernière caractéristique est de reconnaître et de valoriser l'expertise des informaticiens de la communauté, personnels souvent isolés avec des compétences sur certains domaines, via leur contribution à PLUME en tant que rédac-

teur (chaque fiche comporte le nom de l'auteur) ou responsable de rubrique (dont les noms sont publics).

## Quels sont les gains attendus du projet technique ?

Les économies en dépenses logicielles sont le premier gain. Il est évident que passer de solutions commerciales payantes à des solutions gratuites ou presque entraînera des économies à tous les niveaux, national, de laboratoire, d'université. Le second gain est la rationalisation du travail des informaticiens, chacun pouvant accéder très rapidement à des logiciels classifiés, avec toutes les informations de base. Ils pourront alors se concentrer sur le support à la recherche et à l'enseignement dans des domaines techniques spécifiques. Ce gain est énorme. Dans la même continuité, en ne proposant que quelques logiciels pour une même fonction, cela limitera la myriade de solutions possibles. Cette hétérogénéité souvent non justifiée pose d'énormes problèmes de compatibilité et de besoins de compétences trop diverses. Nous avons déjà parlé de la valorisation des développements internes affichés dans PLUME ainsi que de la reconnaissance de l'expertise des ingénieurs contributeurs. Ce projet permettra aussi de les intégrer dans une communauté et de les inciter à la veille technologique et au développement, ce qui est nécessaire pour le support à la recherche. Pour les décideurs-acheteurs, PLUME peut servir de référentiel de logiciels économiques et ainsi les aider lors de choix d'achat de solution commerciale, de rédaction de cahier des charges, d'appel d'offres, en permettant d'avoir très facilement un point de comparaison (gratuit) avec les solutions commerciales.

Pour la sécurité, ce référentiel permettra d'agir en amont sur les recommandations de produits en choisissant des produits avec une sécurité suffisante. Un expert sécurité participera au comité technique pour avoir une vision globale sécurité sur les logiciels présentés et alerter sur certains trop laxistes et peu sérieux.

L'utilité d'un comité stratégique est évidente : il permettra d'avoir des actions concertées et cohérentes entre personnes qui travaillent de la même

manière et, dans un premier temps, d'être un lieu d'échanges sur ce sujet, lieu qui n'existe pas.

## Projets dérivés

La création de rubriques amène les contributeurs et spécialistes en logiciels à se regrouper en réseaux par grands domaines d'intérêt. Ceux-ci offrent l'opportunité d'un travail coopératif d'où peut émerger des projets. Par exemple, deux sont actuellement en cours.

La mise en place d'une plate-forme de formation à distance pour les logiciels libres nous a été proposée dans le cadre d'un stage longue durée. C'est à étudier avec la formation permanente du CNRS. Toujours en termes de formation, à destination des chercheurs et ingénieurs développeurs dans les laboratoires, nous préparons pour 2008 une école thématique avec les départements avec les départements MPPU (Mathématiques, Physique, Planète et Univers) et ST2I (Sciences et Technologies de l'Information et de l'Ingénierie) du CNRS et l'INRIA appelée ENVOL : École pour le développement et la Valorisation des Logiciels en environnement de recherche. Le constat est que, faute de méthode, les développements souvent très inventifs des laboratoires ont beaucoup de mal à passer de l'utilisation locale à une diffusion vers l'extérieur, car le travail de remise en forme est beaucoup trop lourd. L'école présentera des méthodes et outils de développement utilisables dans un laboratoire de recherche et les différents modèles de valorisation (en particulier logiciels libres). Une partie sur la sécurité (comment développer du code sans trou de sécurité) est d'ailleurs prévue.

Cela donne deux exemples de projets que fait émerger PLUME. D'autres suivront. PLUME veut être un catalyseur dans le domaine du logiciel. Il ne veut pas tout faire mais peut facilement regrouper des compétences pour qu'elles travaillent ensemble.

