Changement d'echelle dans les technologies de l'information et de la communication : nouveaux defis scientifiques, ethiques et organisationnels

Roberto Di Cosmo *

Résumé

Les avancées rapides des technologies de l'information et de la communication ont revolutionne notre societe, a des degrees qui etaient impensables il y a seulement quelques annees. Nous echangeons plus de donnees, plus vite, avec plus de personnes ; nous collaborons avec des partenaires au quatre coins de la planete ; nous accumulons des masses enormes d'informations ; nous ecrivons ou utilisons des logiciels plus complexes, en puisant dans des grandes masses de composantes, qui evoluent continuellement.

Il s'agit d'un veritable changement d'echelle qui pose des nouveaux defis scientifiques, ethiques et organisationnels, auxquels on attend des reponses nouvelles de la part de nos chercheurs.

Note : cet article est basé sur l'exposé donné à Lille, le 11 décembre 2007, à l'occasion des 40 ans de l'INRIA.

Table des matières

Pour les 40 ans de l'INRIA, j'ai eté invité à donner une conférence plenière dans le cadre des manifestations qui se tenait à Lille, en Décembre 2007. Ce n'était pas une défi simple, pour un Italien qui était arrivé à Paris un peu par hasard comme étudiant de doctorat quelque vingt ans auparavant, donc à la

*Professeur d'Informatique, Université Paris Diderot - www.dicosmo.org - roberto@dicosmo.org

moitié du parcours historique de l'Inria qu'on était en train de fêter.

J'ai décidé alors de profiter de l'occasion pour prendre un peu de recul et jeter un regard sur les évolutions technologiques des années passées, souvent appelées révolutions aujourd'hui par ceux qui n'ont pas eu la chance d'en suivre le dévéloppement et d'en comprendre les fondements scientifiques. C'est un exercice fort utile, si on veut en tirer quelques leçons pour l'avenir.

1 Un peu d'histoire

On peut commencer en choisissant quelques éléments de l'histoire de ces vingt dernières années sur lesquels l'INRIA a beaucoup pesé pour mener aux révolutions actuellement en cours. Nous avons tous constaté l'engouement récent des médias pour le web 2.0 : pour qui ne s'en serait pas aperçu, il suffit de regarder la couverture du très réussi numéro de Courrier international titré « Révolution 2.0 » qui est reproduit dans la figure 1.

FIGURE 1 – Couverture du Courrier International de Octobre 2007.

Mais s'agit-il *vraiment* d'une révolution ?

La question prend tout sons sens pour qui sait que toutes les technologies mises en avant par le web 2.0 ne sont pas si nouvelles qu'on ne l'imagine, et cela d'autant plus si on a la chance d'être Français. Commençons par faire un petit saut en arrière, au lointain 1989. C'est l'année de ma première visite en France, et je dois avouer que j'étais plutot perturbé pas des nombreuses affiches arborant l'inscription 3615 ULLA, qui montraient de jolies demoiselles, assez peu vêtues il faut le dire : j'avais le plus grand mal à comprendre ce que pouvaient bien publiciser.

L'Italien qui débarquait en France que j'étais a mis quelque temps à découvrir cet objet magique qu'était le Minitel (voir la fig. 2), et qui était déjà massivement répandu. Si cette technologie est restée hexagonale, elle n'en a pas moins eu un impact non négligeable, et avait permi le dévéloppément de véritables réseaux sociaux, bien avant la vague d'aujourd'hui.

FIGURE 2 – Un minitel

Le world wide web est né quelques années après le Minitel, avec la généralisation de l'Internet. Ah, l'Internet ! Quand je demandais en 1995 aux Argentins s'ils savaient ce qu'était Internet, ils confondaient avec Internet Explorer et le web, et me répondaient que c'était une invention de Microsoft, ce qui m'énervait passablement. En France, on sait bien que le world wide web n'est pas exactement une invention américaine. Ce sont Tim Berners-Lee et Robert Cailliau qui l'ont développé en 1989 au CERN, en Suisse. Mais ce que même les Français ont tendance à oublier est que le world wide web a été dévéloppé sur des machines NeXT, et pas sur la caricature d'ordinateurs qui étaient à l'époque les PC sous Windows (un des premiers serveurs web au monde est dans l'image de la figure 3). Et c'est dommage de l'oublier, parce-que le système NeXTSTEP, sans quoi le web n'existerait pas, s'il était bien intégré et commercialisé par une

entreprise américaine [1], était largement basé sur des idées françaises sorties de l'INRIA.

FIGURE 3 – Le premier serveur Web au monde.

Dans l'image de la figure 4, on voit un aperçu du world wide web de 1989 : il contenait déjà de l'hypertexte, des images et de la navigation.

L'image de la figure 5 est plus intéressante ; elle montre une capture d'écran de la version couleur du navigateur/éditeur web sur NeXTSTEP qui date de 1993. Elle contient de nombreux onglets et fonctionnalités qui *ne figurent plus* dans nos navigateurs actuels, et notamment la possibilité de *éditer* une page web distante, via la méthode `HTTP PUT`, qui était bien incorporée dans la fonctionnalité de base de ce navigateur, mais qui a été après rétirée des navigateurs dévéloppés pour d'autres plateformes.

A bien y regarder, donc, ce qu'on appelle aujourd'hui "Web 2.0" n'est qu'une tardive reimplémentation (avec les améliorations rendues possibles par 15 ans d'évolution technologique) des idées originaires de Cailleau et Berners Lee, qui avaient déjà conçu, et réalisé, un outil collaboratif.

Il serait alors plus approprié d'oublier le marketing, et de reconnaître que nous vivons l'époque du web *1.0*, après avoir épuisé les potentialités des versions dégradées (web alfa ou web beta) du concept originaire, qui a été simplifié pour être massifié.

Quand nous évoquons les moteurs de recherche aujourd'hui, tout le monde pense à Google. Là aussi, le concept de moteur de recherche est encore peu connu quand Webcrawler, le premier d'entre eux, est inventé en 1992. Il a été

1. Il s'agissait de NeXT, fondée par Steve Jobs, et c'est bien la technologie NeXT qui est au coeur du succés des Mac, iPod, iPhone et iPads d'aujourd'hui.

