

***La nouvelle économie du logiciel : compte rendu du workshop initial  
des 6 et 7 décembre 2000.***

Organisé par Alcôve, l'ENS de Cachan et l'ENST Bretagne en collaboration avec l'INRIA.

**Jeudi 6 décembre.**

L'accueil des participants a été réalisé par Laurent Kott (INRIA), qui en a rappelé l'objet : réfléchir à un ou des modèle(s) économique(s) pour le logiciel libre. Il a également souligné le caractère novateur d'un projet où participent notamment chercheurs et professionnels.

Jean-Michel Dalle (ENS Cachan) a présenté le projet, les trois institutions qui le soutiennent (ENS Cachan, ENST Bretagne, Alcôve). L'enjeu est de créer une communauté, réunissant académiciens et praticiens, venus de disciplines différentes (économie, droit, sociologie) capable de réfléchir à froid sur cette nouvelle économie.

Godefroy Dan Nguyen (ENST Bretagne) a souligné que l'émergence du logiciel libre représente un défi majeur pour les économistes, puisqu'il présente des formes hybrides entre marché et non-marché, alors que le cadre de pensée de la discipline reste structuré par l'économie de marché.

***1 – Vers une nouvelle économie du logiciel ?***

Présentation de **Nicolas Jullien** : “ *Le Logiciel libre : création d'un nouveau modèle économique ?* ”

Le logiciel libre, produit par tous et utilisé par tous, ne s'inscrit pas dans le schéma producteur-vente-utilisateur. Son émergence pose deux grandes questions.

**1 – Le modèle économique est-il soutenable ?**

L'organisation de la production du LL est celle d'un bazar (Raymond) néanmoins très organisé, d'un système hiérarchisé (il existe des responsables de projet) fondé sur la réputation. Existe ainsi une communauté de contributeurs (users as innovators, Von Hippel). La création de firmes dédiées et les opportunités de carrière accessibles aux développeurs renforcent les incitations. L'engagement est rendu crédible par l'existence de licences telles que la licence GPL, qui assure que la création restera libre ainsi que la publicité de son auteur.

L'apparition d'entreprises dans cet univers hors marchand pose plusieurs questions : comment peuvent-elles se faire accepter ? quel sens peut avoir le fait de “ vendre du libre ” ? comment faire face alors à une demande nécessairement plus hétérogène ?

Surtout, comment doit alors être géré le risque du passager clandestin, une entreprise vendant du libre faisant du profit avec le travail des autres, et de nombreux projets libres étant financés sur fonds publics ?

**2 - Quelles sont les conditions à réunir pour passer à une modèle économique soutenable ?**

Ce modèle est-il optimal ? Quel est la place de ce modèle vis à vis des autres modèles économiques de production de logiciel (coexistence ou substitution) ?

Quel doit être le rôle des pouvoirs publics : doivent-ils financer des projets libres ? Doivent-ils organiser la coexistence entre logiciel libre et propriétaire ?

**Discussion**

Le logiciel devrait-il être fait en entreprise, alors que d'autres savoirs (par exemple les mathématiques) ne le sont pas ? Si à l'avenir les logiciels s'organisent autour de composants de base, ceux-ci devraient être publics.

L'entreprise n'est incitée à produire que si elle peut mettre des droits de propriété. Cela relève d'un choix de société. IBM fait néanmoins de la recherche dans des domaines où il n'y a pas de droits (par exemple les mathématiques), car la présence des compétences dans l'entreprise est en soi importante.

Certains types de logiciels ne peuvent naître qu'en entreprise, en particulier lorsqu'ils doivent répondre à un cahier des charges très précis et engagent la responsabilité de son créateur (par exemple un système de contrôle aérien). On peut douter de la capacité de la communauté libre à formaliser les contraintes et assumer les responsabilités.

Présentation de **Laurent Kott** : “ *un modèle économique du logiciel ou des modèles économiques du logiciel ?* ”

Le logiciel est un objet dont la définition ne va pas de soi. Cette présentation s'appuie sur les expériences de l'INRIA et les points de vue de ses chercheurs sur le logiciel.

### *De quoi parle-t-on ?*

Les logiciels produits à l'INRIA sont gros ; ils sont l'aboutissement du travail de plusieurs personnes sur plusieurs années, temps de maturation en grande partie incompressible. Ils sont des résultats de recherche dont on ne découvre souvent qu'après coup le potentiel commercial ; en outre, en tant que résultats, ils sont assimilables à une publication de recherche, à une concrétisation de résultats. Mais les logiciels sont aussi des outils quotidiens, qui peuvent être développés en interne. Tout ceci montre qu'on est éloigné ici du schéma de l'entreprise : problème – cahier des charges – solutions.

Il existe en outre diverses catégories (niveaux) de logiciels : logiciels de base (langages de programmation, éditeurs), “ *middleware* ”, boîtes à outils et bibliothèques, applications “ fermées ” (les programmes au sens classique du terme). Les logiciels produits par l'INRIA et diffusés librement (environ 60) entrent dans ces diverses catégories.

### *Considérations économiques*

Des sociétés ont été créées sur la base de licences avec exclusivité temporaire et option d'achat. Pour le libre, en revanche, on n'observe pas pour l'instant de réel impact en termes de création d'entreprises, juste quelques frémissements (par exemple une société produisant des applications métiers fondées sur un LL).

### *Questions sur le libre*

- qui organise le développement : l'initiative de départ d'un chercheur isolé doit nécessairement être relayée ;
- qui assure le service après-vente ? les groupes sont heureux d'accueillir les contributions, à condition d'en contrôler l'inclusion ;
- qui fait le marketing ? (que serait Linux sans Windows ?)

De manière générale, la question est de savoir qui prend en charge les coûts initiaux de diffusion.

### *Conclusions*

La position “ je publie, l'utilise qui veut, entreprise ou particuliers ”, est inefficace. Il faut donc définir le modèle économique avant le régime de propriété, la stratégie de diffusion avant le mode de diffusion ; de même que, dans le “ libre ”, c'est plus l'open-code que la gratuité qui est intéressante.