## Accueil du projet et avenir de PLUME

Le retour des informaticiens sur la plate-forme de référence est très positif, voire enthousiaste. Beaucoup ont conscience qu'ils perdent un temps énorme et sans intérêt à réinventer la roue en essayant de trouver seuls le « bon logiciel ». La preuve en est les nombreuses — suite page 5 —

..... suite de la page 4

propositions spontanées (77 en attente actuellement), reçues sans publicité. C'est très encourageant car cela démontre le besoin de ce service par les clients visés, mais aussi leur volonté de contribuer.

Le retour des structures nationales informatiques en place est bon côté recherche, plus mitigé dans l'enseignement supérieur. Ce projet est nouveau. Il concurrence des initiatives plus réduites déjà existantes, il risque de modifier des processus d'achats et il peut entraîner des réticences sur le fait que le CNRS soit l'initiateur du projet. Le projet est dans une phase de maquette pour montrer ce que peut être une plate-forme; la forme définitive du serveur ne sortira qu'en octobre. Il est donc normal que, n'ayant pas vu le résultat, certains doutent. Mais il n'y a pas de critiques sur les objectifs ni de propositions d'une autre méthode pour arriver aux mêmes résultats. Malgré le peu de contacts que nous avons eus, plusieurs organismes de recherche, universités et gros laboratoires ont affiché officiellement un soutien, parfois avec une contribution forte. La liste est tenue à jour sur le site PLUME. Ce sera avec les soutiens officiels que

nous lancerons la structure de pilotage fin 2007. À noter que des contacts avec des universités ou laboratoires de recherche étrangers (en particulier Québec et Belgique) ont été initiés et le projet les intéresse. Il est obligatoire d'avoir une ouverture internationale pour mutualiser les compétences en restant dans le métier « enseignement supérieur-recherche ». Mais le serveur étant en français, sans réelle possibilité de faire autrement (il serait difficile de demander à des rédacteurs bénévoles d'assurer une traduction et de la maintenir), nous allons contacter nos homologues dans les pays francophones.

Jusqu'à présent, nous avons donc été confortés dans l'utilité du projet technique et dans sa faisabilité avec des moyens limités. Nous sommes très optimistes sur le passage en production. Néanmoins, des ressources, l'équivalent de trois temps pleins, seront à assurer dans l'année 2008 pour avoir le noyau dur minimal et solide qui pilotera l'ensemble. Actuellement, l'UREC fournit

l'équivalent d'un ingénieur permanent et d'un ingénieur en CDD jusqu'à la fin décembre. Cela est vraiment très minime face aux économies réalisées, ne serait-ce que par rapport au budget de licences logicielles des organismes de recherche et des universités.

Le projet technique bien lancé, le projet stratégique pourra s'y adosser et son lancement est un des chantiers dans les prochains mois avec des contacts à établir avec le ministère, toutes les directions d'organismes de recherche et certaines universités.

Pour trouver une fiche, pour plus d'informations, pour suivre l'évolution ou pour participer, connectez-vous sur le site <http://www.urec.cnrs.fr/plume>. ■

Jean-Luc.Archimbaud@urec.cnrs.fr

## Le logiciel libre, un modèle issu de la recherche

**Bernard Lang**, directeur de recherche à l'INRIA, vice-président de l'AFUL

Le logiciel libre garde un certain mystère. Faut-il le considérer comme une (r)évolution technologique, économique ou sociologique? Probablement un peu des trois. En est-on aux balbutiements ou à la maturité? Les avis divergent mais, ce qui compte réellement, c'est de réaliser que c'est devenu incontestablement un facteur à prendre en compte dans tous nos choix, choix d'équipement informatique, choix de politique de valorisation, choix de contexte de la recherche qui, dans bien des disciplines, intègre de plus en plus d'informatique.

Chercheur en informatique, je me suis alarmé dans les années quatre-vingt-dix d'une monoculture envahissante, quel que soit le domaine concerné de l'informatique: systèmes d'exploitation, bases de données, conception assistée, systèmes de calcul scientifique, bureautique, etc. Cette monoculture est la résultante de divers facteurs technologiques, économiques et sociologiques — effets de réseau techniques et sociologiques, coût de revient nul de chaque copie supplémentaire d'un logiciel — qui favorisent l'émergence de monopoles économiques, dans la mesure où l'on se cantonne aux logiciels propriétaires suivant le modèle commercial qui s'est développé dans les années quatre-vingt et quatre-vingt-dix.