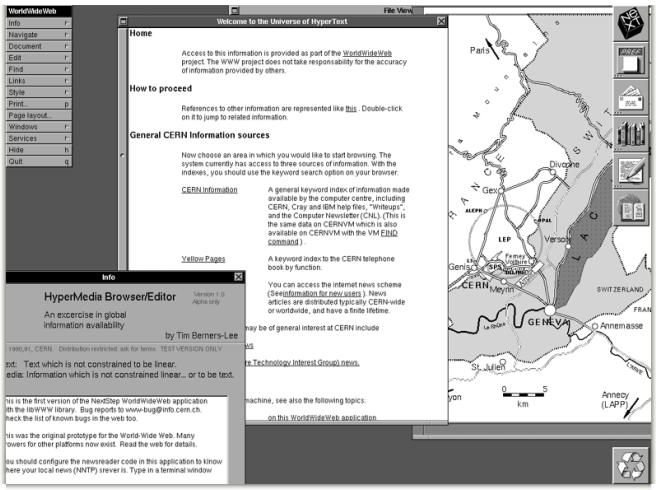


FIGURE 4 – Le premier navigateur et éditeur Web.

développé par un étudiant américain en utilisant l'indexing kit de NeXTSTEP. Encore une fois, vous pouvez donc trouver dans ce moteur de recherche du code en provenance de l'INRIA, bien que celui-ci ait été efficacement maquillé et mis sur le marché ailleurs. Dans le même temps, en 1990, la révolution de la création d'un noyau de système d'exploitation appelé Linux se met en marche. Cette révolution aussi naît en Europe.

La conclusion est simple : le moteur *technologique* de la croissance des technologies de l'information et de la communication (TIC) est né en Europe. Mais si le moteur est construit ici, la carrosserie, et les bénéfices de la commercialisation du produit, sont faits ailleurs. Etant d'extraction académique, j'ai l'irrimédiable defaut de penser qu'il faut donner crédit aux personnes qui font le travail créatif : je considère donc que cet état de fait est insatisfaisant, et qu'il s'agit d'un problème récurrent auquel il serait temps de trouver un remède.

Au délà des questions de paternité, il est intéressant de noter comme, déjà avec ces quelques simples exemples, on constate que toutes les idées fondamentales qui font le succès du web actuel existaient déjà il y a dix ans. La révolution ne se trouve donc pas dans la technologie logicielle *en tant que telle*, mais ailleurs : si les idées fondamentales n'ont pas changé, la massification, elle, a changé tout, et pour permettre cette massification, il a fallu baisser les coûts et augmenter les performances.

Prénons le temps d'observer l'evolution entre 1997 et 2007 : quelques données significatives sont présentées dans les figures 6, 7 et 8.

Les processeurs passent de 10 millions de transistors à plus de 2 milliards. Les disques durs des ordinateurs portables avaient une capacité de 3 gigaoc-

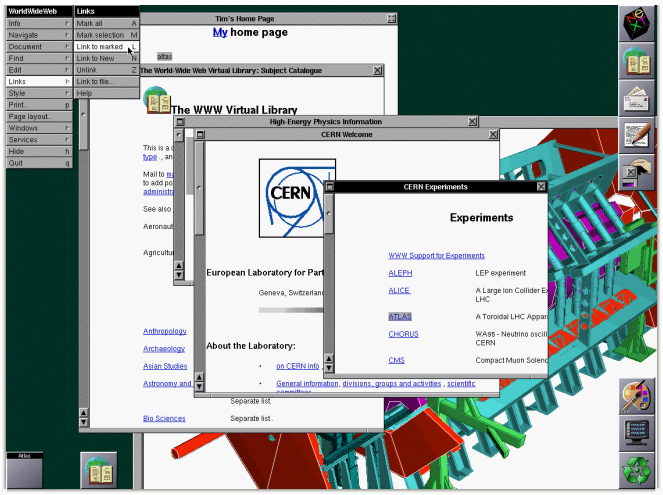


FIGURE 5 – Le navigateur et éditeur Web sur NEXTStep en 1993.

tets contre plus de 250 gigaoctets aujourd'hui. La capacité des disques durs des serveurs était de l'ordre de 10 gigaoctets alors qu'elle est de 1 teraoctet actuellement. La bande passante descendante de la connexion à Internet ne dépassait pas 256 kilobits ; elle atteint désormais 28 mégabits.

Le passage de la bande passante montante de 128 kilobits à 1 mégabit en ce moment constitue une exception importante dans cette liste d'exemples, et devrait attirer l'attention : cela correspond au fait que malgrè tout, on continue de considérer l'internaute comme un *consommateur* plutôt qu'un acteur. Cette dissymmétrie va rapidement disparaître avec la généralisation de la fibre optique, qui rendra les connexions aussi rapides en montée qu'en descente, et c'est là que les innovations majeures nous attendent : qui cherche à les anticiper pour créer une nouvelle industrie a intérêt à commencer à y reflechir rapidement.

Nous sommes passés de quelques milliers de sites web en 1997, à 1,5 million en 2000, à plus de 150 millions de sites aujourd'hui.

Le nombre de foyers connectés à l'Internet haut débit en France atteint 14 millions en 2007 alors qu'il était négligeable en 1997.

Enfin, au lieu d'être connectés et facturés à la seconde ou à la minute, nous sommes désormais connectés en permanence au réseau.

Cette évolution est phénoménale : nous communiquons cent fois plus vite et stockons cent fois plus de données qu'il y a dix ans, et les effets sont si criants qu'on s'aperçoit de la limite des explications de notre maîtresse d'école qui nous enseignait que la quantité ne faisait pas la qualité, car ce saut quantitatif a bien

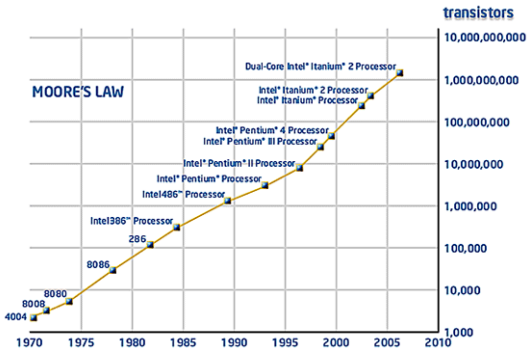
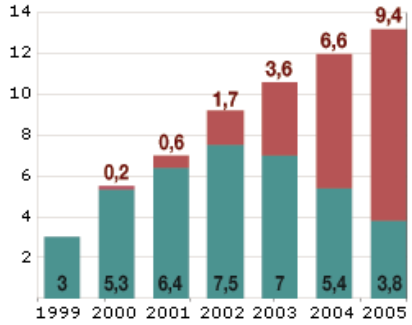


FIGURE 6 – Transistors sur un chip.