### **Discussion**

Contre-exemples : l'HTML est un logiciel qui s'est imposé sans marketing ; de même que Mosaic précède IE et Netscape.

Le problème essentiel est de passer des *early-users* aux “ tièdes ”, dont la conviction a un coût.

Présentation de **Mélanie Clément Fontaine** : *Licences libres et protection juridique du logiciel.*

Les licences de logiciel libre sont nées aux Etats -Unis dans les années 80s ; elles autorisent la copie, diffusion et l'accès au code -source du logiciel. La question s'est rapidement posée de leur validité juridique.

Un logiciel est une œuvre de l'esprit, et se trouve en tant que telle sous le régime du droit d'auteur. Ce droit a d'ailleurs été aménagé en 1985 pour mieux s'appliquer au logiciel. Une licence autorise à utiliser le logiciel sans entraver le droit. Elle peut être libre ou propriétaire ; la différence réside dans les conditions de l'utilisation (droit de copier / copie pour la seule sauvegarde ; invitation à la diffusion / utilisation sur un poste déterminé ; droit de modification / seules quelques exceptions légales). Il existe une diversité de licences libres. Elles ne présentent pas toutes les clauses contractuelles habituelles, certaines n'ont ainsi ni définition ni préambule (exposé des motivations des parties, capital pour le juge en cas de litige).

La question est la suivante : le droit offre-t-il à un auteur la liberté d'offrir à l'utilisateur ces libertés ? Les droits d'auteur renvoient à deux droits :

- les droits moraux : les licences LL ne contreviennent ni au droit de paternité, ni au droit de divulguer ;
- les droits patrimoniaux : si la démarche négative de renonciation à l'exercice du droit de propriété, acte unilatéral, n'a aucune valeur juridique, la démarche positive d'autorisation d'utilisation, qui est par exemple celle de la licence GPL, a une validité contractuelle.

Il est donc possible au créateur d'utiliser le droit des obligations pour libérer l'usage de son droit.

Les questions qui restent à résoudre :

- quand il y a coordination des tâches, quelle est la qualification juridique du produit ?
- quelle est la licence " libre " la plus efficace ?
- quelle est la place de la réservation dans la propriété intellectuelle ?

#### **Discussion :**

De fait le logiciel se brevète ; les brevets logiciels représentent un îlot de réservation dans l'archipel du LL. C'est une façon détournée d'établir une réservation. On peut considérer que c'est le fait d'avoir choisi le droit d'auteur pour le logiciel (ce choix remonte à la fin des années 1960, et correspond à des motivations de politique économique internationale) qui conduit à cacher le code source, et que l'open -source est une réaction à ce mouvement. Pourquoi alors (suggestion de Laurent .Kott) ne pas échanger le brevet contre l'accès aux sources ? Le brevet n'entraînerait cependant pas nécessairement l'open -source : du fait de la taille des programmes, la contrefaçon (involontaire) est inévitable, ce qui incite à fermer l'accès au code.

Des comparaisons heuristiques ont été faites avec les mathématiques d'une part (un logiciel est -il plutôt un algorithme ou un produit ?), avec la musique et d'autre part (principe des huit notes : peut-on définir de façon identique un seuil minimal de contrefaçon pour le logiciel ?)

**Gérard Roucairol** (président du comité d'orientation du RNTL, Directeur Scientifique de BULL) a ensuite rappelé rapidement les chiffres du marché du progiciel : 15% de croissance/an en moyenne (prospective), 47% de parts de marché pour les Etats -Unis, une très faible présence française. La nécessité de mise en place de nouveaux *business-models* et de leur diffusion est d'autant plus pressante.

Le problème à résoudre est le suivant : aujourd'hui, les grandes entreprises de logiciels consacrent 19% de leur CA en R&D. Si on fait du service et que le produit est gratuit, comment financer la R&D ?

## **2 - Les logiciels libres entre communautés et opportunités économiques.**

## **Christophe Le Bars**, présentation du modèle économique d'*Alcôve*.

C'est un modèle de service ; la source du produit n'est pas dans le code, mais dans le conseil. La licence libre accorde trois libertés : utilisation, modification et reproduction. La seconde est la plus difficile à exercer, dans la mesure où les entreprises n'en ont pas toujours le temps et les compétences. Le modèle économique est alors celui d'un " mercenaire de la liberté ". Les entreprises ont le plus souvent l'argent mais pas le temps de modifier et expliquer les LL <sup>1</sup>. Alcôve ne fait pas d'argent sur le code, mais accepte les règles du jeu de la communauté libre et les contraintes de la licence GPL : si un *patch* est créé, il est ensuite intégré au LL. Alcôve tente donc d'être expert autour du LL. Ceci implique de faire de la R&D : 20% du temps des ingénieurs – qui sont tous originaires de la communauté du libre – y est consacré, le code produit par cette R&D étant intégré au LL. Par ailleurs, l'entreprise tente de tirer des enseignements de l'organisation de la communauté libre pour sa gestion interne : les ingénieurs sont par exemple candidats sur leurs missions. Alcôve ne vend donc pas du logiciel, mais du temps et de la connaissance (refus de la qualification de SSII).

### **Discussion :**

La discussion a porté pour l'essentiel sur les relations entre l'entreprise et la communauté du libre. Alcôve travaille sur un logiciel existant, utilise le travail des autres, ce qui risque d'avoir pour conséquence de diminuer le bon fonctionnement et la créativité de la communauté. Ce à quoi il a été répondu qu'Alcôve produit aussi du code, pour ses clients et dans le cadre des 20%, code qui est libre. En outre, l'ensemble de ses ingénieurs viennent de la communauté du libre, c'est ce qui fait leur compétence et le fait qu'ils respectent les règles du jeu. Q : Les 20% de R&D se justifient par l'entretien des compétences, et idéologiquement par le respect des règles du jeu. Qu'advierait-il si un concurrent réduisait ce chiffre et devenait ainsi plus rentable ? R : on peut penser qu'il serait sanctionné pour ne pas respecter les règles du jeu, et ne parviendrait plus à attirer les ingénieurs les plus compétents. D'un point de vue financier, il s'agit d'un *business-model* de service : marges faibles, basé sur des ressources humaines donc menacé de hausses de salaires inévitables. Cela pose la question plus générale de la soutenabilité d'une entreprise dont la culture spécifique est une pièce essentielle.

