

Logiciels libres et réseaux sociaux de compétences.

Michaël Vicente

Costech- Université de technologies de Compiègne.

[michael.vicente@utc.fr](mailto:michael.vicente@utc.fr)

Version 2.

13 Mai 2008. Ne pas citer.

Ces dernières années le logiciel libre a transformé le modèle économique, d'une partie de l'industrie du logiciel, qui reposait jusqu'alors quasi exclusivement sur la propriété intellectuelle, et la vente de licences logicielles. On pouvait dès lors mesurer la viabilité et la tangibilité de l'activité industrielle, grâce au nombre de licences vendues. L'introduction du logiciel libre au cœur des stratégies de géants de l'informatique tels que IBM, Sun semble remettre en cause ce modèle, dans le sens, où le logiciel dont le code est librement partagé, en devenant librement accessible à tous devient la propriété de personne. Des firmes telles que Red Hat ont totalement dédiés leur activité autour de ces logiciels et emploient des milliers de salariés. Nous faisons l'hypothèse que les développeurs professionnels de logiciels libres sont les principaux acteurs de ce changement, et que l'observation fine de leurs parcours professionnels nous apprend énormément sur leur mobilité mais aussi et surtout sur la mutation du secteur de l'informatique en général. Pour cela nous avons dans cette étude mis en avant les réseaux sociaux de développeurs d'un projet en particulier que nous avons renommé Libre-Serveur, et nous verrons que ces développeurs forment un réseau social de compétences.

### Apprentissage et compétences au coeur du logiciel libre.

Le logiciel libre et notamment la licence qui lui est le plus généralement lié, la GPL (General Public Licence), repose sur quatre libertés, (1) la liberté d'exécuter le logiciel, pour n'importe quel usage (2) la liberté d'étudier le fonctionnement d'un programme et de l'adapter à ses besoins, ce qui passe par l'accès aux codes sources, (3) la liberté de redistribuer des copies, (4) la liberté d'améliorer le programme et de rendre publiques les modifications afin que l'ensemble de la communauté puisse en bénéficier. Certes il existe des variantes au niveau de la licence, licences BSD, MIT, Apache, mais le dénominateur commun pour les licences les plus restrictives comme pour les moins restrictives reste l'obligation de citer l'auteur original. Ainsi, le développeur-auteur appose son nom sur son code via le copyright. Car aussi paradoxal que cela puisse paraître, le logiciel libre engendre dans ce sens un renforcement du copyright pour l'auteur. Celui ci n'étant cependant plus confondu avec celui de l'entreprise

productrice, mais singularisé au niveau du développeur lui même. Ainsi, on peut assigner un auteur à la moindre petite partie de code. Cette notification à l'extrême de la division du travail dans ces systèmes productifs complexes, a un avantage certain, lorsque l'on sait que ces projets, sont portés par des communautés aussi hétérogènes que distantes. Ainsi, lorsque les développeurs utilisent du logiciel libre, et donc le plus souvent personnalisent le code pour l'adapter à leurs propres besoins, s'ils rencontrent des difficultés, ces derniers ont la possibilité de contacter le développeur qui avait la charge du code en question. Il se crée ainsi des réseaux denses de coopération entre des développeurs, qui bien souvent ne se connaissent, qu'à travers leurs pseudo et les réalisations qui y sont associées. La reproduction de ces entraides, est le fondement d'un savoir commun, basé sur les compétences, questions et expertises de chacun. Ce mécanisme est aussi cumulatif dans le sens où l'expérience et les réalisations des autres ne sont jamais perdues, elles laissent toujours des traces, soit dans le code lui même lorsque la transformation est intégrée ou "commitée", soit sous la forme de questions réponses dans les mailing list et forums archivés sur Internet.

Il se forme ainsi, ce que nous appellerons un "réseau de compétences", où l'expertise de chacun, librement disponible peut être utilisée par tous quelle que soit leur localisation, leur appartenance et leurs domaines d'application.

Ce mécanisme n'est pas en soit nouveau pour les sciences cognitives, puisqu'il correspond à ce qu' Hutchins<sub>1</sub> a appelé « cognition distribuée ». C'est à dire que la cognition est partagée au sein des équipes par l'ensemble des membres et par l'ensemble des éléments techniques ou artefacts qu'ils utilisent. Celle ci repose en partie sur l'idée de « Transactive Memory », théorie développée par Werner<sub>2</sub> , qui indique que le savoir est constitué par l'ensemble des connaissances d'un individu et par sa capacité à « savoir qui sait quoi? ». Dans ce contexte, la coopération est plus performante quand les individus sont capables de localiser où se trouve d'information et qui la détient. Dans le cas de la création de logiciel l'ensemble des savoirs et compétences est distribué parmi les développeurs, aucun d'entre eux ne serait capable de développer le logiciel dans son intégralité. Le travail de création de logiciel constitue ainsi une coordination d'expertise comme le notent Faraj et Sproull (2000) , celle ci repose sur la capacité à savoir où se trouve l'expertise mais aussi à reconnaître les besoins d'expertise, ces savoirs étant souvent tacites ( Nonaka 1995 ). Il s'agit aussi

---

1 Hutchins (1995)

2 Werner et Alii ( 1987)

d'imaginer des processus pour opérer ce transfert de connaissance.

Dans le cas de la coopération à l'oeuvre dans le logiciel libre, nous avons un bel exemple de cette cognition distribuée. Nous voyons aussi que l'ensemble des dispositifs techniques ( arbre CVS, mailing lists, forums ) et organisationnels ( processus d'entrée dans les communauté, normes et conventions dans les échanges, valeurs communes autour du partage...) permettent la fixation et la circulation de cette mémoire transactive. . D'un point de vue individuel, toutes ces pratiques se mettent en forme lors du processus d'apprentissage constant qui a lieu lors de la participation à un projet de logiciel libre. D'une manière plus générale, Moreland et Alii (1996) ont ainsi montré que la mémoire transactive se développe davantage lorsque l'apprentissage se fait en commun. Or, c'est exactement ce qu'il se passe dans le cadre du logiciel libre. Ainsi, l'entrée dans les communautés implique le plus souvent un processus d'apprentissage technique où il faut faire ses preuves par l'envoi de « patch » réparant des erreurs logiciels. Dès ces premiers envois ou dès les premières discussions sur les mailing list, les futurs développeurs sont déjà en contact avec des développeurs plus expérimentés. Cet apprentissage à fois technique et culturel, peut se présenter sous le modèle « oignon », qui débute avec l'inscription sur la mailling list, puis l'envoi de patch; suit ensuite un processus de sélection qui peut être plus ou moins long selon les communautés.

Tout au long de cet apprentissage, le développeur accumule ainsi un ensemble de savoirs et des relations avec d'autres développeurs eux-mêmes détenteurs de connaissances spécifiques. Du point de vue de l'entreprise productrice (ou simplement utilisatrice de logiciels libres), on peut supposer que l'enjeu est alors pour elle de faire appel à des personnes qui maîtrisent ces compétences spécifiques.

### L'apparition de développeurs professionnels de logiciels libres.

Les logiciels libres étaient auparavant exclusivement dédiés à des usages complexes, et étaient pour la plupart issus de la recherche académique, ou bien étaient issus du travail de quelques militants passionnés qui "bidouillaient" sur leur temps libre. On a aussi pu voir au sein de certaines professions que le code pouvait être développé sur le temps de travail, ce qui est comparable à la "perruque" dans le monde ouvrier ( Michel

Anteby 2002 ). Certains administrateurs système dont le métier consiste à mettre en place et à maintenir les réseaux internes et externes au sein des entreprises et administrations, ont une activité en pics au gré des pannes, ce qui leur permet de libérer du temps. De plus, dans les années 90, les principaux logiciels libres qui se sont développés étaient des logiciels liés à l'internet en pleine expansion, (Bind, Apache, Samba, Mail, ect...), les administrateurs systèmes se sont retrouvés au coeur de ces développements et sont de ce fait, devenus les premiers utilisateurs de logiciels libres. Ainsi, aujourd'hui, 60 à 70% des serveurs web tournent à partir de logiciels libres.