Année	ADSL Rx	ADSL Tx
1998	256Kb	128 Kb
2007	28Mb	1 Mb

(a) Débit du reseau



(b) Foyers connectés en France, en millions (donnéées JdN, haut debit en rouge)

FIGURE 7 – Croissance du reseau

impliqué un saut qualitatif. Ce sont deux ordres de grandeurs en dix ans, mais *sur tous les axes.*

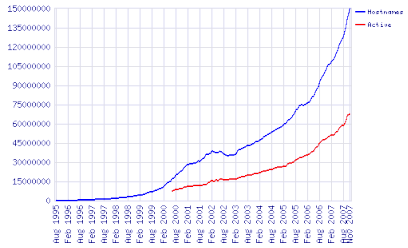
Rappelons-nous que les idées fondamentales étaient déjà là, elles attendaient simplement que la technologie soit en mesure de les exploiter. Mais la massification de l'ensemble de ces technologies pose de nouveaux défis éthiques, technologiques et organisationnels que nous nous devons, en tant qu'informaticiens et citoyens, de relever.

2 Défis éthiques

Dans l'histoire de la recherche, si on regarde bien, on s'aperçoit que l'échange scientifique a été toujours *tout aussi important* que la possession des connaissances : s'il était essentiel d'avoir une grande bibliothèque à Alexandrie, il était

Année	2.5in	3.5in
1997	3Gb	10Gb
2007	250Gb	1000Gb

(a) Capacité des disques



(b) Sites web (Netcraft)

FIGURE 8 – Croissance des donnéés

aussi indispensable de voyager pour être au courant de ce qui se faisait de nouveau ailleurs, et qui, forcement, ne se trouvait pas encore dans la grande bibliothèque.

Mais dans l'époque de changement ultrarapides dans laquelle nous vivons, il est devenu évident que l'*échange*, maintenant, *prime* sur la *possession*, et cela est devenu vrai *pour tous*, pas seulement pour les scientifiques.
Si vous n'êtes pas convaincus que l'échange des données prime sur leur possession, faites le test de déconnecter votre ordinateur et voyez combien de temps vous tenez sans être connectés à un réseau. Je peux vous dire qu'en ce qui me concerne, j'ai du mal à tenir plus de quelques minutes.

Ce bouleversement inattendu met à mal bon nombre de modèles économiques industriels, à l'instar de celui, rélativement récent, de l'économie de la culture qui freine des quatre fers face aux évolutions en cours [2].

Autre nouveauté de taille : l'utilité immédiate prime souvent, malheureusement, sur les approches réflechies, en raison de l'évolution rapide des technologies. Je me rappelle m'être demandé il y a plus de dix ans comment les premiers moteurs de recherche indexaient le web ; pas comment ils présentent les résultats des recherches, ce qui a été l'objet d'une vaste litérature, mais comment ils mantiennent à jour leur image de ce que c'est le Web : un moteur de recherche qui n'est pas au courant des changements des pages intervenus récemment n'est pas très utile.

Je me suis aperçu que les serveurs arpentaient en permanence la totalité des sites web pour surveiller les mises à jour, d'une manière similaire à celle des micro-ordinateurs des années 70, dont les micro-processeurs interrogeaient en permanence les claviers, touche par touche, pour détecter une éventuelle activité (cette méthode est connue sous le nom de *polling*).

Ce système dispendieux et fastidieux a été remplacé il y a longtemps, du moins pour les claviers, par un mécanisme énormement plus efficace : il s'agit tout simplement d'inverser les obligations, et demander au clavier de signaler

2. Lire à ce sujet le "Manifesto pour une création artistique libre dans un Internet libre.", disponible sur `http://www.dicosmo.org/Books`

```
66.249.66.139 - - [10/Jan/2009 :23 :56 :03 +0100] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.71.231 - - [11/Feb/2009 :20 :10 :05 +0100] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.72.197 - - [05/Apr/2009 :06 :49 :14 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.72.197 - - [07/Apr/2009 :04 :26 :32 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.72.197 - - [08/Apr/2009 :18 :47 :34 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.72.197 - - [12/Apr/2009 :09 :06 :14 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.72.197 - - [17/Apr/2009 :03 :15 :31 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.66.82 - - [20/Jun/2009 :21 :37 :43 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.66.82 - - [23/Jun/2009 :03 :54 :23 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.66.82 - - [27/Jun/2009 :20 :41 :48 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.66.82 - - [02/Jul/2009 :10 :52 :52 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.66.82 - - [06/Jul/2009 :12 :37 :20 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.66.82 - - [10/Jul/2009 :21 :57 :01 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.71.108 - - [23/Sep/2009 :17 :12 :55 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.71.108 - - [27/Sep/2009 :14 :20 :30 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.71.117 - - [02/Oct/2009 :16 :01 :48 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.71.117 - - [03/Oct/2009 :23 :53 :22 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.71.117 - - [10/Oct/2009 :17 :06 :43 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.65.4 - - [16/Oct/2009 :19 :52 :15 +0200] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.65.1 - - [25/Oct/2009 :08 :32 :10 +0100] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.65.1 - - [26/Oct/2009 :22 :27 :16 +0100] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.71.117 - - [28/Oct/2009 :01 :53 :00 +0100] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 200 3287 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.71.24 - - [18/Nov/2009 :00 :02 :11 +0100] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
66.249.65.118 - - [17/Dec/2009 :20 :11 :47 +0100] "GET /CourseNotes/Compilation/9900/Cours03/Slides01.html
  HTTP/1.1" 304 - "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http ://www.google.com/bot.html)"
```

FIGURE 9 – Quelques téléchargements inutiles (*polling*) d'une page web par Googlebot

au microprocesseur (par le biais d'une *interruption*) le fait qu'une touche a été pressée ; le reste du temps, le micro-processeur est libre de s'atteler à des tâches plus nobles.

Pourtant, l'indexation des sites web, en 2010, n'utilise toujours pas les interruptions, mais est restée au *polling*, avec un énorme gaspillage de ressources (réseau, et CPU) ; et c'est un gaspillage qui n'a rien de virtuel : les ordinateurs consomment bien de l'énergie.