## **Jacques Le Marois**, présentation du modèle économique de *Mandrakesoft*.

*Mandrakesoft* revendique un *business-model* différent de celui d' *Alcôve* : faire du logiciel et se rémunérer autour.

Le produit Linux-Mandrake contient le système d'exploitation, les applications et des liants. L'entreprise récupère l'ensemble des développements Linux, et y contribue. L'ensemble est open source, et peut être téléchargé gratuitement.

La source de revenu de l'entreprise est double : d'une part, le produit est vendu " en boîte ", complété d'une documentation et de logiciels propriétaires ; l'argument du gratuit permet d'en déterminer le prix en fonction de la cible. L'autre part des revenus proviennent de la formation et du conseil réalisé autour de ces produits. L'augmentation rapide du nombre d'utilisateurs est favorable à ces deux sources de revenu ; le téléchargement gratuit permet de s'étendre rapidement, l'implantation d'agences commerciales n'intervenant que dans un second temps.

La question du rapport à la communauté Linux, dont le travail profite à l'entreprise, est gérée par la restitution de tous les développements réalisés par l'entreprise. Ce " *give back* " est essentiel à l'image de marque de l'entreprise. En outre, on ne peut bâtir un revenu sur un logiciel que s'il est suffisamment complexe et critique pour une société.

Le développement très rapide de l'entreprise (3 personnes en 1998, 120 aujourd'hui), peut être interprété comme le résultat du jeu complet de l'open source : le site a ainsi été traduit par des bénévoles dans plus de 11 langues.

Il s'agit donc d'un " projet open-source, avec une société derrière ".

---

<sup>1</sup> Logiciel Libre.

La **Discussion**, assez animée, a portée sur les questions suivantes :

Viabilité du modèle économique : les bénéfices sont faibles à ce jour, ce en quoi J. Le Marois voit le résultat de la nécessité d'investir et de se développer mondialement. Mandrakesoft ne serait-il pas dans un modèle économique traditionnel d'édition (aujourd'hui, 98% des revenus proviennent des boîtes ; à terme, il est prévu que 20% viennent des services) ? Auquel cas, l'enjeu est d'atteindre une taille critique, qui reste à déterminer. Néanmoins, le modèle open-source permet une réactivité beaucoup plus grande.

Quelle est la supériorité de Mandrake sur ses rivaux (notamment RedHat) ? Mandrake s'est voulu *user-friendly* et tourné vers le grand public, quand RedHat refusait de l'être ; quelques choix lui ont en outre permis d'apparaître beaucoup plus innovant que ses concurrents. L'image de marque (du moins tant que la taille critique n'est pas atteinte) s'avère déterminante pour la diffusion (Mandrake est tourné vers le " grand public ", puis importé par ses utilisateurs dans leur entreprise).

Présentation d'**Olivier Dupouet** : *Linux, communauté de pratique ou communauté épistémique ?*

Il s'agit ici de caractériser la communauté Linux à la lueur de la littérature économique portant sur le lien entre apprentissage individuel et apprentissage organisationnel. Une organisation est en effet composée de communautés entremêlées et interagissantes ; la création d'une communauté organisationnelle repose sur l'échange entre ces communautés.

On distingue différents types de communautés : notamment le groupe fonctionnel (hiérarchique), l'équipe (hétérogène tournée vers un but commun), le réseau (informel basé sur la confiance). Les deux qui nous intéressent ici sont la communauté de pratique (les membres cherchent à améliorer leur compétence individuelle et s'appuient sur la communauté pour faire circuler les pratiques ; ils partagent des valeurs et des compétences techniques) et la communauté épistémique (l'accumulation des connaissances est ici un but en soi ; une autorité procédurale est nécessaire, ainsi que la création d'un langage commun).

Parallèlement, on distingue deux types d'apprentissage : le mode 1 respecte la séparation traditionnelle entre création et mise en œuvre de la connaissance, et pose un problème de rapatriement de l'expérience. Le mode 2, qui se développe aujourd'hui, affaiblit cette barrière, ce qui offre aux communautés de pratique la possibilité d'être des moteurs de l'innovation.

Linux est un bon exemple du glissement d'une communauté de pratique vers une communauté épistémique : à l'origine développé par le sous-groupe des hackers, qui est une communauté de pratique (objectif de développement des compétences, cooptation des membres, culture du don et de la réputation), il a été conduit à mettre en place une nouvelle articulation de la communauté permettant de gérer l'innovation constante, notamment une autorité procédurale.

#### **Discussion:**

Au départ, Linux n'était qu'un jouet, n'était pas le projet qu'il est actuellement. Aujourd'hui, il est organisé par projets séparés, structurés, parfois munis de constitutions. Les règles s'élaborent au fur et à mesure de l'augmentation de la taille de la communauté. Linux apparaît ici comme une sorte de dictateur bienveillant. Le fait que les entreprises interviennent change-t-il la donne ? Vont-elles vouloir intervenir dans l'organisation, en échange de la mise en libre de leurs propriétaires ? Dans la mesure où les membres de la communauté sont aussi les salariés des entreprises, elles ont sans doute peu intérêt à le faire.

Présentation de **Christian GENTHON** : *Internationalisation des logiciels et des services informatiques, une introduction.*

La question posée est : quel est le modèle économique du logiciel aujourd'hui, et à dix ans ?