Du point de vue de l'entreprise, cette diffusion rapide des outils libres parmi les acteurs de l'Internet et de l'industrie, les oblige à traiter avec une ressource rare, " le bon développeur de logiciel libre". La gestion de ces ressources implique différents coûts. Selon la théorie des coûts de transactions, ces coûts dépendent de trois éléments majeurs, la fréquence des transactions, l'opportunisme des acteurs et la spécificité des actifs. La spécificité des actifs y occupe une place essentielle et correspond à des investissements pour supporter des transactions très précises, qui ne sont pas utilisables pour une autre transaction. Les actifs dont il est question ici sont des actifs hautement spécifiques dans le sens où il s'agit de compétences liées à des parties logiciels pointues, et donc à des compétences elles aussi précises. Dans son article "Why Is It So Hard to Find Information Technology Workers?" Peter Cappelli (2001) , montre que même après la bulle de l'Internet, il y a une pénurie certaine de personnels compétents en informatique, et que cela se traduit par les salaires parfois mirobolants des développeurs employés. On se trouve donc dans un marché du travail malthusien. Cette situation dans le marché du travail global des technologies de l'information, est encore plus accentuée dans le sous-domaine spécifique au logiciel libre. Ceci en partie parce qu'il n'y a pas de formation de type universitaire propre au logiciel libre.

Aussi, certains grands groupes tels que IBM, Sun ou d'autres ont investi dans le logiciel libre et l'ont placé au coeur de leurs stratégies. Il en résulte la création d'un véritable marché du travail dédié au logiciel libre. Même si, comme nous le verrons, plusieurs développeurs déjà implantés dans leur entreprise, ont développés des activités liées au logiciel libre.

Des compétences localisées.

Une étude du Merit de 2006 permet de localiser les développeurs de logiciel libre. En observant où ils se trouvent géographiquement, force est d'admettre que l'utopie d'une information librement distribuée n'a pas démocratisé la participation des populations d'un point de vue géographique. ( Voir les cartes p 42 du rapport du Merit 2006)

En effet, les développeurs de logiciels libres sont géographiquement situés là où existent déjà des ressources économiques et universitaires. Il en est ainsi en Europe, notamment sur l'axe Grande-Bretagne-Allemagne-France, mais aussi dans certaines régions des Etats Unis, par exemple San Francisco et sa baie... A l'inverse, il existe encore des zones où le logiciel libre est peu présent et cela ne peut s'expliquer qu'en partie par la fracture numérique, puisque lorsque l'on regarde la proportion de développeurs par connexion Internet, on peut noter que la participation de ces pays reste faible, à l'exception de la Russie, Madagascar, et quelques pays d'Amérique latine. Ces sites sont aussi les lieux où est concentrée la plupart des industries informatiques, y compris les sites émergents comme le sud de l'Inde ou la cote brésilienne. Les nouvelles compétences liées au logiciel libre semblent donc se créer là où d'autres compétences pointues existent déjà. Ce point illustre dans une certaine mesure la professionnalisation que nous avons évoquée précédemment. C'est parce que l'on a la possibilité de participer au libre de manière professionnelle que l'on peut s'y impliquer massivement. Dans un précédent article, nous avons déjà évoqué le fait que la présence d'intérêts privés ne démotivait pas des développeurs bénévoles de participer ; il n'y a pas de crowding out effect<sup>3</sup>, comme on peut le trouver dans d'autres activités bénévoles.

Ainsi, si l'on regarde de plus près le statut des développeurs officiels du logiciel Libre-Serveur, on peut se rendre compte qu'une écrasante majorité a une activité professionnelle liée de près ou de loin au logiciel libre.

### Méthodologie:

C'est parce qu'activité professionnelle et logiciel libre sont extrêmement liés que nous nous sommes attaché à la compréhension des développeurs de logiciel libre professionnels. Dans le cadre de cet article, nous centrerons notre analyse sur les parcours professionnels de développeurs d'un projet en particulier, le projet Libre-

---

3 Voir Vicente ( 2007)

Serveur . Dans le cadre d'un enseignement pas projet à l'Université de Technologie de Compiègne, un élève ingénieur<sup>4</sup>, a permis la reconstruction longitudinale des réseaux sociaux de développeurs à partir du CVS du logiciel. Le CVS (Control Versionning System) est l'outil principal de coordination du travail des développeurs dans l'espace et dans le temps. L'activité des développeurs est conservée par des logs inscrits dans le logiciel. Nous avons donc extrait ces traces, construit des bases de données relationnelles sous forme de matrices d'adjacence, qui nous ont ensuite permis de reconstruire ce réseau social sous forme de graphe.. On a ainsi pu mettre en évidence les liens de coopération entre développeurs en indiquant par des flèches orientées quel est le meilleur collaborateur de chacun. Nous avons ensuite corrélé ces graphes avec les curriculum vitae des développeurs recueillis manuellement sur le web (pages perso, blogs, sites types Linkedin...). Nous sommes ainsi parvenu à reconstruire 96% des parcours professionnels des développeurs.

Nous pouvons ainsi, années par années depuis le début du projet en 1996 jusqu'en décembre 2007, avoir une photographie du réseau de développeurs, avec pour chacun d'entre eux des éléments concernant leur profession, l'entreprise pour laquelle ils travaillaient à l'époque et leur localisation. Cet ensemble de graphes nous permet de reconstruire la dynamique du projet, et les trajectoires professionnelles de chacun.

### **Trajectoires professionnelles et trajectoires d'entreprise au sein d'un projet de logiciel libre.**

Le projet dont il est ici question est un projet majeur du logiciel libre et de l'Internet, puisqu'il s'agit d'un logiciel de serveur installé sur plus de 60% des serveurs Web. Il est issu d'un projet universitaire mené dans un laboratoire d'informatique.