Si je regarde la trace des accès faits sur `http://www.dicosmo.org`, je vois que, par exemple, Googlebot télécharge en moyenne une fois toutes les sémaine tous les fichiers de mon site web (intégralement, il ne se gène pas à essayer un HEAD pour voir), y compris ceux qui n'ont pas changé depuis 10 ans, comme on peut voir dans la Figure 9. Ces téléchargements inutiles occupent de la bande passante, remplissent mes disques de logs, compliquent les mésures d'audience, et gaspillent de l'énergie sur mes serveurs tout autant que sur ceux de Google.

Ce n'est pas faute d'avoir identifié le problème (j'avais rédigé en 1998 un

draft soumis à l'IETF, qui n'a jamais été finalisé, pour traiter ce problème `http://tools.ietf.org/id/draft-rfced-exp-cosmo-00.txt`) : la raison en est que le système actuel fonctionne *assez bien*[3] et a été vendu à une grande population pas très regardante sur son efficacité, parce-que mal formée pour en comprendre les enjeux.

Un économiste pourraît aussi voir là l'effet d'une externalité : mettre en place un mécanisme comme celui indiqué dans ma proposition de 1998 demanderait un effort collaboratif entre moteurs de recherche et sites web, alors que le système inefficace actuel gaspille énormement de ressources *communes*, qui ne sont pas perceptibles comme coût significatif par les acteurs isolés[4].

Robin Milner, dans son exposé donné pour les 40 ans de l'Inria en décembre 2007, avait justement remarqué combien peut être grande la différence entre « faire des choses qui fonctionnent (plus ou moins)» (l'ingénierie, il disait) et « faire les choses bien» (la science, selon lui) ; ce gaspillage injustifié des ressources par les moteurs de recherche n'est somme tout qu'un exemple parmi d'autres : nous assistons sans vraiment réfléchir à une refonte globale de l'organisation, de l'économie, de la société et de la culture.

Des notions anciennes comme l'anonymat ou le respect de l'individu et de sa vie privée sont mises à mal par l'utilisation, volontaire, de sites comme Facebook, LinkedIn, Flickr, Youtube et d'autres. Il y a plusieurs raisons à cette acceptation volontaire de renoncer aux acquis fondamentaux, mais la plus évidente est que l'utilité *immédiate* de ces services *prime* sur la reflexion plus profonde qui sérait nécéssaire pour arriver à la construction *réfléchie* de méchanismes qui prennent en compte le respet de la vie privée, et les données personnelles se retrouvent par voie de conséquence diffusées sans qu'aucune réflexion n'ait lieu.

Pourtant, ce manque de réflexion n'est pas dû à un manque d'intérêt scientifique de la question : il y a environ 15 ans, avec des amis et collègues, on avait proposé un projet européen nommé *GASP : Guaranteed Anonymous Security Protocols*, dans lequel on proposait de s'attaquer au défi fondationnel du contrôle, par les propriétaires des données, de l'usage qu'un tiers pouvait faire de ces données sur une machine distante. Le projet fut réfusé avec des rapport qui indiquaient qu'il n'y avait aucun *business interest* dans cette ligne de recherche. Ce n'était pas faux : les champions du Web 2.0 nous montrent chaque jour que leur *business interest* est de concentrer dans les mains de quelques acteurs une grande masse de données personnelles pour en extraire de la valeur marchande, et pour cela il faut empêcher toute maîtrise des données de la part de l'utilisateur. La myopie des financements de la recherche à court terme nous a déjà fait

3. Good enough, comme on dit chez nos amis anglosaxons.

4. Depuis 2006, un protocole push/pull qui ressemble de loin à ma proposition de 1998 a commencé à être utilisé : un (administrateur de) site web peut soumettre une description du site au moteur de recherche dans un fichier `sitemap.xml`, qui peut contenir des informations sur la fréquence des mises à jour de certaines parties du site : cette approche est interessante pour les sites dynamiques corporate, comme ceux des journaux et autres média, mais est encore loin de resoudre le problème général pour des sites qui n'ont pas de politique de publication précise, comme ceux de la plupart d'entre nous. Plus d'information sur `http://sitemaps.org`.

perdre énormement de temps : l'étude des propriétés fondamentales de notions essentielles comme le *pseudonymat*, ou la compléxité algorithmique de la confidentialité des données, n'ont pas reçu l'attention qu'elles méritaient, et nous laissent démunis à l'heure ou la société toute entière a besoin de nous.

C'est bien tardivement que l'on commence à voir émerger un peu d'intérêt politique sur ces questions, à travers des concepts comme *privacy by design* et *privacy by default*, qui sont certainement une avancée, mais ne remplacent pas les 15 années de recherche perdues jusque là [5].

Le manque de reconnaissance de l'Informatique en tant que Science a aussi fait qu'on depense des sommes considérables dans des programmes dits de "alphabétisation informatique", qui se limitent à apprendre à nos enfants comment utiliser une souris, un traitement de texte ou à la limite un tableur, alors que la revolution qui est en marche nécessite que tout le monde connaisse les fondement de l'Informatique en tant que Science, pour demystifier les ordinateurs, les algorithmes, les logiciels et les réseaux.

L'exemple du million et demi d'électeurs qui ont *voté sur des machines électroniques* en mai 2007 en France est à ce titre probant. Des journalistes m'avaient demandé à cette occasion de leur montrer en quelques secondes de séquence télévisée comment une telle machine pouvait être piratée. Je leur avais rétorqué qu'il ne leur viendrait pas à l'idée de solliciter un professeur de physique des matériaux pour vérifier si les urnes en plexiglas posaient un problème.

Devant la nouveauté de l'objet informatique, habillé de mystère quand cela convient aux vendeurs, leur sens critique avait abdiqué.

Une autre notion qui est transformée par la généralisation des TIC est celle de *la construction de la confiance*. Les informaticiens ont leur mot à dire sur le sujet car la confiance repose en partie sur des aspects algorithmiques : la confiance entre deux personnes est d'autant plus forte que leur histoire commune est importante. Comment transposer cette confiance à l'échelle informatique d'un réseau de plusieurs millions d'utilisateurs qui ne se connaissent pas ?

La mesure de la *qualité* dans cette nouvelle échelle représente à son tour un défi. En effet, évaluer la qualité d'un article scientifique parmi des milliers de publications sur le même sujet dépasse les capacités d'un individu. De la même manière, une recherche sur Google avec les mots-clés « chaussures de sport » vous renvoie au site Internet de Nike. Cela ne signifie pas pour autant que la qualité des chaussures de sport produites par Nike est la meilleure. Cela implique simplement que l'algorithme de recherche a déterminé que ce site Internet était le plus pertinent, pour lui, en réponse aux mots-clés recherchés ; mais nous ne connaissons pas l'intégralité de l'algorithme, ni ses critères de sélection des sites.

