

Epidémiologie des logiciels libres

Bastien Guerry

4 avril 2002

1 Introduction

Dans *La contagion des idées* (Sperber, 1996), Dan Sperber défend la pertinence du modèle épidémiologique pour l'analyse des représentations culturelles. Nous inspirant largement des arguments que Sperber invoque, nous voudrions ici soutenir que ce modèle épidémiologique est aussi le plus pertinent pour une explication causale de la propagation des logiciels libres. Par « logiciels libres », nous entendons tout logiciel dont le code source est accessible et librement redistribuable¹.

Ce que nous tentons ici est une analogie : pour ce qui est du problème de la diffusion, un logiciel libre est à l'ordinateur individuel ce qu'une représentation est à un individu. Notre analogie ne consiste pas à dégager une parenté de nature entre les représentations et les logiciels, ni à alimenter la rhétorique opposant systématiquement *software* et *hardware* : nous voulons simplement montrer que, même si les logiciels et les représentations culturelles sont des phénomènes essentiellement distincts, l'explication de leur diffusion d'après le modèle épidémiologique est la plus utile dans les deux cas, et pour les mêmes raisons. Notre propos n'est donc pas de nous interroger sur la formation d'une « culture technique », sur le mode de transmission des représentations publiques liées à ces objets techniques que sont les logiciels libres; nous prenons directement les logiciels libres comme objets de notre analyse, et cherchons à voir en quoi le modèle épidémiologique permet une explication causale de leur diffusion.

En anthropologie, la force du modèle épidémiologique est de nous rendre moins naïfs sur les contraintes écologiques et cognitives qui pèsent sur la diffusion des représentations publiques, de rendre obsolète l'idée d'une simple *réplication* des représentations. Nous espérons qu'à la fin de notre parcours, nous aurons déstabilisé l'idée qu'un logiciel libre est seulement quelque chose à « télécharger », objet

1. Il nous arrivera de dire simplement 'logiciel' quand le contexte indique clairement qu'il s'agit de logiciels libres.

d'une pure copie; pour cela, nous devons analyser les contraintes techniques qui pèsent sur leur mode de diffusion.

2 Motivation de l'analogie

Sperber propose que l'anthropologie se nourrisse des résultats de la psychologie comme l'épidémiologie s'appuie sur les données de la pathologie. C'est dire que pour comprendre la manière dont les représentations sont propagées dans une population, il ne suffit pas de s'intéresser à ces représentations, mais il faut aussi se pencher sur la manière dont les individus les reçoivent et les transforment. Nous suggérons que pour comprendre les mécanismes de diffusion des logiciels libres via Internet, il ne suffit pas de s'interroger sur le contenu de ces logiciels, mais il faut aussi se pencher sur la manière dont ils sont reçus et transformés par l'utilisateur et son ordinateur. Trois remarques serviront à motiver notre analogie. D'une part, les représentations et les logiciels sont, les uns comme les autres, des objets matériels. D'autre part, ces deux types d'objet s'échangent au sein d'une économie d'abondance. Enfin, les représentations et les logiciels libres ne sont pas seulement transmis, mais communiqués.

2.1 Matérialité

Dans ce qui suit, nous distinguerons représentations publiques et représentations mentales. Lorsque l'on me donne un livre, j'ai en main propre la représentation publique d'un contenu; lorsque je me souviens du livre que j'ai lu, j'accède à une représentation mentale de ce contenu.

Mais le terme de « représentation » nous pousse à deux erreurs : l'une serait de croire que les représentations publiques sont des entités mentales autonomes, chargées d'un sens propre; l'autre serait de confondre les représentations publiques avec de pures abstractions. La première erreur vient de ce que nos représentations mentales sont chargées d'intentionnalité, et que cette intentionnalité n'est pas traduisible telle quelle dans les représentations publiques. La deuxième erreur vient de ce que nous confondons la publicité de certaines représentations et la généralité de certains traits partagés par différentes représentations mentales. Ainsi, un conte écrit est une représentation publique qui est neutre du point de vue de l'intentionnalité, et qui est en lui-même tout à fait singulier. L'intentionnalité n'apparaît qu'au moment où quelqu'un s'approprie le conte pour en faire quelque chose; la généralité n'apparaît qu'au moment où différentes versions mentales du même conte ont certains traits en commun.

Ce qui nous importe ici est d'insister sur la matérialité des représentations, pu-

bliques ou mentales. La matérialité des représentations mentales est bien sûr sujette à caution, mais nous devons en faire l'hypothèse pour rendre possible une explication causale de la diffusion des représentations. Les représentations publiques sont aussi matérielles que les mémoires externes qui leur servent de support, de même que les représentations mentales sont aussi matérielles que - par hypothèse - le cerveau qui leur sert de support.

Dans le même esprit, on peut distinguer la version publique et la version privée d'un logiciel, toutes les deux matérielles. La version publique d'un logiciel libre est son code source, la version que nous appelons « privée » correspond au programme compilé tournant sur une machine particulière².