Cette présentation s'appuie sur une base de données créée par l'auteur. Contrairement à l'hypothèse faite habituellement dans les travaux d'économie, il est considéré ici que services et logiciels

informatiques n'appartiennent pas au même secteur, qu'ils ont des régimes de concurrence et connaissent des mouvements d'internationalisation distincts.

Parmi les cinquante premiers groupes du secteur, on trouve environ pour moitié des entreprises de services, et pour moitié des éditeurs. Au cours des années 1990, la part des éditeurs par rapport aux sociétés de service s'accroît.

On observe pour les deux secteurs deux profils bien distincts :

*Des écarts d'internationalisation* : la part des exportations et des investissements directs à l'étranger est très différente selon que l'on considère l'un ou l'autre secteur. Elle se situe autour de 20% dans les services, de 45% pour l'économie du logiciel.

En outre, contrairement à ce que pourrait laisser penser le contexte de mondialisation, ces chiffres sont restés stables au cours des dix dernières années. Cela s'explique sans doute par la vigueur de la croissance américaine sur la période, qui a permis aux entreprises d'accroître fortement leur chiffre d'affaire sans courir le risque de l'internationalisation ; ceci dans un contexte où l'économie du logiciel (édition) est essentiellement américaine.

*Des régimes de concurrence distincts* : la part de la R&D est de l'ordre de 2% dans les services, tandis qu'elle se situe autour de 20% dans l'édition ; corrélativement, la part de SG&A ( *sales, general and administrative expenses* ) se situe autour de 20% pour le premier secteur, 40% pour le second. Ils obéissent donc à des régimes de concurrence distincts ; notamment, des barrières à l'entrée existent pour l'édition logiciel (coûts fixes de production et de ventes) qui n'existent pas dans le secteur des services. D'autres ratios confirment la séparation des deux secteurs.

Le marché s'est donc désintégré verticalement : les fabricants de logiciels -outils, en amont, se distinguent aujourd'hui des fabricants d'applications utilisant ces outils. C'est un modèle qui rend compte de 99% du CA du secteur ; c'est un modèle lourd, qui s'est mis en place sur vingt ans, et qu'il est improbable de voir se transformer en moins de 20 ans.

Comment le logiciel libre peut-il s'insérer dans cette organisation industrielle ? Ce ne peut être que celle d'une niche, étant donné les coûts fixes importants. Certes, une partie du coût de R&D peut être transféré à une communauté de hackers, mais ce transfert ne saurait être complet. La solution trouvée aujourd'hui est double : sponsoring et surenchère boursière. Les stratégies à l'origine du premier et les logiques de la seconde sont également réversibles. Et un modèle dans lequel les services financeraient le développement des outils n'est pas tenable car il suppose une intégration verticale qui a existé au début de l'industrie mais qui n'existe plus.

#### **Discussion :**

- Le modèle de l'édition (*one-to-one*) s'est construit sur environ 50 ans ; celui de l'audience (*one-to-many*) a nécessité 10 ans. Peut-être le modèle de l'Internet et du LL, *many-to-many*, se mettra-t-il en place plus rapidement que ne le laisse supposer l'inertie des structures économiques.
- La question de la réussite de l'édition du LL peut être envisagée sous l'angle d'un problème de standard. C'est ce standard qui est à la source de la position de force de Windows. Le LL (tel que Mandrake) doit-il devenir un nouveau standard pour être viable ? Quel est le seuil ?

Présentation de **François Horn** : *Les mondes de production du logiciel.*

L'économie du logiciel présente deux grandes caractéristiques :

- une très forte hétérogénéité des produits. Coexistent ainsi des logiciels sur mesure, dont la production relève quasiment de l'activité de service, et des logiciels, bien intangibles, qui ont les caractéristiques de biens collectifs purs. A cette hétérogénéité des produits correspond une

diversité des producteurs : quelques grandes entreprises dominent les activités matures, tandis qu'une multitude de petites entreprises se positionnent sur les segments nouveaux.

- l'économie du logiciel est jugée en crise permanente : elle présente des gains de productivité faibles, beaucoup plus faibles que dans le secteur du matériel informatique, par exemple. En outre, les produits continuent de comporter beaucoup d'erreurs, d'offrir une fiabilité et une adaptations aux besoins insuffisantes par rapport aux exigences.

En suivant le schéma théorique mis en place par R.Salais et M.Storper, on peut considérer que l'économie du logiciel est caractérisée par la coexistence de quatre mondes de productions différents, reposant sur des conventions différentes :



	<b>Absence de standardisation</b>	<b>Standardisation du produit et/ou des composants</b>
<b>Produits dédiés</b>	<b>Monde interpersonnel</b>	<b>Monde de la production flexible</b>
	Type de produit : <i>Logiciels sur -mesure développés "ex-nihilo"</i>	<i>Logiciels standards et services sur mesure. Logiciels sur mesure à partir de composants standardisés.</i>
	Exemple caractéristique : <i>Application spécifique</i>	<i>Système de gestion d'entreprise</i>
	Producteurs principaux : <i>Services informatiques internes, S.S.I.I.</i>	<i>S.S.I.I., sociétés de conseil, liées à des producteurs de matériel ou de progiciel</i>
	Utilisateurs principaux : <i>Grandes entreprises, Administrations</i>	<i>Grandes et moyennes entreprises, Administrations</i>
	Productivité dans la production des logiciels : <i>Faible. Contradictoire avec fiabilité</i>	<i>Assez élevée</i>
	Fiabilité des logiciels produits : <i>Contradictoire avec productivité.</i>	<i>Assez élevée</i>
	Adéquation aux besoins des utilisateurs : <i>Variable selon le niveau d'intercompréhension producteurs/utilisateurs</i>	<i>Dépend de la qualité de la relation de service</i>
<b>Produits génériques</b>	<b>Monde de la création</b>	<b>Monde fordiste</b>
	Type de produit : <i>Logiciels libres (code source)</i>	<i>Progiciels commerciaux (code objet et services limités)</i>
	Exemple caractéristique : <i>Le système d'exploitation Linux</i>	<i>Tableur</i>
	Producteurs principaux : <i>Universités, centres de recherche, créateurs indépendants</i>	<i>"Éditeurs" de logiciels</i>
	Utilisateurs principaux : <i>Au départ informaticiens</i>	<i>Ménages, entreprises (y compris P.M.E.), administrations</i>
	Productivité dans la production des logiciels : <i>Variable</i>	<i>Elevée</i>
	Fiabilité des logiciels produits : <i>Amélioration rapide possible (pour les logiciels qui connaissent un succès initial)</i>	<i>Souvent insuffisante</i>
	Adéquation aux besoins des utilisateurs : <i>Forte pour la communauté informatique. Plus problématique pour les simples usagers.</i>	<i>Uniquement pour des besoins standards</i>

La production du Logiciel libre relève dans cette perspective du monde de la création.