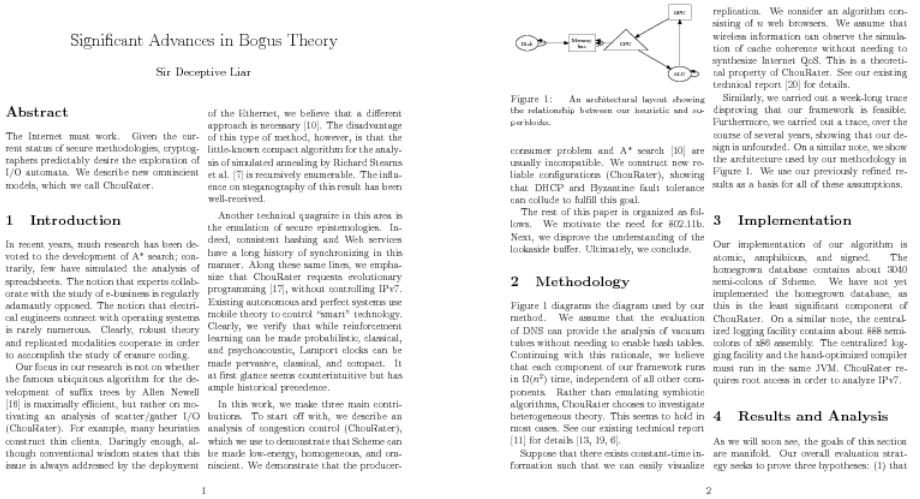
La croissance exponentielle que nous vivons sur tous les axes a augmenté énormement la complexité des informations dont nous disposons, sans pour au-

5. Il s'agit de concepts avancés par Mme Kavoukian, au Canada, et repris par Mme Redding en Europe, voir `http://privacybydesign.ca` et `http://europa.eu/rapid/pressReleasesAction.do?reference=SPEECH/11/183`.

tant qu'on dispose des outils appropriés pour en extraire du sens : en conséquence, il est de plus en plus facile d'induire l'utilisateur en erreur.

Comme un exemple vaut mille discours, voyons le cas de SCIgen (`http://pdos.csail.mit.edu/scigen/`), qui se prête merveilleusement bien à la démonstration.

SCIgen est un logiciel qui utilise des grammaires conçues pour la synthèse du langage naturel pour produire automatiquement et aléatoirement des articles d'apparence scientifique, en quelques secondes. Des étudiants l'ont développé voilà cinq ans pour dénoncer une conférence scientifique, que je ne nommerai pas, qui acceptait n'importe quel type d'articles. Dans la figure 10, on voit un aperçu d'un article produit à la volée juste au milieu de la conférence donnée pour les 40 ans de l'Inria.



Significant Advances in Bogus Theory

Sir Deceptive Liar

Abstract

The Internet must work. Given the current status of secure methodologies, cryptographers predictably desire the exploration of I/O automata. We describe new omniscient models, which we call ChouRater.

1 Introduction

In recent years, much research has been devoted to the development of A* search; contrarily, few have simulated the analysis of spreadsheets. The notion that experts collaborate with the study of e-business is regularly adamantly opposed. The notion that electrical engineers connect with operating systems is rarely numerous. Clearly, robust theory and replicated modalities cooperate in order to accomplish the study of erasure coding.

Our focus in our research is not on whether the famous ubiquitous algorithm for the development of suffix trees by Allen Newell [18] is maximally efficient, but rather on motivating an analysis of scatter/gather I/O (ChouRater). For example, many heuristics construct thin clients. Daringly enough, although conventional wisdom states that this issue is always addressed by the deployment of the Ethernet, we believe that a different approach is necessary [10]. The disadvantage of this type of method, however, is that the little-known compact algorithm for the analysis of simulated annealing by Richard Stearns et al. [7] is recursively enumerable. The influence on steganography of this result has been well-received.

Another technical quagmire in this area is the emulation of secure epistemologies. Indeed, consistent hashing and Web services have a long history of synchronizing in this manner. Along these same lines, we emphasize that ChouRater requests evolutionary programming [17], without controlling IPv7. Existing autonomous and perfect systems use mobile theory to control "smart" technology. Clearly, we verify that while reinforcement learning can be made probabilistic, classical, and psychoacoustic, Lamport clocks can be made pervasive, classical, and compact. It at first glance seems counterintuitive but has ample historical precedence.

In this work, we make three main contributions. To start off with, we describe an analysis of congestion control (ChouRater), which we use to demonstrate that Scheme can be made low-energy, homogeneous, and omniscient. We demonstrate that the producer-consumer problem and A* search [10] are usually incompatible. We construct new reliable configurations (ChouRater), showing that DHCP and Byzantine fault tolerance can collude to fulfill this goal.

The rest of this paper is organized as follows. We motivate the need for 802.11b. Next, we disprove the understanding of the lookaside buffer. Ultimately, we conclude.

1

Figure 1: An architectural layout showing the relationship between our heuristic and superblocks.

2 Methodology

Figure 1 diagrams the diagram used by our method. We assume that the evaluation of DNS can provide the analysis of vacuum tubes without needing to enable hash tables. Continuing with this rationale, we believe that each component of our framework runs in $\Omega(n^2)$ time, independent of all other components. Rather than emulating symbiotic algorithms, ChouRater chooses to investigate heterogeneous theory. This seems to hold in most cases. See our existing technical report [11] for details [13, 19, 6].

Suppose that there exists constant-time information such that we can easily visualize replication. We consider an algorithm consisting of n web browsers. We assume that wireless information can observe the simulation of cache coherence without needing to synthesize Internet QoS. This is a theoretical property of ChouRater. See our existing technical report [20] for details.

Similarly, we carried out a week-long trace disproving that our framework is feasible. Furthermore, we carried out a trace, over the course of several years, showing that our design is unfounded. On a similar note, we show the architecture used by our methodology in Figure 1. We use our previously refined results as a basis for all of these assumptions.

3 Implementation

Our implementation of our algorithm is atomic, amphibious, and signed. The homegrown database contains about 3040 semi-colons of Scheme. We have not yet implemented the homegrown database, as this is the least significant component of ChouRater. On a similar note, the centralized logging facility contains about 888 semi-colons of x86 assembly. The centralized logging facility and the hand-optimized compiler must run in the same JVM. ChouRater requires root access in order to analyze IPv7.