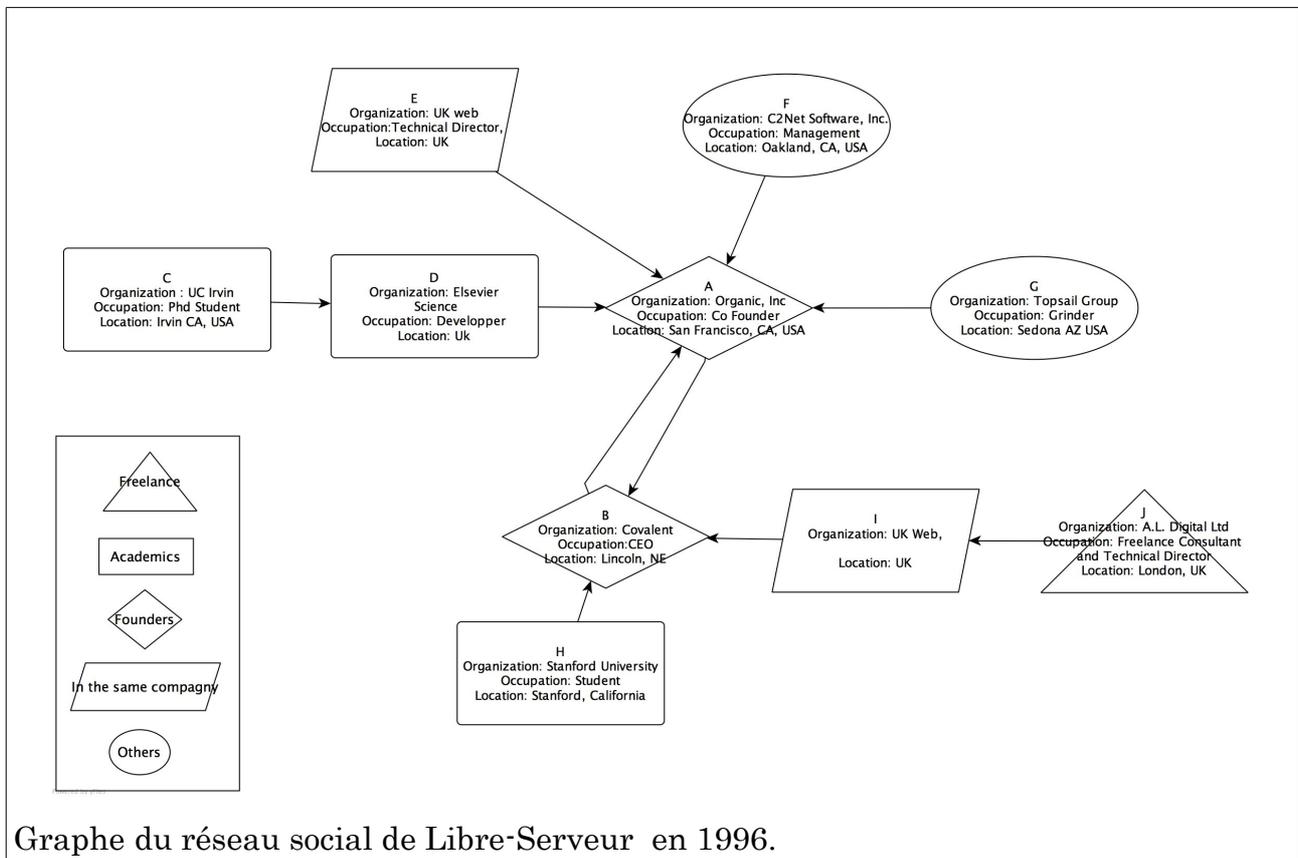
Ainsi, le projet débute par un "fork » (bifurcation), c'est-à-dire une séparation du projet initial en deux autres branches, dont celle qui prendra le nom de Libre-Serveur. Le code est disponible, il est donc tout à fait possible de continuer le développement de celui-ci autour d'une autre équipe ; il s'agit d'une pratique assez courante dans le milieu du logiciel libre.

Au travers de l'étude des graphes, on note que le projet à été initié par A., le co-

---

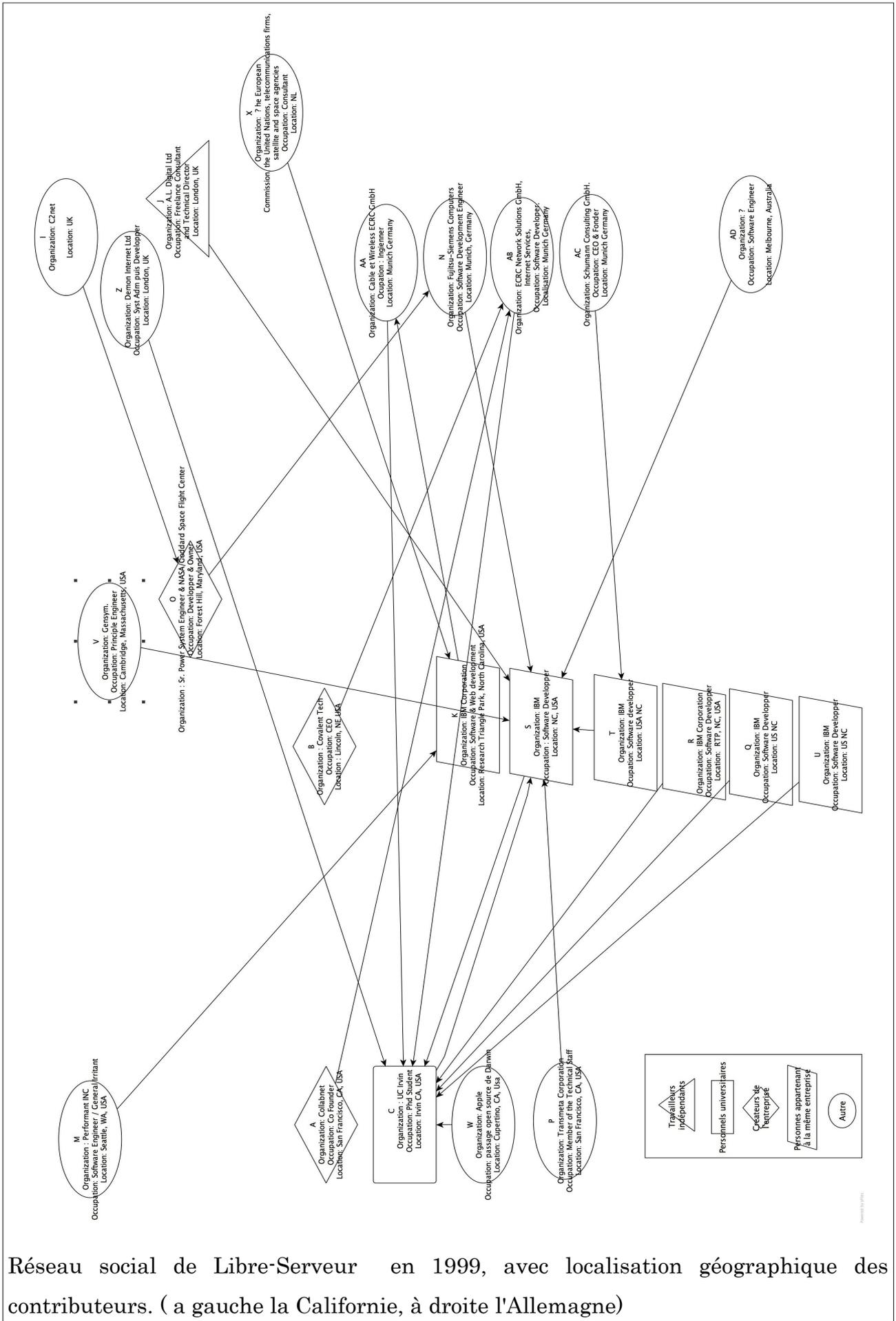
<sup>4</sup> Cet article doit beaucoup à Remi Pallancher à son travail et à son investissement bien au delà le cadre de l'UV, qu'il en soit remercié.

fondateur d'une entreprise de San Francisco liée à l'internet. En 1996, le projet est composé de dix développeurs ; il se retrouve donc "naturellement" au centre du projet, en duo avec B, un autre créateur d'entreprise de Lincoln dans le Nebraska et qui, pour sa part, était impliqué dans le logiciel libre depuis 1989. Aussi, dès le début du projet, deux personnes E. et I. travaillent pour une même entreprise à Leeds au Royaume-Uni, spécialisée dans les serveurs Internet. Cette société n'aura eu qu'un faible succès puisqu'elle n'a existé qu'une seule année. Parmi les autres participants, on retrouve un doctorant de l'Université de Californie à Irvin C. et un étudiant de Stanford, H., (universités toutes deux situées en Californie), ainsi qu'une personne travaillant pour l'édition scientifique. Parmi les autres personnes, on identifie F., le gérant d'une petite entreprise d'Oakland, ville voisine de San Francisco.