Des dynamiques existent entre les mondes de production : l'industrialisation de certaines activités les fait passer du monde interpersonnel au monde fordiste ; l'objectivation des activités du monde interpersonnel les transpose dans le monde de production flexible ; une flexibilisation de la production fordiste transfère ses produits dans le monde de production flexible. En ce qui concerne le LL, sa production peut passer du monde de la création à celui de la production flexible par un processus de valorisation.

Le développement du LL apparaît donc comme une opportunité importante pour l'ensemble du secteur économique. Il apparaît néanmoins menacé pour plusieurs raisons : son succès diminue les

motivations à la production, dans la mesure où on évolue vers des développements moins prestigieux, dont les producteurs ne sont pas les premiers utilisateurs ; la complémentarité des activités laisse place aux comportements de *free-riders* ; le soutien dont il bénéficie de la part des entreprises pourrait disparaître à mesure que les LL deviennent des produits concurrents plutôt que complémentaires.

On peut en déduire la nécessité d'un soutien du LL par les pouvoirs publics.

#### **Discussion:**

Traditionnellement, les secteurs économiques passent du monde de interpersonnel au monde fordiste (industrialisation), puis au monde flexible (flexibilisation). Il est remarquable qu'il existe ici un passage direct du premier au troisième (objectivation).

Il faut établir une distinction entre l'évolution des produits et des entreprises, qui peuvent connaître des dynamiques d'évolution (d'un monde de production à l'autre) différentes.

### **3 - les composants logiciel.**

Présentation de **Jean Aguado**, Thomson Multimedia.

L'activité de Thomson a été marquée, au cours des dernières années, par le passage d'une activité de développement logiciel à une activité d'intégration. La croissance s'est accompagnée d'un recentrage de l'entreprise sur son métier, et la vocation de Thomson aujourd'hui la conduit de plus en plus à utiliser des composants logiciels venus de l'extérieur. Ceci pose plusieurs problèmes :

- une perte de contrôle sur le cycle de vie des composants ;
- un manque de visibilité sur la manière dont ces éléments sont développés ;
- une perte de la capacité à corriger les éléments défectueux.

Le métier d'intégrateur de Thomson est donc de construire des systèmes logiciels complexes, dont il garantit au client une durée de vie de 20 à 30 ans, en intégrant des composants consommables ne présentant aucune garantie, et dont la durée de vie est d'un ou deux ans.

Le fait que le logiciel soit libre ou propriétaire est ici secondaire.

Les composants logiciels peuvent être soit développés en interne, soit achetés sur le marché, soit résulter de l'utilisation de LL.

- composants internes : d'une part, développer coûte cher ; ce ne doit être fait en interne que si c'est indispensable, et réutilisable (problématique court terme/long terme, général/particulier). En outre, cela nécessite la mise en place d'une architecture organisationnelle efficace permettant la circulation et le développement des composants dans l'ensemble du groupe Thomson.
- composants externes propriétaires : le marché tend à évoluer vers un marché de volume (bas prix, faibles marges...) dont les produits sont fortement inadaptés aux besoins de l'entreprise : les attentes d'une entreprise telle que Thomson résident au contraire dans une garantie de qualité, une pérennité des performances, des interfaces, des fonctions, une obligation de correction des erreurs.
- composants LL : l'intérêt des tels composants est que, leurs sources étant disponibles, il est possible de les corriger et de les faire évoluer, outre qu'ils dispensent du coût des licences d'utilisation. En revanche, leur évolution très rapide est problématique ; surtout, les termes des licences et les implications juridiques concrètes des licences libres sont insuffisamment claires.

Thomson se trouve face au LL dans une situation d'insuffisante clarté des règles du jeu : comment s'applique, dans le cas d'une intégration de composants libres dans un produit Thomson, le droit des fournisseurs ? le droit des utilisateurs ? quelles sont les garanties associées au LL ?

En définitive, le LL pose un problème de responsabilité (qui est responsable en cas de *bug* d'un composant libre ?) et de compétences (il fait appel à des compétences extrêmement spécifiques, dont il faut décider si on les possède en interne ou non).

### **Discussion :**

Les besoins de Thomson vont à contre-courant de l'évolution du marché, de plus en plus dirigé par les besoins du grand public. D'où un appel que l'entreprise lance aux gens du libre : mais ceux-ci ne proposent qu'une expertise, tandis que la demande est aussi une demande de responsabilité.

*Alcôve* tente de développer des services "de garantie". Le LL est dans l'ensemble plus apte à répondre à ces besoins, car la situation de concurrence y pousse.

### Présentation d'**Antoine Beugnard**, *Les composants logiciels*.

Les composants logiciels visent trois objectifs : une gestion time-to-time des logiciels, leur évolution, leur maintenabilité.

Cela correspond à une évolution du marché : on passe d'applications monolithiques sur mesure à des applications modulaires et évolutives ; des composants logiciels à des lignes de produit. On observe aussi une tendance à l'évolution vers le service (des temps d'utilisation d'un logiciel sont achetés sur un serveur).

L'enjeu majeur de l'alternative entre applications et composants logiciels réside dans les interfaces.