4 Results and Analysis

As we will soon see, the goals of this section are manifold. Our overall evaluation strategy seeks to prove three hypotheses: (1) that

2

FIGURE 10 – Un vrai faux article scientifique

Si l'expérience peut prêter à sourire, on devrait être au contraire très inquiets devant cette énorme facilité à fabriquer des faux paraissant vrais.

Dans un roman de science-fiction d'Isaac Asimov, on trouve une préfiguration de tout cela : lorsque la planète A decidait d'attaquer la planète B, elle était tenue, tout de même, à respecter le protocole diplomatique, qui était lourd et complexe. On écrivait alors sur une tablette l'essence du message (dans le

genre, “on debarque chez vous demain à 8h”), puit on mettait la tablette dans la 'machine protocolaire' qui en sortait un document de 400 pages avec tout ce qu'il fallait. Sur la planète B, on recevait le document de 400 page, on le mettait dans la machine protocolaire, à l'envers, et on en sortait la petite phrase qui l'informait de l'attaque.

Avec SCIgen, on n'est pas loin d'avoir fait une partie du travail pour avoir la machine protocolaire (si bien, que des papiers bidon ont même été acceptés dans des conférences qui sont indexées dans les différents Citation Index). Et d'ailleurs, dans le monde scientifique, comme on nous oblige souvent à passer plus de temps à rediger des gros dossiers pour obtenir des financements, qu'à faire vraiment de la recherche, il ne serait peut-être pas mal qu'on se mette à travailler sur un générateur de propositions de projets.

Mais le vrai problème est que si SCIgen représente un des sens de la machine protocolaire, l'autre sens n'existe pas : comment faire, alors, pour distinguer le faux du vrai dans cette masse de documents ?

Et comment faire pour aider nos concitoyens à se former une opinion, les hommes politiques à prendre des décisions, les marchés à fonctionner, les électeurs à avoir confiance dans leur vote, dés le moment que nous avons contribué à créer une technologie qui fait perdre tous les répères, en rendant facile la production de faux, la manipulation des opinions, la recolte massive des données personnelles, la falsification des votes, pour n'en citer que quelques uns ?

Trouver une reponse à ces défis, et convaincre largement nos collègues à y travailler est quelque chose qui me tient vraiement à coeur, et cela peut se comprendre mieux en sachant qu'à l'orgine, mes études étaient plutôt de nature littéraire que scientifique [6], avec une composante importante d'humanisme, allant des langues anciennes, à l'histoire et à la philosophie.

C'est avec l'argent gagné dans un concours de traduction du Grec au Latin [7], que j'avais pu acheter un ZX80 [8], et je n'ai pas quitté l'informatique depuis : cette science merveilleuse me donnait une nouvelle liberté, car, contrairement à la physique, elle permet de construir son propre monde, et contrairement aux mathématiques traditionnelles, elle permet ensuite de lui donner une vie concrète, en exécutant les programmes.

Je pensais en ces premiers temps que dans l'informatique un scientifique n'aurait pas a se poser les problèmes éthiques qui avaient hanté d'autres disciplines, mais c'était bien naïf que de croire cela.

Les pères fondateurs de la physique moderne, travaillant sur l'énergie nu-

6. En Italie, la sélection ne se faisait pas par les mathématiques, à mon époque, mais par le grec et le latin.

7. Le *Certamen Classicum Florentinum*, pour les curieux.

8. Une sorte de micro-ordinateur minimaliste produit par Sinclair, avec 1Kb de RAM, partagé entre mémoire programme et mémoire vidéo, ce qui posait des défis intéressants : plus long était le programme, et moins d'espace on avait pour en afficher les résultats.

cléaire, avaient à se demander s'il ne contribuaient en réalité à la destruction de notre monde.

Quand j'étais sur les bancs de l'Université à Pise la tragèdie de Seveso était vive dans les esprits, et mes collègues qui faisaient des études de chimie rasaient les murs.

Nos collègues mathématiciens ne sont pas mieux lotis aujourd'hui : après l'ébriété provoquée par le succès des mathématiques financières, qui permettaient d'attirer des très bons étudiants vers une discipline en perte de vitesse, en leur vendant le rêve de maîtriser totalement le risque financier, quelle geule de bois, en découvrant, au réveil, qu'ils avaient contribué à fournir aux apprentis sorciers de la finance les armes avec lesquelles ils ont presque détruit l'économie réélle, et les espoirs, et la vie de tellement de gens !

En tant qu'informaticiens, il nous revient aujourd'hui d'apprendre des erreurs des autres, et de rélever sans hésitation les défis éthiques fondamentaux que notre discipline pose à notre société : la confidentialité, l'anonymat, la sécurité, la mâitrise de la complexité induite par le pouvoir de calcul, stoquage et échange que nous avons créé, pour n'en citer que quelquesuns, ne sont pas *seulement* des sujets pour des papiers scientifiques ; l'obtention d'une bourse, un prix ou un financement de la part de tel ou tel autre géant industriel de l'informatique en échange de nos connaissances, de notre crédibilité, de notre silence, ou d'un changement d'orientation de notre activité de recherche, ce n'est pas *toujours* une bonne chose.

En tant qu'êtres humains, nous nous devons de ne pas nous cantonner aux simples aspets techniques et scientifiques de notre métier : nous dévons en assumer tous les aspets sociétaux aussi.

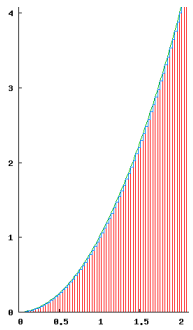
3 Défis technologiques

Le besoin de maîtriser la complexité de l'univers qui nous entoure est un besoin ancestrale : depuis les temps les plus anciens, les hommes rêvent de *formules magiques* qui permettent, avec quelques incantations arcanes, mais concises, de produire de grands effets.

Courtes formules, grands effets, cela n'est pas très loin de ce qu'on appelle aujourd'hui des *abstractions*, et l'histoire des mathématiques peut bien être vue comme le récit d'une quête millénaire à la recherche d'abstractions puissantes qui permettent de capturer en quelques formules concises la solution à des problèmes de plus en plus complexes. Ce n'est d'ailleurs pas étonnant de voir la fascination que les mathématiques ont suscité pendant des long siècles : il s'est agit, fort longtemps, de la seule forme de *magie* qui s'avérait *vérifiablement* efficace.