L'année 1997 voit l'arrivée de nouveaux développeurs, ce qui porte leur nombre à 14, et l'on retrouve au centre du réseau un nouvel entrant K, travaillant chez IBM depuis plusieurs années et qui est connu pour avoir intégré et influencé la politique *Open Source* au sein de la firme. En 1996, on peut noter que K n'a pas signé son code avec une adresse mentionnant son appartenance à IBM, ce qui suppose qu'il réalise ce travail bénévolement. En 1999, alors que le nombre de développeurs est passé à 24,

cinq nouveaux employés d'IBM sont entrés dans le projet et K se voit remplacé au centre du projet par le thésard de UC Irvin et par un de ces collègues d'IBM. Cette date correspond à l'annonce officielle de l'implication d'IBM dans le logiciel libre et dans ce projet en particulier. Le fait que K soit présent dès 1997 conforte la thèse selon laquelle l'*open source* s'est initialement développé dans les entreprises via ses employés. Ce sont eux qui ont imposé l'*open source* à leurs entreprises et non l'inverse. Il faut cependant noter que parmi les cinq nouveaux membres travaillant chez IBM, quatre d'entre eux n'étaient pas présents au sein de la communauté avant d'être employés par IBM. Seul R s'est intéressé au projet deux ans avant son entrée chez IBM. Les autres ont quasiment tous la particularité d'avoir été recrutés par IBM un ou deux ans auparavant, à la sortie de l'université. Ces personnes ont donc été initiées au logiciel et au logiciel libre au sein de cette entreprise. Les six développeurs IBM sont implantés sur le même site d'IBM en Caroline du nord, à Raleigh, là où se trouve le IBM Research Triangle Park.



Il semble particulièrement intéressant de suivre la trajectoire de deux ces employés, à la fois au sein du projet et dans leur carrière professionnelle.

### Les cas de T. et S.

T a été recruté en 1997 par IBM à la sortie de son Master en Information Networking à l'université Carnegie Mellon. Il a donc quitté la Pennsylvanie pour rejoindre la Caroline du Nord, région qui ne lui est pas inconnue, puisqu'il avait auparavant validé son Bachelor à l'université d'Etat de Caroline du Nord. Il n'avait auparavant aucun lien avec le logiciel Libre-Serveur, puisque son premier post sur le projet date du 31 octobre 1997. T devra cependant attendre 1998 pour intégrer le projet en tant que « official commiter ». A partir de ce moment-là, il aura des liens privilégiés avec K, première personne d'IBM présente sur le projet.

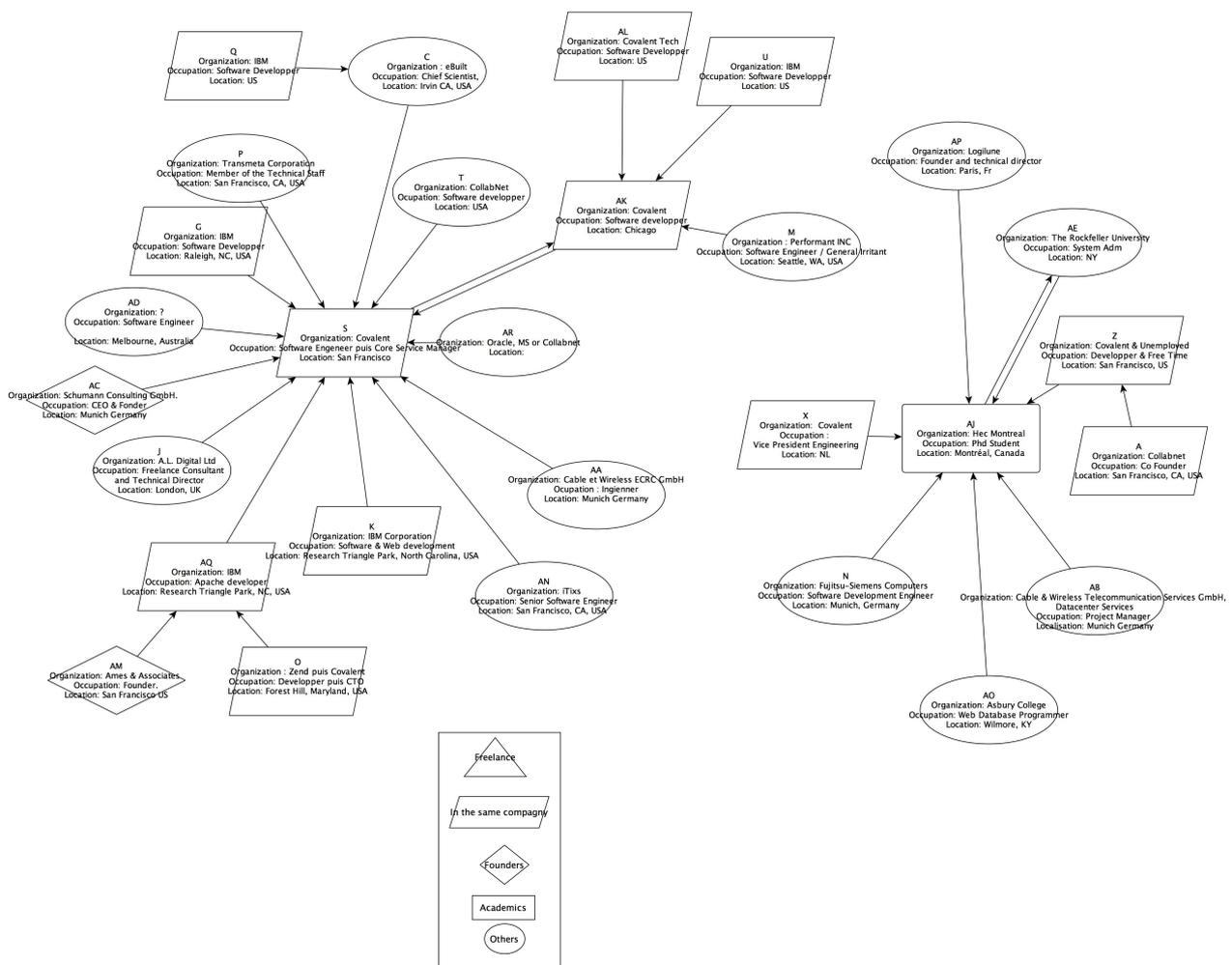
En 2000, il est recruté par CollabNet, l'entreprise fondée par A., l'initiateur du projet Libre-Serveur . Cependant, même si ces deux partenaires ont une forte proximité sociale (ils travaillent pour la même entreprise, vivent dans la même ville), ils ne semblent pourtant pas travailler en duo sur le projet. L'analyse de leur coopération à partir du graphe révèle en effet qu'en 2001 T. travaille dans la "clique"<sup>5</sup> qui s'est formée autour du code, tandis que A. travaille à la documentation du projet.

---

<sup>5</sup> Terme statistique indiquant l'ensemble des sommets d'un sous-graphe, pour une définition plus précise voir p 76 Degenne A. et Forsé. *Les réseaux sociaux*. 2004.

## Encadré: Division du travail au sein des réseaux de développeurs.

L'analyse des réseaux, à partir de l'observation des liens entre développeurs, nous permet de différencier des cliques de développeurs autour d'activités techniques spécifiques. Sur le graphe ci-dessous, on identifie une clique autour de la documentation à droite, et la partie purement dédiée au code à gauche.



## Réseau social de Libre-Serveur en 2001.

A partir de 2002, ces deux membres sont absents des "commits". Cela coïncide en effet avec la montée en puissance d'un autre projet, soutenu par l'entreprise Collabnet. On peut supposer qu'ils ont disparu de Libre-Serveur au profit du nouveau projet. Par la

suite, T. sera recruté comme développeur chez Google.

S. a lui aussi été recruté dès sa sortie de l'université en 1998, ayant validé un Bachelor à Worcester Polytechnic Institute et a également rejoint le site du IBM research center à Raleigh. Son premier post sur les listes de diffusion a lieu en janvier 1999 et lui non plus n'a pas de liens avec la communauté avant son entrée chez IBM. En janvier 2000, il sera recruté par Covalent, une entreprise créée par B., un des fondateurs du projet. Le modèle économique de cette entreprise est directement consacré au support de Libre-Serveur. Dès 2001, nous voyons qu'il se trouve au centre du projet. Au sein de l'entreprise, quatre mois lui suffisent pour passer de la position de Senior Software Engineer à celle de Core Services Manager. Il est alors en charge d'une équipe de cinq développeurs, dont l'activité est totalement dédiée à l'*open source*. Sa notoriété de développeur Libre-Serveur lui a aussi permis de devenir un "gourou", c'est-à-dire une personne qui fait autorité, qui est une pointure internationale et qui est invité à des conférences pour donner des "keynote speeches ».