Les deux erreurs faites au sujet des représentations publiques pourraient être répétées au sujet des versions publiques d'un logiciel : on pourrait d'une part imaginer que la version publique porte en elle une finalité intrinsèque, en dehors même de son implémentation sur une machine particulière; on pourrait d'autre part imaginer que la version publique d'un logiciel est relativement « abstraite » par rapport à son implémentation concrète sur une machine. Pour ce qui est de la finalité, il va de soi qu'un logiciel n'a aucune utilité propre tant qu'il n'est pas implémenté; il est certes un agencement de fonctions, mais cet agencement n'est lui-même fonctionnel qu'une fois compilé ou interprété par la machine. Pour ce qui est de l'abstraction, il conviendrait peut-être de distinguer deux sens du terme « abstrait » : d'après un premier sens, l'abstrait s'opposerait au physique comme le virtuel s'oppose à l'actuel; d'après un second sens, il s'opposerait simplement à « déterminé ». Dire que la version publique d'un logiciel est abstraite est juste si l'on entend par là que toutes les variables du logiciel ne sont pas encore déterminées; mais cela devient faux si l'on veut suggérer qu'il y a en elle quelque chose de virtuel, d'immatériel.

Il peut sembler étrange de parler de la *matérialité* d'un logiciel : nous avons tellement appris à ne voir dans les logiciels que leurs interfaces utilisables, nous avons tellement entendu prononcer ce mot de « virtuel » pour qualifier les états de réalités propres au monde dans lequel sont supposés évoluer les logiciels, qu'il devient presque aussi choquant de parler de la matérialité des logiciels que de celle des représentations. Et pourtant, représenté en langage de haut niveau ou en langage machine, implémenté ou non, un logiciel ne contient rien qui ne soit, à un quelconque niveau, actuel et physique, de même qu'un livre n'est jamais fait que d'encre et de papier. Les langages de la virtualité et de la potentialité sont le plus souvent métaphoriques³, et jouent sur notre incapacité à en revenir au niveau matériel. Mais c'est à ce niveau que nous devons rester si nous nous intéressons au

2. Notre analogie se limite bien au logiciel libre : dans le monde du logiciel propriétaire, la version publique est souvent une version *déjà* précompilée.

3. Pour une tentative de mise au point sur ce qu'est le « virtuel », voir (Lévy, 1998).

modèle épidémiologique en vue d'une explication *causale* de la diffusion des logiciels.

2.2 Economie d'abondance

Sperber insiste sur la propension que nous avons à partager nos représentations (Sperber, 1996, p.131). Cette propension fait que nos esprits évoluent dans un monde saturé de représentations en tous genres. Du point de vue d'un même individu, le nombre de représentations publiques surpasse largement le nombre de représentations mentales. Si les représentations publiques devaient être comparées à des biens dont nous serions les consommateurs, cette économie serait donc une économie de l'abondance. Dans une telle économie, les lois de la concurrence ne sont plus tout à fait les mêmes; la consommation est d'autant plus sensible à la pertinence du produit consommé, et la production laisse place à la résurgence d'échanges de type don/contre-don. Devant la masse des représentations publiques, notre « consommation » rend de plus en plus nécessaire une évaluation de leur pertinence, c'est-à-dire du rapport entre l'*effort* cognitif qu'elles exigent et l'*effet* cognitif qu'elles produisent. De plus, cette abondance de représentations nous incite en retour à partager les nôtres, car leur valeur n'est plus tant dans leur prix intrinsèque que dans la reconnaissance qu'on pourra publiquement en tirer.

De manière analogue, les logiciels libres sont innombrables : non seulement le nombre de versions publiques d'un même logiciel libre est largement plus élevé que le nombre de machines implémentant ce logiciel, mais le nombre même de logiciels libres est immense au regard du nombre de machines. Cette prolifération intense est notamment rendue possible par des licences dont le prototype est la *General Public Licence - GPL* : schématiquement, celle-ci interdit à quiconque de rendre inaccessibles les sources d'un logiciel en se les réappropriant, et autorise tout un chacun à rediffuser librement la version publique d'un logiciel. On a coutume de désigner cette licence comme extrêmement « contagieuse ».

Ces considérations numériques floues laissent entrevoir qu'ici aussi l'économie sera une économie de l'abondance. Face à un choix très large de logiciels très spécifiques, l'utilisateur pourra à loisir sélectionner les logiciels dont les fonctionnalités correspondent au plus près à ses besoins. Et l'équivalent de la propension toute humaine au partage des représentations sera le partage du code source, ainsi que la diffusion de morceaux de codes nouveaux, pour ceux des utilisateurs qui sont aussi des développeurs. Cette diffusion, pour apparemment spontanée qu'elle se donne, entrera dans une logique du don/contre-don : le contributeur attendra en retour - consciemment ou non - une retribution en terme de reconnaissance⁴.

4. Pour un abord lucide sur le mécanisme de l'économie du don dans le monde du logiciel libre, cf. *L'économie du don high tech* de Richard Barbrook (Blondeau et Latrive, 2000, pp.141-170); pour

Le fait que la surabondance soit de mise pour les représentations comme pour les logiciels libres suggère l'idée que des facteurs « écologiques » de diffusion joueront de manière similaire dans les deux cas, comme nous le développerons par la suite.

2.3 Communication

« Toute représentation met en jeu au moins trois termes : la représentation elle-même, son contenu et un utilisateur; trois termes auxquels peut s'ajouter un quatrième : le producteur de la représentation lorsque celui-ci est distinct de la représentation. » (Sperber, 1996, p.49)

Ce cadre permet d'insister sur deux facteurs essentiels de la distribution des représentations : les facteurs cognitifs et les facteurs écologiques. Les facteurs cognitifs sont, par exemple, les capacités de mémoire et les différentes dispositions naïves des individus (ontologie naïve, physique naïve, psychologie naïve, etc). Ce sont eux qui contraindront le rapport entre la représentation et son contenu, qui régleront l'appropriation mentale du contenu de la représentation publique. Les facteurs écologiques sont les éléments de l'environnement communicationnel influants sur la distribution des représentations publiques : typiquement, les institutions sont des relais de distribution des représentations publiques qui tendent à homogénéiser les représentations mentales. Ainsi comprises, les institutions sont des points de convergence entre les producteurs de représentations publiques et leurs utilisateurs.