Avec les siècles, les mathématiques ont énormement évolué : le calcul de la surface délimitée par une courbe quadratique, qui avait demandé le génie

d'Archimede il y a plus de 2000 ans, est aujourd'hui à la portée de tous



$$A = \int_0^2 x^2 dx = \frac{8}{3}$$

L'Informatique partage avec les Mathématiques [9] ce défi permanent de l'abstraction toujours plus poussée, et il suffit de comparer le code qu'on utilisait pour manipuler des listes dans les années '60 avec celui devenu populaire dans les années '90 pour s'en rendre compte.

Pascal

```
type pointer = ^cell;
     cell = record
       Next: pointer; Data: integer;
     end;
function len(s: pointer): integer;
var cur: pointer;tmp: integer;
begin
  if s = nil then begin len := 0; Exit; end;
  tmp := 1; cur := s;
  while (cur^.Next <> nil) do
  begin
    tmp := tmp +1; cur := cur^.Next;
  end;
  len := tmp;
end;
```

OCaml

```
type 'a list = Nil | Cons of 'a * 'a list;;
let rec len = function Nil -> 0
              | Cons (_,r) -> 1+(len r);;
```

9. En réalité, je crois fermement que toutes les sciences, exactes ou humaines, font face au défi du langage, mais cela est bien plus visible dans les Mathématiques et l'Informatique qu'ailleurs.

L'objectif commun est de développer des outils plus concis, plus précis et plus puissants qui permettent d'écrire en quelques lignes ce qui prenait avant des pages et des pages de calculs.

Pour parvenir à ce résultat, il est nécessaire de prendre du recul et de se donner le temps de faire les choses bien. La recherche fondamentale et à long terme est en ce sens un élément *essentiel* de l'évolution de notre savoir, qu'on peut ne pas savoir mésurer, mais sans laquelle, rien n'est possible.

Mais notre discipline, l'Informatique a une particularité qui me paraît important de souligner : la *vitesse* à laquelle grandit la complexité des artifacts qui sont notre objet d'étude, et pour maîtriser laquelle nous créons des nouveaux artifacts, qui rajoutent à leur tour à la complexité du tout.

On a déjà vu comment en 10 ans nous avons assistés à une croissance de *deux ordres de grandeurs* sur tous les éléments de base des technologies informatiques : microprocesseurs, réseau, espace de stoquage, nombre de documents disponibles ; cela a bien évidemment un impact direct sur l'efficacité, et l'accéleration des découvertes, dans toutes les disciplines.

Mais pour nous Informaticiens, il est important de s'intéresser aux problèmes posés par la croissance des artifacts informatiques par excellence, les logiciels, et ici la généralisation du Logiciel Libre à laquelle on assiste depuis quinze ans a un rôle majeur à jouer.

3.1 Les défis du Logiciel Libre

Un logiciel *libre* est un logiciel dont les termes d'usage donnent aux utilisateurs quatre droits fondamentaux : la liberté, sans restriction, de se servir du logiciel ; la liberté de modifier le code du logiciel (et donc de disposer des sources du logiciel) ; la liberté de donner ce transmettre ce logiciel ; et la libertè de le transmettre avec ses sources, même modifiéés [10].

Ces quatre caractéristiques du logiciel libre, qui sont à l'exacte opposé des caractéristiques des logiciels propriétaires, peuvent à première vue paraître techniquement irrélevantes : après tout, un logiciel libre est avant tout un logiciel, et les conditions de distributions permises par la licence ne devraient avoir d'impact que sur son modèle économique.

Evidemment, l'impact disruptif du logiciel libre sur le modèle économique traditionnel des logiciels éditeurs est tout à fait majeur. L'industrie du logiciel a vecu des décennies à sur la vente de licence, c'est à dire le fait de faire payer la seule chose qui ne coute rien, la copie d'un logiciel déjà existant.

C'est bien une économie byzarre, fondée sur une raréte artificielle construite à coup de restrictions juridiques, et que les acteurs traditionnels cherchent à maintenir à tout prix, en utilisant non seulement le droit d'auteur, mais aussi le droit des brévets [11].

10. En anglais Free Software et Open Source, même si avec des différences phylosophiques importantes, recouvrent approximativement l'ensemble des Logiciels Libres.

11. Ce qui est particulièrement inapproprié ; mon point de vue sur la question se trouve dans l'article : `http://www.upgrade-cepis.org/issues/2003/3/up4-3DiCosmo.pdf`.

En rayant de la carte la vente de licences, le logiciel libre permet de faire apparaîtres les vraies ressources rares sur lesquelles on peut baser une économie saine

Mais il serait très naïf de s'arrêter là.

Des exemples de logiciels libres existent depuis longtemps, mais le logiciel libre n'a véritablement pris son essor qu'avec le développement du réseau, qui a permis de concrétiser leur potentiel de dévéloppement collaboratif, et souvent distribué. Leur cycle de développement est dévenu plus rapide et, comme il permet de reprendre du code existant pour le modifier et l'améliorer, les interdépendances entre les logiciels libres peuvent être très fortes.

De plus, nous avons pour la première fois de l'histoire accès non seulement à l'ensemble du code d'une immense quantité de logiciels, mais aussi à la trace complète de ce code (auteur, modifications, date de modification, etc.) Cet accès offre un « terrain de jeu » infini à qui désire étudier le phénomène, car absolument tout est exposé sur le web.

Nous y trouvons d'ailleurs des logiciels très complexes, comme cette représentation du noyau linux qui tourne sur mon ordinateur, que vous apercevez en ce moment. Grâce au développement collaboratif, cet exemple de modélisation très complexe a acquis une remarquable stabilité.

L'image suivante illustre le GNU BG, un paquet logiciel qui permet de jouer au backgammon sous Linux. Si vous souhaitez jouer à ce jeu, le logiciel va déterminer les librairies dont il aura besoin pour vous permettre de jouer. Il m'indique en l'occurrence que j'ai besoin de deux librairies. Or elles-mêmes nécessitent d'autres composantes.