On remarque en effet que la société Covalent est présente à partir de 2000, mais c'est en 2001 qu'on la retrouve au centre du projet, avec six contributeurs sur les 36 présents, et avec une forte centralité autour de deux commiteurs S. et AK. . Parmi les cinq autres personnes recrutées par l'entreprise, quatre étaient déjà présentes dans la communauté, soit en tant que *commiter*, soit via des *posts* sur les *mailing list* et la plupart d'entre eux travaillaient en *free-lance*. Par la suite, S restera très présent parmi les contributeurs, et il sera recruté en 2004 par la firme Red Hat, spécialisée dans le logiciel libre, et notamment dans la distribution Fedora.

On observe qu'après avoir recruté S., Red Hat s'intègre de plus en plus au projet, puisque l'entreprise a embauché trois autres personnes pour travailler sur ce projet, dont E., l'un des initiateurs de Libre-Serveur.

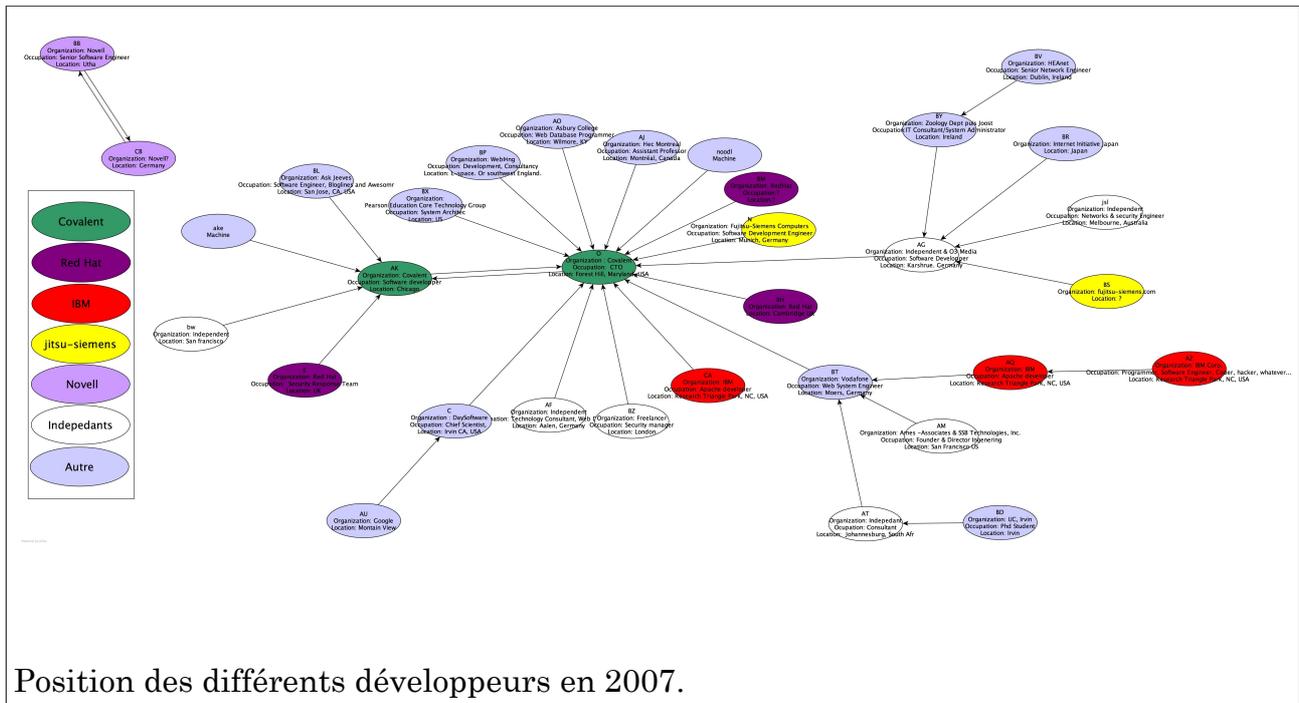
En 2007, on peut voir que cette tendance au sein du projet s'est stabilisée et cristallisée. Covalent est toujours au centre du projet, avec un duo autour de O et AK, O était présent dans le projet dès 1997 alors qu'il était développeur pour la NASA, et AK qui participe à la communauté depuis 2000 est employé de Covalent depuis 2001. Les développeurs de la firme Red Hat, eux aussi sont présents mais à des positions

moins centrales. Aussi si l'on regarde les positions respectives des développeurs AZ et AQ, on peut voir que le désengagement d'IBM se confirme,

L'implication des entreprises dans le logiciel libre semble donc davantage être résultat de trajectoires personnelles que de stratégies délibérées de l'entreprise.

Ainsi, si l'on prend le cas d'IBM, la firme s'est engagée malgré elle dans le projet, poussée par l'un de ses ingénieurs. C'est par la suite, flairant le potentiel du projet, que l'entreprise a décidé de dédier une partie du temps de travail de ses ingénieurs à ce projet. Cette observation contredit la thèse selon laquelle les communautés constituent un réservoir de main d'oeuvre déjà implantée. Cela contredit, dans une certaine mesure, la thèse de Lerner et Tirole (2000) selon laquelle la participation aux projets *open source* servirait de signalement de compétences en vue d'un recrutement. Au regard de nos deux cas, ces personnes ont été recrutées par la firme avant de participer au projet, et il semblerait que ce recrutement ait été réalisé uniquement sur la base de diplômes universitaires.

D'autre part, il est intéressant de noter que le projet a essaimé, produisant un certain nombre d'entreprises nouvelles directement liées au projet. Nous avons mis l'accent sur deux d'entre elles mais, nous pouvons en identifier d'autres de moindre importance ainsi que de nombreux travailleurs indépendants. Les deux entreprises que nous avons citées ont, quant à elles, effectué la plupart de leur recrutement au sein de la communauté, débauchant même des développeurs qui travaillaient auparavant pour IBM. Aussi, peut-on voir que ce sont ces entreprises qui s'installent le plus durablement au sein de la communauté. Le fait que ces deux sociétés aient été créées par des fondateurs de la communauté a sans doute été un élément décisif pour inciter ces personnes à s'engager auprès d'eux.



## De la transparence à la confiance.

Les liens dans ces communautés étant très lâches et ne reposant que sur la participation libre à un projet (c'est-à-dire la liberté de participer mais aussi la liberté de quitter), la confiance reste un élément essentiel au maintien dans la communauté. Dans ce cas précis, il s'agit d'une confiance plus tacite qu'organisationnelle. Il n'y a pas de règles explicites et de contrats formels mais un ensemble de conventions et de règles implicites qui visent à maintenir les relations entre les développeurs, dans le but de mener à bien un projet commun. Ces règles concernent la réciprocité, la disponibilité et différents rites de passage, incorporés au processus d'apprentissage<sup>6</sup>. Dans la théorie des contrats, la confiance a pour rôle de combler les manquements résultant de l'incomplétude de ces contrats. Cette dernière est due à la singularité des actifs mis en jeu. Pour reprendre Lucien Karpik (2007), "les produits singuliers sont des entités incommensurables: ils sont caractérisés par des constellations de qualités et de dimensions, dont les significations sont inscrites dans leurs relations mutuelles"<sup>7</sup>. La plupart des produits singuliers (les grands vins, les conseils d'un avocat ou la psychanalyse) comportent une grande incertitude quant à leur signification, leur qualité et leur dimension. Il en est de même lorsque l'on fait appel à

<sup>6</sup> Pour plus d'indications voir Vicente (2005)

<sup>7</sup> p 39 Lucien Karpik 2007

une société de service en informatique. Cette entreprise peut promettre à un client de résoudre certains problèmes techniques particuliers sans que ce client soit a priori certain de l'identité et des compétences du développeur qui le résoudra et donc, par la même, si celui-ci sera résolu ou pas. Par exemple, dans le monde des sociétés de service en informatique françaises, deux types de contrats sont à l'œuvre : le forfait qui repose sur le résultat (il y a ici indétermination sur le temps que prendra la résolution du problème) et le travail en régie, qui repose sur le temps de présence dans l'entreprise ( dans ce cas se pose le problème de la détermination du tarif de la transaction). Dans les deux configurations, il y a incertitude et risque d'opportunisme des agents. La variable centrale de ces indéterminations est celle de la compétence des acteurs, ici des développeurs.