Cette distinction des facteurs cognitifs et des facteurs écologiques est rendue possible par la compréhension de la transmission des représentations comme *communication* : si les représentations se communiquent, alors leur compréhension est soumise au même principe de pertinence que n'importe quel acte communicationnel (cf. (Sperber et Wilson, 1989)). Relativement aux contraintes cognitives imposées à l'individu, une représentation sera d'autant plus pertinente qu'elle sollicite moins d'effort interprétatif et qu'elle suscite une modification plus marquée de son environnement cognitif. Relativement aux contraintes écologiques, une représentation sera d'autant plus pertinente qu'elle se véhicule facilement et qu'elle produit un effet décisif sur les individus avoisinants.

Le concept de pertinence permet alors d'expliquer pourquoi certaines représentations se répandent plus efficacement que d'autres. Par exemple, Pascal Boyer (cf. (Boyer, 1999)) explique pourquoi les « esprits » sont plus fréquents que les « zombies »⁵ : les deux correspondent à des représentations qui demandent peu

une vision plus naïve de la générosité et de la « coopération » sur le web, cf. *L'économie connectée* de D. de Kerckhove (de Kerckhove, 2000, pp.213-232).

5. Les zombies sont d'apparence ordinaire, mais n'ont aucune capacité mentale; les esprits

d'effort cognitif, car ces représentations sont formées à partir de violations réglées de nos catégories physiques et psychologiques ordinaires; mais l'invocation d'esprits ayant plus d'impact sur notre environnement cognitif (et notamment sur la croyance d'après laquelle nos intentions sont publiquement accessibles), celle-ci se trouve plus souvent pertinente que la représentation des zombis. De manière générale, Sperber montre comment les « mystères » gagnent en pertinence du fait de la richesse des inférences et des interprétations qu'ils suscitent.

Comment poursuivre l'analogie avec le logiciel libre ? Où retrouver le domaine d'application du principe de pertinence ? Nous avons déjà suggéré de rapprocher représentation publique et version publique, puis représentation mentale et version privée d'un logiciel libre. De même que l'utilisateur n'est aucunement passif dans la réception et la transmission des représentations, l'utilisateur n'est pas passif dans la réception et la transmission des logiciels libres. Ce que nous appelons 'utilisateur' est ici le complexe formé par l'individu et sa machine (voir 3.3).

Premièrement, il ne suffit pas de télécharger un logiciel pour pouvoir s'en servir, comme il ne suffit jamais d'être témoin d'une représentation pour se l'approprier. Aux facteurs cognitifs individuels correspondent ici les facteurs techniques liés à l'utilisateur et sa machine. Ces facteurs techniques forment une hiérarchie de contraintes, que l'on peut brièvement esquisser ainsi :

- celles liées à l'architecture de la machine;
- celles liées au système d'exploitation installé sur la machine;
- celles liées aux interfaces de communication des différents programmes avec le système d'exploitation;
- celles liées aux interfaces de communication des différents programmes entre eux;
- celles liées aux interfaces d'utilisation des différents programmes;
- celles liées aux capacités cognitives de l'utilisateur.

Ainsi, pour rendre compte de la diffusion d'un logiciel libre, nous devons tenir compte de son adaptabilité à ces différentes contraintes. Certains logiciels doivent leur succès à la manière dont ils s'adaptent à l'architecture d'une machine (comme ce fut le cas pour le noyau Linux, programmé au départ pour des machines d'architecture i386); d'autres à leur portabilité sur différents systèmes d'exploitation; d'autres au langage utilisé pour les écrire⁶; d'autres enfin à la manière dont leur interface les rend accessibles à une grande majorité d'utilisateurs. Quoiqu'il en soit, tous les niveaux de contraintes peuvent jouer simultanément et à divers degrés.

manquent au contraire d'un corps, et ont des capacités mentales extraordinaires.

6. Un logiciel en C ou C++ trouvera presque partout de quoi être compilé, ce qui n'est pas le cas d'un logiciel écrit en Rebol.

Deuxièmement, le milieu Internet n'est pas un milieu homogène au sein duquel toutes les transactions sont également possibles. Nous trouverons ici aussi des facteurs « écologiques » qui viendront régler la diffusion des logiciels. Ces facteurs écologiques sont multiples : comme dans le cas des représentations, ils interviennent surtout au niveau des vitesses de diffusion et de la constitution de bassins d'attraction pour les échanges, que ces bassins d'attraction correspondent à des institutions ou non. Dans un réseau dont le débit est faible, les logiciels trop lourds seront moins privilégiés que les programmes légers. Et dans un réseau foisonnant de sources diverses pour se procurer des logiciels, certains lieux centraux seront privilégiés. Citons par exemple les sites <http://freshmeat.net> et <http://sourceforge.net> qui sont devenus des références incontournables en matière de téléchargement de logiciels libres. Ces références, la plupart institutionnelles, sont des facteurs qu'une explication causale de la diffusion des logiciels libres se doit de prendre en compte.