Le cycle de vie d'un composant logiciel peut être mis en relation avec un certain nombre de métiers, avec différentes étapes de la division du travail de production du logiciel : l'invention, la conception-réalisation, la validation, la gestion des composants (mise en place de bibliothèques), l'intégration ou assemblage de composants, la configuration, l'administration.

La production du logiciel semble donc promise à évoluer vers plus de réutilisation. Cette nouvelle configuration pose par ailleurs deux problèmes de confiance : un problème de confiance sur le produit, un problème de confiance sur le service.

### Présentation de **Nicolas Carayol** : *Brevets logiciels et "anti-commons"*.

Que nous enseigne l'analyse économique sur la question des brevets ?

En analyse statique, on montre que l'octroi d'un droit de monopole incite à l'innovation, au prix d'une petite distorsion du marché. D'un point de vue dynamique en revanche, il apparaît que cet octroi exige en contrepartie la divulgation des connaissances sous-jacentes au brevet.

Ce sont en fait deux problèmes qui se posent dans cette question des *commons* : celui de la complémentarité des usages, celui de la séquentialité dans la création.

*Anti-commons problem n°1* : la complémentarité des usages.

Si les connaissances sont "encloses", des coûts de transaction apparaissent, qui s'accroissent avec les problèmes de confiance. Les coûts liés à l'*enclosure* sont également d'autant plus élevés que sont fortes les combinatoires technologiques. Les vagues de brevets accroissent donc les coûts de transaction ; elles produisent en outre des externalités sur la propension à breveter, et conduisent souvent à l'appropriation de connaissances existantes.

Le LL peut réduire ce problème d'usage, en garantissant la liberté d'utilisation, garantissant le *knowledge diffusion*. Mais cela réactive les problèmes d'incitation à l'innovation ; notamment du point de vue des producteurs de logiciel propriétaire, qui peuvent craindre que le LL soit la voiture-balai des innovations.

La question est donc, si on se place dans un cadre dynamique, de savoir comment fixer les caractéristiques du système de brevets de manière à rétribuer justement les innovateurs successifs. L'avantage du système des brevets dans ce cadre réside dans les externalités de connaissances ; son désavantage, sur le risque de préemption sur la génération suivante.

*Anticommons problem n°2*.

Les brevets sont bloquants, en ce qu'ils incitent à déposer les brevets qui vont le plus gêner les concurrents ; ils encouragent également à la production de connaissances inutiles. Si les firmes anticipent ce processus, l'autosélection et la spécialisation des champs de recherche qui en résultent

accroissent les difficultés d'absorption des connaissances extérieures. En bref, un tel système incite à la prise de brevet en raison de la place qu'il occupe, plutôt que de la valeur qu'il pourrait générer. Il s'agit d'un *problem of problem choice*. Ce terme renvoie aux analyses économiques des communautés scientifiques, représentées par des arbres plus ou moins difformes selon le degré de convergence des recherches menées dans ces communautés. De mauvaises incitations conduisent ainsi à des formes pathologiques, du type " puits de science " ou dispersion radicale. Il faudrait mener une analyse similaire dans le cas des brevets, afin de savoir comment corriger la distorsion. Cela nécessite d'établir une description des caractéristiques optimales du logiciel, afin de pouvoir fixer une norme de brevetabilité, notamment en termes de longueur (durée), de largeur, de niveau exigé de nouveauté, de degré de divulgation des connaissances.

En définitive, le problème est plutôt de savoir quel système de brevet conviendrait au logiciel.

**Discussion :**

- le brevet n'est pas la première arme des entreprises, par rapport à la R&D et au marketing notamment.
- 20 ans de monopole d'utilisation ne représentent pas (à la lueur des expériences de l'INRIA) une durée excessive. Le problème est plutôt la question de la compatibilité entre brevets et droits d'auteur.
- Le problème essentiel du degré de nouveauté apparaît indécidable (le parallèle avec les 8 notes de musique semble inefficace ici).
- Dans un système de brevet, un LL pourrait être en contrefaçon.

## Vendredi 7 décembre

### 4 - *Nouveaux modèles économiques*

Présentation de **Pierre Bruno**, Matra DataVision

Pierre Bruno présente Open Cascade, plate-forme de développement de composants logiciels. Il définit les trois marchés essentiels de l'open source :

- un marché grand public, avec des prestations relativement standard
- un marché multisecteurs de services autour de logiciels open source
- un marché ciblé.

L'avantage essentiel de l'open source est pour P. Bruno sa capacité d'accélérateur de diffusion.

Deux grands types d'activité d'open cascade :

*les services* : la formation, le conseil en ingénierie logicielle. P. Bruno souligne que les industriels sont très demandeurs d'une valorisation du savoir-faire de leur métier par le développement de logiciels.

*les supports* : le logiciel open source est important car il est le support de plusieurs plates-formes.

Les clients sont essentiellement des industriels et des éditeurs de logiciels.

#### **Discussion :**

Cette activité a suscité des débats autour de la nature économique du projet et de celle du logiciel.

*l'open source représente-t-il une stratégie commerciale pure ?*

La réponse faite est mitigée : l'open source représente certes une stratégie commerciale mais aussi une création de valeur et une mutation de la forme du logiciel. Il permet de combler un vide dans l'univers du logiciel, sur un marché particulier.

*le logiciel est-il considéré uniquement comme un produit ?*

P. Bruno insiste sur le fait que ce qui est important est non pas la vente d'un produit mais la mise à disposition de compétences. Le développement réalisé pour un client est spécifique, le code lui appartient ensuite.

*le logiciel est-il d'abord un programme (une recette) ou un produit ?*

Tout dépend en fait du logiciel. Dans des applications très spécialisées, la logique de produit est défaillante.

*retombe-t-on dans un schéma proche de celui de la recherche fondamentale à partir du modèle économique présenté ?*

(Cette question est une réaction à la mention du projet SALOME du RNTL, en partenariat entre des grands industriels, des PME et des universités ; ce projet bénéficie d'un financement par le secrétariat d'Etat à l'industrie.)