Selon Karpik, le seul moyen de lever ces indéterminations est de mettre en place des dispositifs de jugement et de confiance. Nous faisons l'hypothèse que le logiciel libre, notamment grâce à la transparence qu'il offre, fait office de dispositif à la fois de jugement et de confiance et ceci de manière plus ou moins inductive. Dans ce dispositif de confiance, la transparence est ainsi aidée par le respect mutuel des normes éthiques autour du logiciel.

Ce dispositif repose sur un modèle de confiance réputation. Dans la lignée des travaux initiés par Kreps (1990), le modèle de la confiance réputation repose sur trois propositions : le passé est le garant du futur (ceux qui ont été loyaux hier le seront demain), la réputation mesure fidèlement la loyauté et les jeux répétés le sont réellement.

Une étude qualitative précédente (Vicente, 2005) nous a confirmé qu'au sein des communautés de logiciels libre, la confiance se construisait via la réputation acquise lors du processus d'intégration propre aux projets. Dans le cas qui nous concerne, l'ancienneté et le maintien d'un fort niveau de participation de ces deux créateurs leur a donc permis de transférer la confiance qu'ils avaient en tant que co-développeurs en confiance employeur- employé. On se place ici dans le cas de la relation d'agence (principal / agent) telle que développée dans la théorie de la confiance autour du contrat de travail. Ainsi, Akerlof et Yellen (1986), remettant en cause la vision walrasienne du marché du travail, indiquent que le rapport salarial et le niveau des salaires ne dépendent pas seulement de la confrontation entre l'offre et la demande, mais que ceux-ci reposent sur un contrat implicite, les employés fournissant plus

d'efforts et de loyauté que ce dont il était convenu dans leur contrat, en échange d'une rémunération supérieure à celle du marché. Ce contrat implicite repose ainsi essentiellement sur la confiance entre les deux contractants et sur une réduction des incertitudes. En effet, conformément à la théorie des coûts de transaction, ces contrats sont toujours sources d'incertitude. D'un côté, l'employeur ne peut pas être assuré de la loyauté et des compétences de son employé, et de l'autre, l'employé ne connaît pas quelles seront ses futures conditions de travail.

Au sein du projet Libre-Serveur, c'est parce qu'ils avaient acquis l'habitude de travailler ensemble que le passage au statut d'employé/salarié a été simplifié. On passe ainsi d'une coopération informelle à un type de coopération qui a pris une forme contractuelle, celle du contrat de travail.

Dans le cas des recrutements intra-communautaires, les incertitudes sont considérablement réduites. D'un côté, l'employeur connaît très précisément les compétences, l'effort et les capacités d'organisation de son futur employé, la transparence permise par l'*open source* offrant la possibilité de pouvoir observer avec exactitude le travail effectué par ces personnes (c'est d'ailleurs grâce à cette transparence que nous avons pu extraire ces données). D'un autre côté, l'employé développeur de logiciel libre connaît très précisément le mode de fonctionnement de l'entreprise pour avoir coopéré avec elle de manière non-contractuelle au cours de son engagement dans le libre.

Un travail qualitatif que nous avons mené reposant sur des entretiens avec des développeurs de deux communautés a en effet démontré que la résistance à la hiérarchie classique de l'entreprise et la volonté de travailler selon un mode alternatif tel que celui du logiciel libre était décisif dans l'engagement des développeurs dans une activité bénévole autour du logiciel libre (Auray et Vicente 2006).

### Une agrégation de firmes autour d'un projet : les consortiums informels.

On voit qu'au fur et à mesure que le projet prend de l'importance, des entreprises de plus en plus diverses prennent part au projet. C'est un élément qui est aussi présent dans l'étude effectuée autour du projet Debian. On peut noter que, même si la part des entreprises reste constante dans le projet, le nombre d'entreprises différentes a

quant à lui explosé. Ainsi, entre 2002 et 2005, la place de la participation des entreprises est passé de 6,49% à 5,93%, alors que le nombre d'entreprises participant à quant à lui doublé, passant de 782 à 1455.

Implication des entreprises au sein du projet Debian.

Nom de la version	Date de sortie	Nombre de paquets logiciels	Part de lignes de code produites par des entreprises	Nombre d'entreprises différentes
Hamm	Juillet 1998	1 096	6, 75%	249
Slink	Mars 1999	1 551	6, 85%	312
Patato	Août 2000	2 611	6, 41%	482
Woody	Juillet 2002	4 579	6, 49%	782
Sarge	Juin 2005	8 560	5, 93%	1455

Sources: Robbles et Al ( 2007)

L'étude qualitative que nous menons montre aussi très nettement cette tendance que l'on peut notamment expliquer par la proximité entre le logiciel libre et la liberté d'entreprise (aspect libertarien) et dans le refus d'une certaine autorité que nous avons évoqué (aspect libertaire). Ce passage peut aussi être contraint, dans la mesure où le développeur, souhaitant se professionnaliser dans les domaines uniquement dédiés au logiciel libre mais ne trouvant pas d'employeur proposant cette activité, est plus facilement enclin à réaliser ce souhait en fondant sa propre entreprise et en restant travailleur indépendant. Pour le travailleur indépendant, le réseau constitué par les communautés de logiciels libres forme un formidable vivier de collaborateurs.

Ce tableau résume, au sein du projet qui nous concerne, la même évolution.

Année	Nombre de développeurs	Nombre d'entreprises différentes (hors indépendants)	Nombre d'universitaires et étudiants	Nombre d'indépendants
1996	10	5	3	1
1997	14	10	3	1
1998	12	7	3	1
1999	23	15	1	2
2001	29	16	2	2
2002	36	13	6	10
2003	35	16	6	8
2004	40	19	5	7
2005	33	16	3	8
2006	34	17	5	8
2007	31	16	1	7

Répartition de l'appartenance des développeurs dans le projet Libre-Serveur

D'un point de vue géographique, on peut s'apercevoir que les travailleurs indépendants se situent principalement en Europe, où le marché d'entreprise dédié au logiciel libre est le moins bien implanté, en comparaison avec certaines zones d'Amérique du nord.

On peut observer deux choses: d'une part le nombre de développeurs reste stable dans le temps à partir de 2001, et, d'autre part, le nombre d'entreprises et d'indépendants tend lui aussi à se stabiliser, sans que l'on aie affaire aux même acteurs.

On voit donc apparaître un réseau d'entreprises travaillant sur un projet, partagé par tous mais aussi librement utilisable par l'ensemble de la planète.

Cette coopération informelle s'apparente aux regroupements connus en économie industrielle sous le nom de « Consortium », un groupement d'entreprises constitué pour la réalisation d'une opération financière ou économique. Dans le cadre de notre projet, cette agrégation d'entreprises repose elle aussi sur la confiance; en effet, ce que l'on a pu observer dans la relation employeur/ employé peut également se retrouver dans la relation entre collaborateurs.