Cette distinction entre facteurs techniques et facteurs écologiques permet de rendre compte du phénomène de « célébrité sociale » de certains logiciels libres. Comme pour les représentations, le principe de pertinence jouera sur le rapport entre effort (cognitif ou technique) et effet (cognitif ou technique). D'une part, un logiciel sera d'autant plus pertinent qu'il est adapté aux contraintes techniques liées à l'utilisateur et aux contraintes écologiques qui règlent sa diffusion dans le milieu Internet. Une bonne adaptabilité à ces contraintes minimise l'effort demandé à l'utilisateur pour installer et utiliser le logiciel sur sa machine. D'autre part, un logiciel sera d'autant plus pertinent qu'il enrichit les fonctions auxquelles l'utilisateur a accès. Comme le dit Eric Raymond dans (Raymond, 1999), « on reconnaît un outil vraiment excellent au fait qu'il se prête à des usages totalement insoupçonnés. »

3 Mémétique versus épidémiologie

Maintenant que nous avons explicité l'arrière-fond motivant notre analogie, passons à la comparaison entre modèle des mèmes et modèle épidémiologique. L'argumentation de Sperber contre l'utilisation du modèle des mèmes en anthropologie consiste à montrer que celui-ci ne parvient pas à résoudre deux difficultés majeures : d'une part celle concernant l'origine multiple de nos représentations, d'autre part celle liée aux transformations que les représentations publiques subissent quand elles sont « mentalisées » et reproduites par les individus. Nous allons montrer que ce sont aussi ces deux difficultés qui s'imposent à une mémétique des logiciels libres.

3.1 Le modèle des mèmes

Le modèle des mèmes a été originairement conçu par Richard Dawkins (Dawkins, 1982) et largement popularisé en philosophie de l'esprit par Daniel Dennett (Dennett, 1991). Mais il convient de montrer tout de suite combien ces deux auteurs abordent les mèmes différemment; une fois ces différences dégagées, nous verrons ce que leur vision partage de présupposés, lesquels seront au centre des objections de Sperber.

Pour Richard Dawkins, les mèmes sont des représentations mentales individuelles. Celles-ci se propagent d'individus à individus, subissant de-ci de-là des mutations. Un tel mode de propagation des mèmes permet de traiter leur diffusion avec les mêmes outils que ceux dont se sert la génétique des populations. Cette approche donne lieu à une explication des faits culturels comme concentration d'une population de mèmes semblables. Pour Daniel Dennett et la perspective wittgensteinienne à laquelle il entend être fidèle, le statut des représentations mentales est essentiellement problématique. D'après sa théorie de la conscience comme « centre de gravité narratif », comme lieu de rencontre entre les *multiple drafts*, les mèmes ne peuvent être que des représentations publiques. Il n'y a pas de représentation mentale située sur un « théâtre cartésien » de la subjectivité, mais une connaissance des règles d'usage des différentes représentations publiques accessibles à l'individu. Ainsi, Dawkins part des représentations mentales et s'intéresse à leur mode de distribution pour expliquer les phénomènes culturels, alors que Dennett part des représentations publiques et s'intéresse à leur mode d'intégration pour expliquer les phénomènes mentaux. Bien sûr, les représentations mentales de Dawkins restent des entités *objectives*, sans quoi elles ne pourraient pas être manipulées comme éléments dans le calcul de leur fréquence; et les représentations publiques de Dennett sont tout de même des entités *subjectives*, sans quoi elles ne pourraient pas entrer dans une explication de la conscience.

Mais voici le présupposé commun de ces deux auteurs : qu'ils correspondent à des entités strictement mentales ou seulement publiquement accessibles, les mèmes se reproduisent par simple *réplication* d'un individu à un autre. Dans les deux cas, les mèmes font certes l'objet d'une pression sélective à l'issue de laquelle ne survivent que ceux présentant un quelconque intérêt adaptatif : cependant, ce n'est jamais que leur seule fréquence qui varie. Leur transmission n'implique aucune transformation intrinsèque; seuls des mécanismes aléatoires de mutation doivent alors expliquer la variabilité culturelle.

Dans le monde du logiciel libre, on rencontre souvent le parallèle entre code source et code génétique, évolution des êtres vivants et évolution des logiciels. Ce parallèle repose sur l'idée que, de même que la pression sélective exercée par l'en-

vironnement influe indirectement sur le code génétique des êtres vivants, la pression sélective exercée par les utilisateurs modifie indirectement le code source des logiciels. Nous obtenons à terme une meilleure adaptation des logiciels libres aux exigences des utilisateurs, comme la nature obtient une meilleure adaptation des êtres vivants à leur environnement, et ce par la seule modification des *programmes* - génétiques ou informatiques.