La nature du logiciel est double pour P. B. : il est à la fois connaissances et produit. Ceci rend possible le passage d'un modèle du logiciel comme produit à un modèle comme connaissances codifiées qui peut par conséquent bénéficier des financements publics, de même que la recherche fondamentale. Les contributions publiques sur SALOME se justifient par le fait que l'ensemble des entreprises ont intérêt à ce développement : la stratégie du cofinancement (40 % du public) est une stratégie gagnant / gagnant. Les institutions publiques peuvent cofinancer des projets open source, alors que les universités contribuent à leur étude.

Présentation de **Franco GASPERONI**, ACT Europe.

F.G. présente le contexte d'émergence du logiciel GNAT et de son environnement de référence Ada 1995. Il souligne quelques avantages des logiciels libres selon lui : ils permettent une collaboration " *engineer-friendly* " qui accueille beaucoup de jeunes ; en particulier les étudiants et les universités peuvent participer.

Le travail se concentre sur l'innovation et non pas sur le marketing  
Clients et usagers peuvent contribuer à l'amélioration du produit.

GNAT constitue la vente d'un produit avec une licence logiciel libre : l'accès aux sources est libre, la redistribution est permise, le nombre d'installations n'est pas limité.

Pour F.G. le logiciel propriétaire n'est pas adapté aux entreprises qui ont un fort support technologique. Au contraire avec le GNAT, la part du support est très grande. Il en résulte pour les clients une réduction des coûts et des risques liés à l'utilisation : ils n'ont pas besoin de devenir experts de l'environnement. Ceci est un avantage, alors que le panorama technologique devient très complexe.

Discussion :

Les questions ont porté sur des modalités concrètes de fonctionnement du système.

*Quel est le chiffre d'affaires ?*

Entre 3 et 5 millions d'Euros, c'est à dire cinq fois moins que si le système était en propriétaire.

*Comment des concurrents peuvent-ils reprendre des codes ?*

Ils peuvent parfois le faire de manière illégale, mais fermer les yeux n'est pas une mauvaise chose : la création d'un environnement concurrentiel, afin de ne pas être seul sur le marché, est préférable.

*la récupération des codes par d'autres a-t-elle pour contrepartie la récupération de parts de marché ?*

Dans cet environnement, il ne faut pas adopter la logique économique habituelle. L'essentiel est une logique globale de don / contre-don, de retour en termes de confiance... Le logiciel libre donne du " *mind share* " ; il aboutit à une sorte d'écosystème global, d'équilibre.

*4) Le marché est en croissance, cependant peu d'étudiants semblent intéressés par ADA*

C'est une question de circulation d'information et de " pompe à amorcer ".

*et les capital risker ?*

Leur objectif étant essentiellement de mettre en bourse rapidement des entreprises pour générer des profits, la culture est ici différente et implique essentiellement des ingénieurs et des techniciens.

*les caractéristiques d'une licence GPL.*

Une fois le produit acheté, la diffusion est possible. Elle ne se fait cependant pas car les acheteurs de ces produits sont eux-mêmes en concurrence.

Chacun a en fait intérêt à la coopération, dans des segments où la tendance au monopole est forte du fait de l'importance des standards en informatique. Celui qui ne devient pas un standard a des difficultés à survivre.

Présentation de **Jean-Benoît Zimmermann** : *propriété intellectuelle et innovation*.

La nouvelle économie peut-elle être donner lieu à des appropriations privées ? Si la réponse est positive, quel régime de propriété intellectuelle envisager ? Quelle est la pratique de création et d'usage des brevets possible ? *Le problème est de trouver un équilibre entre la protection des résultats et l'usage social de l'invention.*

L'accord entre l'incitation à inventer et la diffusion de la connaissance s'obtient en jouant sur l'étendue et la durée de la protection. Les solutions sont toujours limitées et dépendantes des caractéristiques de l'industrie concernée, en particulier du rythme et de la cumulativité des inventions.

*Dans le cas du logiciel, le cadre de la propriété intellectuelle n'est pas unique*. Les Etats-Unis et le Japon par exemple fondent leur système sur les *droits d'auteur* ; aux Etats-Unis l'importance de la jurisprudence conduit à une dérive vers un système de *brevets*. Ceci suscite de vifs débats. En Europe, la Communauté Européenne se montre très prudente quant au problème de l'élargissement au logiciel des brevets.

Aux Etats-Unis, *il semble que les systèmes forts de propriété intellectuelle ont des effets négatifs*, dans les domaines où les innovations sont de nature séquentielle et où elles sont complémentaires. Au contraire, un système faible de propriété intellectuelle ouvre des possibilités d'imitation dont les effets sont bénéfiques.

les bonnes propriétés du logiciel libre

Le logiciel libre constitue un modèle de cumulativité des savoirs et de recombinaison des savoirs. Ses propriétés peuvent être décrites par les termes *d'intelligence distribuée* et *d'apprentissage par l'usage*. Elles sont proches des *systèmes de savoir ouvert* : la distribution rapide de la connaissance conduit à une meilleure coordination, à une baisse des risques de duplication, à un accroissement de la probabilité des découvertes, à une baisse du risque de l'appropriation des connaissances par des agents inefficaces.

En termes de performances des produits, ils peuvent donc être considérés comme fiables et de qualité. Le "système du bazar" du logiciel libre est source d'efficacité économique.

l'incitation dans un système d'innovation collective

Le succès du logiciel libre provient des utilisateurs non développeurs et de l'entrée d'entreprises commerciales qui s'appuient sur ce type de logiciels. Le problème qui se pose est cependant celui de la validité à long terme du modèle, alors que ses bénéficiaires s'appuient sur une communauté de développeurs sans principe de rémunération.

Les incitations individuelles sont principalement de trois ordres :

- les externalités d'usage
- l'apprentissage
- la réputation.