L'Internet a ainsi vu naître de nombreux consortiums dont le plus connu est le W3C,

le World Wide Web Consortium qui vise à l'interopérabilité des formats sur le Web. En France, un consortium directement lié à l'*open source* a vu le jour sous l'impulsion de France Telecom, Bull et de l'INRIA, il s'agit de ObjectWeb, aujourd'hui OW2.

Un aspect caractéristique du consortium autour du projet Libre-Serveur est l'absence formelle de contrat entre les entreprises (contrats de recherche ou de partage de ressources) . Cette absence de contrat est synonyme d'incertitude, mais cependant, comme nous l'avons vu pour le contrat de travail, cette incertitude est comblée et même dépassée grâce à la transparence offerte par l'ouverture du code. Du point de vue économique, il est certain que ce mode de développement coopératif permet une mutualisation des coûts, et du point de vue de l'innovation une fantastique ouverture des possibles sans avoir à en supporter seul le risque.

### Communauté Libre-Serveur et territoires.

Comme nous l'avons mentionné précédemment, les développeurs du projet Libre-Serveur sont concentrés autour de centres névralgiques du secteur des technologies de l'information, fortement concentrés en Californie. Outre le fait que c'est de cette région qu'est issu le fondateur du projet, San Francisco et la Californie sont aussi les lieux les plus importants de l'essaimage informatique. L'étude de Saxenian (1994) montre comment cette région, et notamment la Silicon Valley, a su profiter de la proximité de campus universitaires tels que Stanford. D'autres études, Ferrary (2002) et Grannovetter (2000), ont montré que le dynamisme de cette région est du à la particularité de son réseau social, et notamment à la proximité de capital risqueurs. Ainsi, l'entreprise fondée par A., l'initiateur du projet, a bénéficié du support, en tant que co-fondateur, de Tim O'Reilly, connu pour être l'un des leaders mondiaux de l'édition informatique et un défenseur actif du logiciel libre. En ce qui concerne la proximité universitaire, on notera qu'en 1996, deux étudiants de prestigieuses universités californiennes ont fait partie des premiers participants. Le développement du logiciel libre dans cette région est aussi lié à des éléments culturels tels que la contre-culture locale. Il n'est donc pas étonnant de rencontrer au sein de cette communauté plusieurs organisateurs du BurningMan festival, sorte de fête libertaire se déroulant dans le désert et où l'argent est proscrit. Parmi les autres concentrations de la communauté aux Etats Unis, on retrouve la ville de Raleigh, en raison de la

présence d'IBM et de l'essaimage d'entreprise générée par sa présence. Ainsi, l'entreprise Red Hat, qui compte plus de 2000 employés et dont certains développeurs sont présents dans le projet Libre-Serveur, a installé son siège dans cette ville de Caroline du Nord.

En Europe, on observe une concentration autour de Munich et de Londres, deux villes regroupant un certain nombre de développeurs. En effet, Munich a, en plus de son industrie automobile et électronique, une très forte activité autour du logiciel. Londres et le Royaume Uni, en raison de sa qualité de place financière, ont aussi développé une activité autour du logiciel. La productivité des communautés libres est alors très liée à l'efficacité des territoires productifs où se situent les développeurs. Cela concerne également certaines régions de l'Australie et du Japon.

### Un développement local ?

En s'attachant de plus près à la coopération entre les différents acteurs, on se rend compte que la coopération se fait quasiment systématiquement de manière inter-régionale et très rarement intra-régionale. À titre d'exemple, en 1999, dans les régions de Munich et Londres, il n'y a eu aucune coopération privilégiée au sein de ces zones. En Californie et en Caroline du nord, il n'y a eu qu'une coopération privilégiée intra-zone. (Voir carte 1999)

Le cas de la Caroline du nord est par ailleurs intéressant dans le sens où la coopération intra-zone est de fait une coopération intra-firme, au vu du regroupement géographique de la firme IBM dans cette région. On peut donc s'apercevoir qu'au sein de la communauté Libre-Serveur, les employés de la firme IBM privilégient la coopération avec la communauté, plutôt que la coopération intra-firme (nous rappelons cependant que puisque les liens mettent en lumière les meilleures collaborations, cela ne signifie pas qu'il n'y a pas de coopération là où il n'y a pas de flèches)

On peut néanmoins noter une exception intéressante. En 2007, deux employés de la firme Novell collaborent étroitement, le premier travaillant dans l'Utha aux Etats Unis, le second en Allemagne. On peut remarquer qu'ils forment une clique exclusive,

c'est à dire qu'ils n'ont pas de liens privilégiés avec les autres membres de la communauté. On peut ainsi penser qu'ils participent au projet en vue de développer une application qui ne concerne ou n'intéresse que leur firme, puisque les autres développeurs n'y participent pas.

### Actifs, compétences et valorisation financière.

Le retour aux compétences signifie t-il la fin d'un certain mode de valorisation financière du logiciel? Le modèle de valorisation financière du logiciel reposait auparavant exclusivement sur des phénomènes de mode, de diffusion et de rente. Tout ce modèle reposé sur une propriété bien définie des actifs. Ce modèle semble ainsi être infléchi par ce que nous démontrons pour le logiciel libre. Dans le cas des biens non-rivaux, l'évaluation financière des entreprises est rendue difficile, car l'entreprise en question n'est pas propriétaire des actifs autour desquelles est basée son activité.

En effet, même si le logiciel en question génère une bonne partie du trafic internet, il n'y a pas de valorisation en soi, hormis la fondation qui lui est associée et qui reçoit quelques dons. Les entreprises *open sources* que l'on voit apparaître ont aussi leur valorisation propre. Mais ce qui est remarquable, c'est que parallèlement nous voyons que les développeurs parviennent à valoriser financièrement leur engagement via les différents recrutements dont ils font l'objet. Les parcours professionnels des développeurs nous révèlent qu'une partie de la valorisation économique est transférée au développeur lui-même, qui devient incontournable, et qui profite de l'effet réputation au sein du système *Open Source/Entreprises Associées*. On peut ainsi tenter de faire une relecture du logiciel libre comme une appropriation des actifs. Les développeurs qui travaillent chez un éditeur ou une SSII classique transfèrent l'intégralité de leur travail à l'entreprise. En effet, même si le logiciel est régi par le copyright ou droit d'auteur, une exception a été introduite dans les droits nationaux : dans le cadre d'un contrat de travail, le logiciel réalisé appartient à l'employeur. Dans le cas français, on se référera à l'article L 113.9 du code de la propriété intellectuelle « *les droits patrimoniaux sur les logiciels et leur documentation créés par un ou plusieurs employés dans l'exercice de leurs fonctions ou d'après les instructions de leur employeur sont dévolus à l'employeur qui est le seul habilité à les exercer* ». Aux Etats Unis, cette notion a été introduite par le Copyright Act de 1976 et ses articles

concernant les Creative Employees. C'est donc par la suite l'entreprise seule qui valorise le code et le logiciel. Dans le cas du logiciel libre, les développeurs, en rendant libre leur code, s'affranchissent du transfert direct de l'actif à l'employeur, et, en le redistribuant à tous gratuitement, ils s'en assurent la paternité exclusive, puisque leur nom reste à jamais inscrit dans le code. Ainsi, ce n'est plus le logiciel en lui-même qui est valorisé sur le marché, mais la multitude de compétences, de savoirs et le réseau de personnes qui lui est associé. Ce sont donc les développeurs qui sont au centre de cette valorisation. Cette hypothèse corrobore fortement les niveaux de rémunération de ces derniers. Une étude menée par le Merit en 2006 sur le niveau de rémunération des développeurs de logiciels libres, montre effectivement que les développeurs de logiciel libres travaillant dans de grands groupes de conseil en logiciel ont en moyenne une rémunération douze fois supérieure à celle de leurs collègues non développeurs *open source*. Si l'on prend l'intégralité du secteur, les développeurs de logiciels libre ont des revenus trois fois supérieurs à leurs collègues. Ces écarts de rémunération ne sont pas révélateurs de disparités de revenus liées à la qualité ou à l'intensité de leur travail, mais à un véritable transfert d'actifs et de revenus associés au sein de l'entreprise.