Pour tentante que soit cette comparaison, elle nous semble trop hâtive. Car une telle mémétique élémentaire de la diffusion des logiciels libres prendrait le logiciel comme un tout reproductible tel quel sur les différentes machines. Cette conception nous mènerait à croire que l'acquisition d'un logiciel n'est soumise qu'aux seules exigences ordinaires d'utilité, comme si les exigences de l'utilisateur n'étaient pas déjà contraintes par un ensemble de considérations techniques. La seule opération de téléchargement résumerait à elle seule tout le processus de la diffusion. C'est peut-être une telle vision que Pascal Boyer a en tête quand il écrit :

« Cultural transmission, like other forms of human communication, does not consist in a 'downloading' of concepts from one mind to another. » (Boyer, 1999, p.1)⁷

Une telle image de la diffusion des logiciels libres tomberait sous les mêmes critiques que la conception mémétique en anthropologie. Faire abstraction de l'ensemble des contraintes (techniques ou écologiques) qui pèsent sur la transmission des logiciels, concevoir le couple ordinateur-utilisateur comme un support passif qui « reçoit » un logiciel, conduit à manquer l'explication de trois faits. D'une part, le fait que des logiciels puissent intégrer des morceaux de code provenant d'autres logiciels, soit dans le corps de leur programme principal, soit de manière modulaire; d'autre part le fait que tous les logiciels ne soient pas installables sur tous les ordinateurs; enfin le fait que des logiciels puissent être écrits via la contribution de personnes éparpillées dans le monde. Le modèle mémétique oblige à prendre le logiciel comme un tout, et cette obligation rend impossible une explication de leur transmission mettant en relief les transformations qu'ils sont susceptibles de subir.

3.2 Le modèle épidémiologique

Le modèle mémétique pêche par trop d'abstraction : abstraction des représentations par rapport à leur contenu réel, variable selon les individus; abstraction des représentations par rapport à leurs conditions de transmission, variables selon les contextes. Dans le cas des logiciels libres, l'approche mémétique serait aussi doublement abstraite : abstraction des logiciels par rapport aux contraintes techniques

7. P. Boyer peut bien sûr parler de n'importe quelle donnée téléchargeable; mais le rapprochement entre le niveau logiciel et le niveau conceptuel est si courant que l'image simpliste que nous imaginons est très probable.

d'installation et d'utilisation, variable selon les machines et les utilisateurs; abstraction par rapport à leur conditions de diffusion, variable d'une région du réseau à une autre.

Une première correction de cette abstraction pourrait consister à prendre en compte la « fusion » de diverses représentations en une représentation synthétique nouvelle, et l'intégration de morceaux de codes épars en un nouveau programme. L'invocation d'influences multiples entraînant la production de représentations ou de logiciels nouveaux pourrait effectivement donner une réponse au problème des origines multiples, mais il ne permettrait toujours pas de comprendre quelles sont les différentes contraintes qui pèsent sur cette « fusion » des influences. L'idée une opération de *morphing*⁸ est encore trop vague pour rendre compte précisément de la formation de nouvelles représentations; et l'idée d'une simple intégration de morceaux de code est absurde, à moins de spécifier les conditions dans lesquelles cette intégration est possible.

A l'issue de cette critique du modèle des mêmes et du modèle de l'influence, nous avons une meilleure idée du défi que doit relever le modèle épidémiologique : pour ce qui est de l'anthropologie, il doit nous permettre d'expliquer comment les individus sélectionnent et transforment les représentations selon leur pertinence; pour ce qui est des logiciels libres, il doit nous faire comprendre comment a lieu la sélection et la transformation des logiciels. Dans le cas de l'anthropologie, ce seront des considérations psychologiques qui détermineront les limites de la pertinence possible d'une représentation; dans celui des logiciels libres, ce seront des remarques d'ordre technique qui souligneront les contraintes rencontrées dans le choix des logiciels.

Soit la distinction que fait Sperber entre croyances intuitives et croyances réflexives (Sperber, 1996, p.123). Il désigne les croyances intuitives comme des produits de processus perceptuels et d'inférences spontanées, inconscientes. Les croyances réflexives sont quant à elles des interprétations de représentations, des croyances de second ordre, enchâssées à l'intérieurs d'autres croyances (intuitives ou réflexives). Les croyances intuitives correspondent à des dispositions innées et culturellement enrichies; l'explication de la manière dont elles sont répandues fera essentiellement appel à des facteurs psychologiques. A l'inverse, les croyances réflexives sont plus sensibles aux contextes dans lesquels elles sont communiquées. Par exemple, la croyance réflexive en un Dieu tout-puissant est sensible à l'autorité de la personne qui la communique, ainsi qu'à la confiance dont cette autorité est investie. Pour les croyances réflexives, il faudra donc s'intéresser principalement aux facteurs écologiques de distribution. Aidé de cette simple distinction entre

8. L'expression est de Sperber : (Sperber, 1996, p.146).

croyances intuitives et croyances réflexives, le modèle épidémiologique pourra expliquer la répartition des unes et des autres en se demandant chaque fois quelles sont les conditions cognitives et écologiques de leur pertinence, sachant que des croyances intuitives sont conditions de la pertinence possible des croyances réflexives, et que celles-ci peuvent être à leur tour conditions de pertinence d'autres croyances réflexives.

Dans le cas des logiciels libres, le modèle épidémiologique permet d'exprimer trois faits. D'abord, comme nous l'avons déjà dit, il permet de considérer les chances de propagation⁹ d'un logiciel libre au regard de son adéquation aux contraintes techniques de chaque ordinateur. Ensuite, il permet de rendre compte du fait qu'un logiciel libre est plus ou moins facilement accessible selon la manière dont il est référencé dans le réseau Internet, rendu ou non « visible » par le biais de sites institués. Enfin, ce modèle prend en considération la possibilité pour un individu de contribuer à la modification du code source d'un logiciel, comme un individu malade modifierait un virus avant de le transmettre, ou comme un individu modifierait une représentation avant de la communiquer. Pour ce troisième fait, mentionnons par exemple des outils comme *CVS (Concurrent Version System)*, lesquels organisent cette possibilité qu'ont les utilisateurs de contribuer au développement d'un logiciel libre¹⁰.