J.-B. Zimmermann *modélise les effets négatifs qui peuvent résulter de l'inexistence d'incitations pécuniaires*. Il considère : une communauté d'utilisateurs-développeurs. Parmi cette communauté, des individus qui coopèrent si les gains qu'ils en retirent (apprentissage et réputation) sont supérieurs au coût de la coopération.

Plus les individus qui coopèrent sont nombreux, plus les externalités positives qui en résultent en terme d'amélioration du logiciel sont fortes.

Avec la diffusion commerciale des logiciels, la communauté des utilisateurs-développeurs s'agrandit très fortement, alors que la fraction des coopérants augmente moins vite. Le gain relatif de la coopération s'en trouve réduit, ce qui peut conduire à un cercle vicieux de la défection.

Il faut donc *d'autres instruments d'incitation, en particulier de nature financière*. Une des suggestions est que les pouvoirs publics soutiennent le logiciel libre par l'embauche de jeunes développeurs.

Présentation de **Nicolas Jullien**, *les incitations à produire du libre*

Pourquoi les entreprises produisent-elles du logiciel libre ? Ceci pose la question de la production d'un bien public en dehors de la sphère marchande. Les incitations sont différentes selon les utilisateurs : non marchands (par exemple les universitaires) : dévoiler ne leur rapporte que des bénéfices, en termes de signal pour les recruteurs, de transfert du coût de la maintenance à la collectivité...

marchands : le coût de divulgation peut être élevé, d'où un arbitrage entre la diffusion et le secret.

Les producteurs peuvent avoir de nombreuses incitations à contribuer au libre, en particulier par les effets de signal en direction de leurs clients (qui peuvent les évaluer et résoudre ainsi les situations de sélection adverse) et par le développement de standards ouverts.

## Table ronde de synthèse

### Jean-Michel Dalle (ENSC)

Les débats ont montré que la soutenabilité du modèle économique du logiciel libre, ses avantages commerciaux, la qualité des produits... ne suscitent plus de doutes.

Une question émergente concerne le problème du financement de la R&D en matière de logiciels. Le problème du passager clandestin est inévitable ; le financement privé par capital - risque pose problème. Le financement peut-il alors être *public* ? Quels sont le cas échéant les logiciels bénéficiaires de ce type de financement ? La question de leur *utilité sociale* est déterminante dans la détermination des types de logiciels qui peuvent faire l'objet d'un financement de la communauté.

Autre source de débats, la complémentarité ou la substituabilité des modèles économiques. Ceci rejoint aussi des considérations de politique publique. Une des manières d'aborder le problème est d'avoir une approche qualitative et finement différenciée des composants de ce domaine.

Enfin, vers quels types d'analyses ces débats peuvent-ils mener ?  
vers des études monographiques ou des *business cases* sur des entreprises  
vers une appréhension des aspects juridiques de la question

### Bernard Lang (INRIA)

Les analyses ont présenté essentiellement les aspects micro-économiques du problème et ont négligé ses dimensions globales. Economiquement, et particulièrement à cause des externalités de réseau, un chef d'entreprise peut avoir intérêt à ne pas passer en logiciel libre ; l'agrégation de ces comportements individuels conduit cependant à une situation sous-optimale. Beaucoup des qualités des logiciels libres concernent la société dans son ensemble mais ne sont pas perceptibles par les individus : ceci est une justification forte de l'intervention de l'État.

De plus, la notion de passager clandestin demande à être relativisée ; en effet en matière de logiciel les externalités de réseau sont telles qu'un simple usager constitue à lui seul une contribution.

### Sylvain Allano (conseil en propriété industrielle)

Dans les rapports avec les clients, la question du logiciel libre ou propriétaire n'est en définitive pas centrale. Le problème est déplacé vers les services payants accessibles par le logiciel. La demande des clients concerne donc l'ingénierie de services.

L'outil du brevet conventionnel, créé pour des objets techniques et des procédés mécaniques, est difficilement utilisable. La suppression de l'exclusion de brevetabilité serait insuffisante ; les difficultés à exercer les droits attachés au brevet seraient de toute manière très grandes. Aux États-Unis, aucune exclusion n'est en vigueur : l'outil de brevet peut être en particulier utilisé pour des méthodes financières et commerciales. Pour faire cela en Europe, il faudrait revoir complètement la conception du brevet.

### Michel Bidoit (directeur du LSV, membre du bureau exécutif du RNTL)

Les conséquences concrètes de ces débats pourraient être :  
d'analyser quelques exemples de *business models* concrets. Ces études pourraient en particulier être diffusées dans une version courte dans la sphère politique

de clarifier (éventuellement toujours à l'usage des décideurs) ce qu'est le logiciel libre : il est nécessaire d'expliquer la double vision de connaissance et de produit ; d'expliciter la différence entre open source et gratuit.

Ces journées ont montré le passage d'un débat idéologique à un débat économique : le logiciel n'est pas un outil de philanthropie mais il est une des armes dans la bataille économique.

Il semble nécessaire de différencier la réflexion économique selon le type de logiciel, en particulier pour les logiciels dits d'intérêt ou de service public. On retrouve ici la nécessité d'intégrer la dimension macro-économique. Peu d'arguments ont été avancés en faveur d'une politique scientifique ou industrielle dans ce domaine : l'efficacité sociale du modèle peut-elle constituer une incitation dans ce sens ?

Il pourrait de plus être intéressant de *mesurer le coût réel de l'utilisation de logiciels propriétaires par les administrations par exemple.*

Laurent Kott (INRIA) fait cependant remarquer la difficulté inhérente à ce type de calcul : comment mesurer en particulier le coût de dépendance et le coût de changement ? De plus, l'administration publique doit prendre en compte aussi des retours en termes de taxes (TVA...), de création d'emplois... La maîtrise des technologies de l'information constitue un enjeu important ; il est délicat de décider à partir de quel moment la stratégie de l'open source devient une stratégie gagnante.