Ratio des salaires par employés : Les firmes Open Source par rapport à la moyenne de l'industrie.

Size Class: Number of employees:	Small <51	Medium 51-250	Large >250	All
Computing equipment	0%	25%	163%	182%
Software consultancy and supply	-9%	29%	1105%	427%
Services Excluding software consultancy and supply	40%	39%	526%	211%
Manufacturing Excluding computer equipment	18%	44%	143%	136%
Other	82%	-20%	202%	204%
All	20%	206%	377%	221%

Difference in mean rev/empl for FLOSS contributor to industry as a % of industry. Sources Merit 2006

Il semble donc que les développeurs de logiciels libres parviennent à capitaliser autour de leurs compétences et de leur positionnement au sein des réseaux. Cette capitalisation est ensuite transformée en rémunération conséquente que l'on voit

apparaître dans le tableau. On peut alors mettre en parallèle notre étude avec celle d'Olivier Godechot (2007) qui dans son travail sur les opérateurs de marché dans l'industrie financière démontre que les *traders* ont opéré un *hold up* sur les actifs des banques par l'attribution de bonus démesurés. Chez les développeurs de logiciels libres nous pouvons voir qu'à travers leurs parcours professionnels, au sein et à l'extérieur des communautés, les développeurs parviennent à attacher la valorisation économique autour de leur activité propre et non plus autour de celle de l'entreprise qui les employait. Les firmes qui par leur participation au logiciel libre misent sur la captation d'externalités positives (logiciel gratuit, support offert par la communauté, ect..) doivent aussi accepter le corollaire, c'est à dire l'ouverture et la non-maîtrise de ces actifs et de leurs employés en charge de leurs développements.

## Conclusion:

Les développeurs de logiciels libres, en faisant le choix de l'*open source*, du partage et de la transparence, sont parvenus à capitaliser une réputation autour du code et des réseaux sociaux de développeurs qui lui sont associés. Ils sont capables de valoriser ces ressources à travers leurs parcours professionnels, la formation d'entreprises spécifiques et le recrutement de collègues au sein des communautés. Ce mécanisme représente une véritable appropriation des actifs auparavant détenus par les entreprises. Il ne s'agit pas néanmoins d'une appropriation sauvage et égoïste telle que celle décrite par Godechot pour le cas des *traders*. Il s'agit plutôt d'une appropriation qui, de manière paradoxale, passe par la mise en commun des actifs, du code et de l'expertise. Avec le succès de plus en plus important de l'*open source*, une nouvelle tendance semble apparaître, celle du déplacement de la tangibilité de l'actif depuis les entreprises vers le commun, par l'intermédiaire de ces développeurs.

Nous ne partagerons cependant pas les conclusions de Lerner et Tirole (2002) selon lesquelles ce sont ces perspectives financières qui sont à l'origine de la motivation des développeurs à participer. Nous considérons en effet que les carrières se construisent de manière « invisible » ( Kaplan 1988) en parallèle à leur participation au logiciel libre. Et qu'on ne peut pas comprendre les motivations initiales des développeurs en extrapolant celle ci de l'analyse de leur parcours professionnels. Ces différents parcours nous apprennent sur la transformation du secteur, et l'impulsion que les développeurs peuvent donner à celui ci. Nous ne pouvons déduire ces motivations qu'à la lumière d'une analyse qualitative et longitudinale de l'engagement. A travers cet article, nous avons cependant vu que la formation de carrière dans le logiciel libre dépendait d'autres facteurs que les simples relations virtuelles. Ainsi, les aspects géographiques et culturels sont tout aussi déterminants. Le travail qualitatif que nous effectuons par ailleurs tentera de mettre en exergue ces éléments.

## Bibliographie:

- Akerlof, George, and Janet Yellen. 1986. *Efficiency Wage Models of the Labor Market*. Orlando, Fla.: Academic Press.
- Anteby Michel 2003 « La « perruque » en usine : approche d'une pratique marginale, illégale et fuyante » *Sociologie du travail* 45 (2003) 453–471.
- Cappelli Peter 2001 « Why Is It So Hard to Find Information Technology Workers? » *Organizational Dynamics*, Vol. 30, No. 2, pp. 87–99, 2001
- Ferrary Michel 2001 « Pour une théorie de l'échange dans les réseaux sociaux. Un essai sur le don dans les réseaux industriels de la Silicon Valley », *Cahiers internationaux de sociologie* 2001/2 - n ° 111
- Godechot Olivier, 2007, *Working rich. Salaires, bonus et appropriation du profit dans l'industrie financière*, La découverte, « Textes à l'appui »
- Granovetter Mark 2000. "Social Networks in Silicon Valley". With Emilio Castilla, Hoky Hwang and Ellen Granovetter. Pp. 218-247 in Chong-Moon Lee, William F. Miller, Marguerite Gong Hancock, and Henry S. Rowen, editors, *The Silicon Valley Edge*. Stanford: Stanford University Press.
- Hutchins Edwin 1995 *Cognition in the Wild*. MIT Press.
- Kaplan Daniels Arlene 1988 « *Invisible careers* » *Women Civic Leaders from the Volunteer World*. University of Chicago Press
- Karpik Lucien .2007, *L'économie des singularités*. Gallimard.
- Kerps Daniel "Corporate Culture and Economic Theory" in J E Alt et K.A Shepsle (éd) *Perspectives on Possitivs Political Economy*, Cambridge, Cambridge University Press, 1990. pp 90-143.
- Lerner Josh et Tirole Jean. « Some Simple Economics of Open Source », *The Journal of Industrial Economics*, vol. 50, n°2, juin 2002,
- Merit 2006. *Study on the: Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU* Final report Prepared on November 20, 2006 Lead contractor: UNU-MERIT, the Netherlands
- Moreland, R.L. Argote, R. Krishnan. 1996. Socially-shared cognition at work: Transactive memory and group performance. J. L. Nye, A. M. Brower, eds. *What's Social About Social Cognition? Research on Socially Shared Cognition in Small Groups*. Sage, Thousand Oaks, CA, 57-84.
- Nonaka, Ikujiro 1995 *The Knowledge-Creating Company* N.Y., Oxford.

- Robles Gregorio , Santiago Duenas, Jesus M. Gonzalez-Barahona 2007 « Corporate Involvement of Libre Software: Study of Presence in Debian Code over Time » in progress.
- Samer Faraj; Lee Sproull 2000« Coordinating Expertise in Software Development Teams » *Management Science*, Vol. 46, No. 12. (Dec., 2000), pp. 1554-1568.
- Saxenian A., 1994. – *Regional advantages : culture and competition in Silicon Valley and Route 128*, Cambridge, Harvard University Press.
- Vicente Michaël. 2005. « Motivation des développeurs de logiciels libres, le cas des projets Debian et OpenBSD » Mémoire de Dea. Université de Technologie de Compiègne.
- Vicente, Michaël . ; Auray, Nicolas. 2006: "Engagement dans le logiciel libre comme réaction au moment intégré", communication at the conference "Le logiciel libre en tant que modèle d'innovation sociotechnique: pratiques de développement et de coopération dans les communautés", Montréal. Canada. May, 16 2006.
- Vicente Michaël .2007 " Crowding out Effect and FLOSS Durability" workshop on "The Diffusion of FLOSS and the Organisation of the Software Industry: From Social Networks to Economics and Legal Models" UNSA Nice .May 31, 2007.
- Wegner D, Toni Giuliano, and Paula T:I Hertel 1987. « Cognitive Interdependence in Close Relationships » in W J Ickés ( Ed), *Compatible and incompatible relationships*. pp 253-276. New York. Spinger Verlag