Les monopoles sont en eux-mêmes un souci, surtout dans un secteur aussi central et omniprésent que l'informatique, car ils se

traduisent généralement par une augmentation injustifiée des prix et une baisse de l'investissement pour l'innovation. Du point de vue de la recherche, en informatique au moins, ils peuvent se traduire par un ensemble de contraintes plus ou moins insidieuses. Les systèmes informatiques sont souvent des réalisations extrêmement complexes dans lesquelles les travaux de recherche d'une équipe ne portent que sur quelques composants. Pour tester leurs travaux sur des données réelles, les chercheurs doivent donc intégrer leur réalisation dans des systèmes plus vastes. Encore faut-il qu'ils en obtiennent l'autorisation et les informations nécessaires, et que ces systèmes n'imposent pas des contraintes incompatibles avec les choix techniques requis par les caractéristiques des expérimentations menées.

On peut donc s'inquiéter que cette monoculture, doublée d'un contrôle étroit des usages par des industriels soucieux de préserver leurs droits de propriété intellectuelle et leur contrôle du marché, ne se traduise par des limitations tant sur la nature et la diversité technique des recherches que sur la publication des résultats sous une forme utilisable par tous, en raison de clauses de confidentialité, ou n'entraîne des duplications inutiles faute de pouvoir utiliser des systèmes existants. Certaines de ces difficultés ne sont pas nécessairement liées à l'existence de monopoles, et résultent souvent du simple souhait ..... suite page 6

d'appropriation et de valorisation financière des résultats par les chercheurs eux-mêmes.

### Le logiciel libre répond à ces soucis

Sans rentrer dans les détails juridiques, le principe en est que chacun peut librement l'étudier, l'utiliser, le modifier et le redistribuer. Et cela est vrai techniquement grâce à la disponibilité du code source qui sert à écrire et à comprendre les programmes, et légalement par un statut juridique (licence) qui l'autorise expressément. Chacun peut donc participer à la réalisation de ces logiciels, dans la mesure où sa contribution est jugée pertinente par la communauté des utilisateurs et, surtout, par celle des autres développeurs qui participent à l'évolution du logiciel. Ce mode de développement n'est pas sans rappeler, et ce n'est pas un hasard, celui de la recherche scientifique. Ce n'est pas un hasard parce que ce paradigme de développement des logiciels est dû à un chercheur, Richard Stallman, qui travaillait à l'époque — début des années quatre-vingt — au MIT. Et, surtout, ce n'est pas un hasard parce que la matière est très voisine. Intuitivement, un logiciel est, comme la science, une forme de connaissance, de savoir-faire, même s'il est destiné à être mis en œuvre par des machines plutôt que par des hommes. Plus formellement, les logiciels ont montré, il y a vingt-cinq ans, qu'un logiciel est de même nature qu'une preuve mathématique. Cette preuve démontre un théorème qui est la spécification de la fonctionnalité du logiciel, *grosso modo*: à partir de telles données, on peut produire tel résultat. Or n'est donc pas déraisonnable de penser que les mêmes modes de travail et de création puissent être adaptés dans les deux cas.

C'est peut-être cela aussi qui explique l'opposition d'un très grand nombre d'informaticiens à la brevetabilité des méthodes logicielles. Que diraient les mathématiciens si on pouvait breveter les techniques de preuve et les obliger à payer des royalties chaque fois qu'ils en font usage, voire chaque fois qu'ils communiquent des théorèmes démontrés grâce à ces techniques ?

