

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

Composants logiciels.

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

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 « puit 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.