3.3 Portée et limites du modèle

On pourrait objecter à Sperber que le modèle épidémiologique n'est pas parfaitement adéquat à l'idée qu'il donne de la transmission des représentations : en effet, à force d'insister sur le fait que la transmission *fidèle* est l'exception plutôt que la règle, Sperber néglige le fait que dans la plupart des maladies microbiennes ou virales (celles que nous avons spontanément à l'esprit quand nous parlons de « contagion »), le microbe ou le virus transmis est très semblable au microbe ou virus reçu. A cela, Sperber pourrait répondre qu'il ne veut pas se focaliser sur ce genre de maladies. Ce qui l'intéresse justement dans le modèle épidémiologique, c'est le fait qu'il n'existe pas de théorie *générale* de épidémiologie; il faut chaque fois faire l'épidémiologique spécifique de chaque maladie séparément. De même, Sperber ne cherche pas une théorie générale de la communication des représentations culturelles : l'invocation du modèle épidémiologique lui sert au contraire à montrer qu'il faudra proposer différentes théories pour les différents types de représentations.

9. Cette propagation n'est pas spontanée : nous ne parlons bien ici que de logiciels, non de virus. Pour une comparaison éclairante entre virus biologique et virus informatique, voir (Delahaye, 1999).

10. Pour une analyse plus détaillée de la manière dont le logiciel libre transforme la vision classique du développement informatique, voir (Guerry, 2001).

Ce que cette objection possible doit nous faire ici remarquer, c'est que le modèle épidémiologique semble encore plus justifié pour ce qui est des logiciels libres que pour ce qui est des représentations. Dans le cas des représentations, du moins telle que Sperber les aborde de manière assez générale et spéculative, les transformations semblent être si fréquentes qu'il est d'avance désespérant de s'intéresser au détail des capacités psychologiques qui les déterminent. Dans le cas des logiciels libres, pour peu que nous parlions de deux ordinateurs ayant la même architecture, le même système d'exploitation, les mêmes bibliothèques de fonctions graphiques et sonores, et les mêmes variables d'environnement, il y a de fortes chances pour que le passage de la version publique à la version privée soit le même dans les deux cas. En outre, la participation à la modification du code source d'un logiciel libre est la plupart du temps indirecte¹¹ et marginale. Les transformations qu'un virus ou un microbe subissent dans le passage d'un corps à un autre a donc plus de ressemblance avec les transformations limitées que nous retrouvons dans les procédures d'installation et de modification d'un logiciel que dans les moyens de mentalisation et de reproduction d'une représentation publique.

Mais cette façon optimiste de montrer l'adéquation du modèle épidémiologique pour l'explication de la diffusion des logiciels libres négligerait un point important de l'analogie que nous voulons construire. Dans le cas de la distribution des représentations, les complications arrivent dès qu'on s'intéresse au fait que l'agent est non seulement psychologiquement déterminé, mais qu'il est en plus doué d'intentionnalité. De même, nous avons tout à l'heure proposé de nous intéresser non seulement aux caractéristiques techniques de la machine, mais aussi aux caractéristiques cognitives de son utilisateur. Avec l'utilisateur se réintroduisent les notions d'utilité, de finalité et d'intentionnalité. Si bien que ce que nous disions à l'instant du passage de la version publique à la version privée d'un logiciel pourrait aussi être dit, pour ce qui est des représentations, de deux individus dont nous savons qu'ils partagent exactement les mêmes capacités psychologiques : leur manière de recevoir et de reproduire les représentations serait alors la même. Mais des individus aux capacités psychologiques indiscernables peuvent néanmoins être doués d'intentions différentes, et des machines parfaitement identiques peuvent être à la disposition d'utilisateurs dont les besoins sont distincts. Aussi faut-il mesurer la portée du modèle épidémiologique dans le cas où nous tenons compte de l'individu (devant les représentations) et du couple utilisateur-machine (devant les logiciels libres) dans toute leur complexité.

Pour ce qui est de la problématique anthropologique, il paraît justifié de s'intéresser à l'homme comme à un tout : ce ne seront pas les seules limitations cog-

11. Lorsqu'on soumet, par exemple, un bug à l'auteur du logiciel.

nitives d'un individu qui détermineront la pertinence d'une représentation; ses intentions entreront aussi en jeu. Pour ce qui est de notre problématique, il paraît en revanche douteux de traiter le complexe utilisateur-machine comme un tout. Or il nous semble au contraire que c'est de ce point de vue que le modèle épidémiologique a toute sa pertinence.

Dans une tentative de définition de la réalité virtuelle (Verna, 1999), Didier Verna construit le modèle *M_{PI}C* (acronyme de **M**odèle **R**éprésentatif de l'**I**nteraction **C**ognitive). Ce modèle formalise de manière assez détaillée les différents niveaux d'interaction possibles entre l'agent mental et l'agent physique. L'agent mental est divisé en deux niveaux :

- l'agent de contrôle (niveau de l'intentionnalité);
- l'agent de transcription (niveau psychologique et pragmatique, incluant les perceptions et les actions).

L'agent physique est aussi divisé en deux niveaux :

- l'agent opératoire (interface physique entre l'agent et son environnement);
- l'agent externe (l'environnement physique).