### Code source ouvert et libre

C'est aussi ce qui fait préférer les logiciels libres, dont le code source lisible est librement disponible, aux logiciels propriétaires, dont le code source reste généralement secret — et dont on n'a que des versions « compilées » pour les besoins des machines —, mais qui sont complètement cryptiques pour un être humain. Pour un mathématicien, cela reviendrait à fournir des théorèmes que les gens pourraient utiliser — moyennant royalties, bien sûr, pour un théorème propriétaire —, mais sans qu'ils puissent en examiner la preuve ni donc rien apprendre des techniques de preuve utilisées. Même si l'on a une totale confiance dans la compétence et la probité d'un mathématicien auteur d'une preuve, nul n'acceptera un théorème s'il n'est pas soumis au processus social du contrôle, de la vérification et des corrections par les pairs. Pourquoi en irait-il différemment de la création des logiciels, alors même que les logiciels sont souvent des preuves particulièrement complexes et donc plus facilement susceptibles d'erreurs, ce que l'expérience confirme amplement ?

Ces erreurs accidentelles relèvent de la fiabilité des logiciels, fiabilité qui est déjà une composante essentielle de la sécurité des systèmes. Trop souvent, les failles de sécurité ne sont que l'exploitation d'une erreur de programmation qui permet de

faire exécuter par le logiciel autre chose que ce qui était prévu par le programmeur.

### Oeufs de Pâques

Mais l'erreur peut aussi ne pas être accidentelle et avoir été intentionnellement introduite dans le but de diffuser un logiciel vulnérable à certaines attaques. Il ne s'agit plus d'erreur, et le logiciel peut même inclure des composants secrets destinés à espionner son utilisateur ou à porter atteinte à l'environnement où il est utilisé. Mais cela sera difficilement détectable si l'on ne peut en examiner le code source, la preuve lisible de ce que le logiciel fait réellement. Certains utilisateurs privilégiés ont parfois accès à ce code source pour des logiciels propriétaires. Mais cela ne devrait en rien les rassurer: les logiciels sont souvent si gros et si complexes que leur contrôle n'est réellement possible que s'ils peuvent être examinés par une très large communauté. En outre, avec l'apparition régulière de nouvelles versions, ce travail est un recommencement perpétuel. Enfin, ce contrôle n'est fiable et réalisable que si l'on peut recompiler soi-même le code source pour être sûr que c'est bien cela qu'exécute la machine. C'est rarement le cas.

Ce genre de risque est bien plus faible avec les logiciels libres. Il est cependant important de veiller à recevoir des versions munies de contrôles d'intégrité, provenant de sources authentifiées, si l'on ne veut pas prendre de risque concernant la sécurité.

L'importance des échanges, de la collaboration, de la concurrence et du jugement des pairs n'est pas sujette à contestation dans la recherche scientifique. On peut légitimement penser que ce modèle de la science ouverte est aussi le plus efficace, techniquement et économiquement, pour la création des logiciels. Le succès croissant des logiciels libres, dans les administrations comme dans le monde industriel et commercial, semble en témoigner, même s'il est ralenti par l'inertie des positions établies selon d'autres modèles de développement (les logiciels dits propriétaires) dont on peut penser que, pour la plupart, ils résultent plus des hasards de l'histoire que de la nécessité techno-économique. ■

[Bernard.Lang@datcha.net](mailto:Bernard.Lang@datcha.net)

## SÉCURITÉ INFORMATIQUE

Numéro 60 juillet 2007  
SÉCURITÉ DES SYSTÈMES D'INFORMATION

**Sujets traités:** tout ce qui concerne la sécurité informatique. Gratuit.

**Périodicité:** 4 numéros par an.

**Lectorat:** toutes les formations CNRS.

**Responsable de la publication:**

JOSEPH ILLAND

Fonctionnaire de Sécurité de Défense

Centre national de la recherche scientifique

3, rue Michel-Ange, 75794 Paris-XVI

Tél. : 01 44 96 41 88

Courriel : [Joseph.Illand@cnsr-dir.fr](mailto:Joseph.Illand@cnsr-dir.fr)

<http://www.sg.cnsr.fr/fsd>

**Rédacteur en chef de ce numéro:**

ROBERT LONGEON

Chargé de mission SSI du CNRS

Courriel : [robert.longeon@cnsr-dir.fr](mailto:robert.longeon@cnsr-dir.fr)

ISSN 1257-8819

Commission paritaire n° 1010 B 07548

La reproduction totale ou partielle des articles est autorisée sous réserve de mention d'origine.