Finalement il est possible de très rapidement se retrouver avec un enchevêtrement de paquets à installer, d'où l'aspect « tas de spaghettis » de l'image. Le plus génial dans le monde libre, ou le plus terrifiant, est que ce tas de spaghettis change tous les jours. En effet, chacun des paquets peut être modifié par n'importe quel utilisateur sans que cette intervention ne nécessite une autorisation. Ce type de système préfigure les systèmes complexes de demain, à savoir des machines dont les composantes logicielles changent en permanence. Evidemment, le propre de l'informatique est de cacher la complexité des actions à entreprendre derrière une couche marketing et une interface graphique merveilleuses. Le cas de l'icône des mises à jour de système est éloquent. Le système nous informe que des mises à jour sont prêtes à être installées sur notre ordinateur et qu'il nous suffit de cliquer sur « oui » pour les installer et ainsi améliorer notre système. La réalité est éminemment plus complexe comme vous pouvez vous en douter. Le système va tirer une ficelle du tas de spaghettis, puis il essaiera de faire en sorte que tout fonctionne. L'utilisateur réfléchirait sans doute par deux fois s'il était au courant de cette complexité et du risque encouru.

Transparence du logiciel.

4 Défis organisationnels

Notre travail d'informaticien consiste à permettre au système de fonctionner correctement. Or la couche marketing nous empêche parfois de déceler un problème qui vaudrait la peine d'être traité. Il faut donc approfondir nos investigations. Le logiciel est un cas d'école du fait de sa longue existence, des quelques succès remarquables qu'il a rencontrés et de sa capacité à évoluer grâce à des outils collaboratifs. Nous pouvons tirer de cette expérience de nombreux enseignements, pour peu que nous ne soyons plus naïfs comme nous l'avons déjà été. Certaines personnes nous expliquent par exemple que le logiciel se crée et s'améliore tout seul alors que ce n'est évidemment pas le cas. Ce genre de propos a entraîné des comportements naïfs voués à l'échec d'avance.

Martin Michlmayr, étudiant en philosophie titulaire d'un master en systèmes d'information, a développé une thèse sur le fonctionnement du logiciel libre. Il a ensuite réussi à se faire élire comme leader du projet Debian pendant deux ans et coordonnait donc la distribution Linux la plus complexe à l'heure actuelle. Il conclut dans sa thèse que les logiciels libres et propriétaires s'opposent sur deux points. Le logiciel propriétaire est bâti sur le modèle de la construction de cathédrale : un architecte unique conçoit le système avant de répartir les tâches à des exécutants. En revanche, un logiciel libre couronné de succès ne fait que commencer par la « phase cathédrale » : c'est le moment où une personne a l'idée du système et fédère un groupe de personne autour d'elle pour le développer.

Le développement du logiciel passe ensuite par une phase de transition où le code doit être ouvert et mis à disposition du public. Il faut donc à ce stade concevoir un prototype qui facilite l'intervention du public dans le système. La conception du logiciel est ici primordiale. Ce n'est qu'après cette transition que nous arrivons dans la phase du développement distribué et du travail collaboratif sur le logiciel.

En regardant de plus près, nous observons que les groupes d'acteurs qui interviennent sur un projet de logiciel libre sont très différents. Certains maîtrisent l'évolution du code et interviennent plutôt lors de la phase cathédrale quand d'autres n'arrivent que dans la dernière phase que je qualifie de « bazar ». Vérifions au travers de quelques exemples si ce fonctionnement s'applique vraiment dans la réalité.

Wikipedia intègre la modularité et le développement collaboratif. Les communautés qui y interviennent sont très modulaires : certaines personnes se concentrent exclusivement sur l'écriture des articles, pendant que d'autres corrigent uniquement les coquilles et s'en satisfont pleinement. D'autres encore contrôlent les contenus. Il est intéressant de remarquer qu'un système totalement ouvert à l'origine a entraîné l'émergence d'un besoin de contrôle afin d'assurer la qualité du système. Le noyau Linux est un autre exemple digne d'intérêt. Lui aussi est très modulaire : la moitié du code concerne les pilotes de périphériques qui sont mis à jour par de minuscules équipes.

A ce niveau, il est important qu'une équipe centrale définisse l'architecture du système pour que d'autres puisse y intervenir. Malgré sa modularité, une crise majeure est survenue qui a failli entraîner la perte du projet. Les contrôles

centralisés avaient été définis de manières trop strictes, ce qui avait eu pour conséquence de remettre la décision d'autoriser une modification dans les mains d'une seule et même personne, bloquant par là même toute l'évolution du système.

La solution au problème a consisté à relâcher la pression des contrôles centralisés et de créer un « cercle des lieutenants » de l'organisation.

A partir de là, une réflexion doit être menée sur notre façon de créer des documents. Tout le monde connaît les wikis, pourtant de nombreuses personnes demandent encore leurs disponibilités à 400 personnes par mail. Je citerai encore les mises à jour de documents envoyés en pièces jointes avec pour résultat de voir deux personnes modifier une version d'un document en même temps, mais sur deux fichiers différents, avant de les renvoyer à une personne pour qu'elle intègre leurs modifications respectives. La réflexion sur notre organisation du travail est impérative. Je vous concède que ces commentaires sont proches de ceux que pourraient faire les sciences humaines, mais nous sommes bien des êtres humains. En conclusion, nous avons de nombreux défis à relever. Je m'attends à voir une deuxième jeunesse émergée de cet Institut qui fête ses quarante ans, au travers de sujets très intéressants comme l'indexation des textes multimédias, la rapidité et la concision dans les nouvelles applications, la recherche de nouvelles « baguettes magiques », etc. Je souhaite aussi qu'un certain nombre de leçons importantes ne soient pas oubliées. La conception modulaire revêt une importance majeure au niveau de la technologie, au niveau de la conception de fonctionnement de communautés d'acteurs, mais aussi au niveau de la structuration des acteurs économiques. En effet, nous sommes au milieu d'une nouvelle révolution industrielle et il est nécessaire de revoir le rôle de chacun dans cette structure économique. Beaucoup de chemin reste à parcourir comme en attestent les conclusions de cette étude sur la proportion de projets libres importants dont le leadership est européen. Elle nous indique que 45% de ces projets sont de leadership européen et que les Etats-Unis accusent un retard en la matière, mais que la situation est bien différente dès lors qu'il s'agit d'étudier le chiffre d'affaires lié au marché du logiciel libre. L'Europe dispose donc une nouvelle fois du leadership technologique, mais se retrouve incapable de l'exploiter. Pour revenir à la conception, nous sommes très bons lors de la phase « cathédrale », mais très mauvais pendant la phase « bazar ». Il faut corriger ce problème et je pense que nous allons y arriver.