Des boucles d'interactions lient entre eux les différents agents. Alors qu'une définition réductrice de la réalité virtuelle définirait celle-ci comme la mise en place d'un environnement simulé au lieu d'un environnement réel, le modèle proposé par Didier Verna lui donne les moyens d'une définition plus fine : il définit la virtualisation comme le remplacement de l'un des agents par une autre instance que l'instance réelle. Même si elle est motivée par le problème spécifique de la réalité virtuelle, cette approche nous intéresse en ceci qu'elle insiste sur le fait que, pour représenter les interactions entre l'homme et la machine, on ne peut plus négliger la manière dont les capacités cognitives de l'utilisateur jouent étroitement avec les caractéristiques techniques de la machine. D'une façon générale, la représentation de l'interaction homme-machine connaît une évolution : alors que nous concevions auparavant l'interaction homme-machine exclusivement sur le modèle du dialogue homme-homme, on en vient aujourd'hui à le concevoir de plus en plus sur le modèle du rapport conscience-subconscient¹².

Nous pensons donc que le modèle épidémiologique pourrait rendre compte de la diffusion des logiciels libres non seulement selon leur adéquation aux contraintes purement techniques liées à leur installation sur une machine particulière, mais aussi selon la manière dont l'utilisateur est disposé à interagir avec le logiciel. Par exemple, selon qu'un utilisateur est disposé à se servir de sa souris ou à n'utiliser que le clavier, ou bien selon qu'un utilisateur préfère le mode graphique ou le mode console, la pression sélective sur la diffusion du logiciel ne sera pas la même.

12. Voir (Pyke et West, 2001) pour une justification de cette évolution et (Verna, 2000) pour un exemple dans le domaine de la reconnaissance d'action.

Remarquons que dans cette perspective, la modularité d'un logiciel est un atout essentiel. Pour un développeur, faire le choix d'un logiciel largement modulaire, c'est anticiper une double volonté de liberté des utilisateurs : liberté de configurer à loisir le logiciel pour l'adapter à ses besoins, et liberté d'enrichir le logiciel avec de nouvelles fonctionnalités. La modularité est le moyen d'optimiser le rapport entre les contraintes cognitives de l'utilisateur et les contraintes techniques du logiciel, car elle permet de limiter la profondeur d'exploration du code source que l'utilisateur doit atteindre pour configurer le logiciel, et elle facilite l'exploitation de nouvelles fonctions (car leur intégration dans le logiciel est déjà prévue). Pour ne citer que deux exemples de ce développement modulaire à succès : le serveur *Apache* et l'éditeur de textes *Emacs*.

4 Exemple d'analyse

Pour rendre plus explicite l'analogie que nous tentons, voici deux tableaux : le premier résume une analyse présentée par Sperber, le deuxième montre en regard une analyse analogue pour le point de vue qui est le nôtre.

4.1 Mythe, Egalité, Gödel

	Mythe	Egalité des hommes	Gödel
Facteurs cognitifs	Forts	Faibles	Forts
Facteurs écologiques	Faibles	Forts	Forts

Pour le mythe : le facteur cognitif (mémoire) est prépondérant par rapport au facteur écologique (autorité des anciens).

Pour le principe d'égalité des droits de l'homme : le facteur écologique (le fait que ce principe devienne pertinent pour un grand nombre d'individu) est prépondérant par rapport au facteur cognitif (l'intelligibilité du principe).

Pour le théorème de Gödel : les facteurs cognitifs (connaissances mathématiques) et les facteurs écologiques (milieux universitaires) posent tous deux de fortes contraintes sur la diffusion de cette représentation.

4.2 Microsoft Word, Emacs, Gedit

	Microsoft Word	Emacs	Gedit
Facteurs techniques	Forts	Faibles	Forts
Facteurs écologiques	Faibles	Forts	Forts

Microsoft Word, *Emacs*, et *Gedit* sont respectivement un logiciel de traitement de texte et deux éditeurs de texte. Le premier n'est pas un logiciel libre : nous nous y référons cependant pour rendre l'analyse plus parlante. Emacs est le célèbre éditeur écrit principalement par Richard Stallman, fondateur de la *FSF (Free Software Foundation)* en 1984. *Gedit* est un petit éditeur de texte fourni avec l'environnement de travail *Gnome*, projet libre tournant essentiellement sous GNU/Linux.

Pour Microsoft Word : les contraintes techniques sont prépondérantes par rapport aux contraintes écologiques. Parmi les contraintes techniques : nécessité d'avoir un PC tournant sous MS-Windows - parfois même nécessité d'avoir telle version du système d'exploitation plutôt que telle autre. Le facteur écologique est de peu de poids - notamment à cause du quasi-monopole de ce logiciel de traitement de texte sur le marché.

Pour Emacs : les facteurs techniques sont moins déterminants que les facteurs écologiques. Parmi les contraintes techniques : *Emacs* est supporté par de nombreux systèmes d'exploitation différents; son système d'encodage *Mule (MUlti Lingual Environment)* le rend universellement accessible. Ce sont donc les facteurs écologiques (visibilité sur Internet, accessibilité auprès du grand public) qui seront davantage déterminants pour rendre compte de sa distribution.

Pour Gedit : les facteurs techniques et écologiques sont tous les deux forts. Pour les facteurs techniques : *Gedit* nécessite non seulement que l'ordinateur tourne sous GNU/Linux, mais encore que les bibliothèques graphiques de l'environnement *Gnome* soient installées sur la machine. Pour les facteurs écologiques : la distribution de *Gedit* est liée à celle du projet *Gnome* dans son ensemble, lequel est son institution de référence pour le développement comme pour le téléchargement. Ce lien entre *Gedit* et *Gnome* entraîne de fortes contraintes écologiques sur la diffusion de cet éditeur.

Remarquons pour finir qu'en plus des contraintes techniques que nous avons considérées ici, l'analogie que nous proposons devraient intégrer les contraintes cognitives liées à l'interaction entre l'utilisateur et sa machine : est-ce que tel logiciel impose un environnement manipulable à l'aide de la seule souris ? est-ce que tel logiciel a une interface largement configurable ? quel est le degré de modularité du logiciel ? Ce sont des questions de ce genre auxquelles il faudrait répondre pour une analyse plus détaillée.

5 Conclusion

Nous avons tenté de montrer l'insuffisance du modèle des mèmes pour une explication causale de la propagation des logiciels libres. Cette insuffisance nous a amené à mesurer l'adéquation d'un modèle épidémiologique pour cette explication. Ce modèle permet de tenir compte des contraintes cognitives, techniques et écologiques qui s'imposent à la diffusion d'un logiciel. Alors que le modèle des mèmes nous oblige à prendre le logiciel comme un tout immuable, le modèle épidémiologique prend en compte la transformation de la version publique en version privée, ainsi que les possibilités de contribution des utilisateurs à la modification de la version publique.

L'intérêt des critiques de Sperber est immédiat : il s'agit de contrer une image naïve de la diffusion culturelle. De ce point de vue, notre analogie est de peu d'intérêt, car l'image naïve de la manière dont les logiciels libres se diffusent n'a aucune conséquence dans un quelconque champ de recherche.

Néanmoins, il nous semble que cette analogie devient intéressante si nous ne nous intéressons plus seulement à la diffusion des logiciels libres, mais si nous nous intéressons aussi à la formation d'une *culture technique*, avec le sens fort que Gilbert Simondon donne à cette expression (cf. (Simondon, 1958)). Le fait que les contraintes techniques prévalent sur les contraintes écologiques dans le cas de l'éditeur *Emacs* illustre le fait que l'utilisateur est plus à même de se faire une *représentation technique* adéquate du logiciel, d'entrer à l'intérieur du programme pour « mettre les mains dans le cambouis ». Cette relative transparence de fonctionnement des logiciels libres est une condition nécessaire (quoiqu'insuffisante) de la formation d'une culture technique : l'accès au code source ne détermine pas seulement des modes de diffusion différents, mais aussi des modes de représentation technique nouveaux.

Table des matières

1	Introduction	1
2	Motivation de l’analogie	2
2.1	Matérialité	2
2.2	Economie d’abondance	4
2.3	Communication	5
3	Mémétique versus épidémiologie	7
3.1	Le modèle des mêmes	8
3.2	Le modèle épidémiologique	9
3.3	Portée et limites du modèle	11
4	Exemple d’analyse	14
4.1	Mythe, Egalité, Gödel	14
4.2	Microsoft Word, Emacs, Gedit	14
5	Conclusion	16

Références

- Blondeau, O. et Latrive, F. (2000). *Libres enfants du savoir numérique*. Editions de l’Eclat, Paris.
- Boyer, P. (1999). « Functional Origins of Religious Concepts: Conceptual and Strategic Selection in Evolved Minds ». *Journal of the Royal Anthropological Institute*, 6:195–214. Malinowski Lecture.
- Dawkins, R. (1982). *The Extended Phenotype*. Oxford University Press, Oxford.
- de Kerckhove, D. (2000). *L’intelligence des réseaux*. Editions Odile Jacob, Paris.
- Delahaye, J.-P. (1999). *Logique, informatique et paradoxes*. Editions Belin, Paris.
- Dennett, D. C. (1991). *Consciousness Explained*. Little, Brown and Company.
- Guerry, B. (2001). « Logiciel libre et innovation technique ». Devoir pour un DEA de philosophie.
- Lévy, P. (1998). *Qu’est-ce que le virtuel?* Editions La Découverte & Syros, Paris.
- Pyke, A. A. et West, R. L. (2001). « The Conscious-Subconscious Interface: An Emerging Metaphor in Human-Computer Interaction ». Submitted to Cognitive Science.
- Raymond, E. S. (1999). *The cathedral and the bazaar*. O’Reilly.

- Simondon, G. (1958). *Du mode d'existence des objets techniques*. Editions Aubier Montaigne, Paris.
- Sperber, D. (1996). *La contagion des idées*. Editions Odile Jacob, Paris.
- Sperber, D. et Wilson, D. (1989). *La pertinence: communication et cognition*. Editions de Minuit, Paris.
- Verna, D. (1999). « Définir le Virtuel: une Vision Cognitive ». Dans *ReViCo'99, Réalité Virtuelle et Cognition*, ENST, 46 rue Barrault, 75013 Paris, France.
- Verna, D. (2000). « Action Recognition: How Intelligent virtual environments Can Ease Human-Machine Interaction ». Dans Thwaites, H. et Refsland, S. T., éditeurs, *VSMM 2000, Sixth International Conference on Virtual Systems and MultiMedia*, pages 703–713, Gifu Research and Development Foundation, VR Techno Plaza, Sue-Chou 4-179-1, Kakamigahara, Gifu 509-0108, Japan. International Society on Virtual Systems and MultiMedia, Ohmsha Press.