Entre marché et communauté : une discussion de la culture participative à l'exemple de Google Maps

Bernhard Rieder

▶ **To cite this version:**

Bernhard Rieder. Entre marché et communauté : une discussion de la culture participative à l'exemple de Google Maps. Ludovia 2008 : Do it yourself 2.0, Aug 2008, France. pp.282-292. sic_00329899

HAL Id: sic_00329899

https://archivesic.ccsd.cnrs.fr/sic_00329899

Submitted on 13 Oct 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Entre marché et communauté : une discussion de la culture participative à l'exemple de Google Maps

Bernhard RIEDER

Laboratoire Paragraphe (EA 349)
Université de Paris VIII

71ème section

bernhard.rieder@univ-paris8.fr
http://bernhard.rieder.fr

MOTS-CLES :

culture participative, open source, google maps, API, amateurs professionnels

RESUME :

Dans cet article nous présentons le cas de l'API de Google Maps *comme un exemple du phénomène plus large de l'émergence d'une « culture participative » dans laquelle la production par des « amateurs professionnels » joue un rôle central. A travers une interface de programmation, ce service de cartographie en ligne permet aux développeurs de l'intégrer dans leurs propres applications Web. Un examen du travail fournit par une communauté de volontaires dégage une productivité non rémunérée considérable dont nous présentons les motivations et nous discutons la valeur pour l'entreprise* Google. *Notre but consiste à exposer cette culture participative comme une « situation stratégique complexe » qui échappe à une logique binaire opposant simplement « ouvert » à « fermé ».*

INTRODUCTION

Dans le chapitre le plus commenté de *La Dialectique de la Raison*, Horkheimer et Adorno [1947] affirment que la production industrielle implique une telle schématisation des artefacts culturels que l'audience est dispensée du travail d'interprétation parce que tout est connu d'avance. La consommation quotidienne de ces objets produirait donc des consommateurs passifs dont la faculté d'imagination serait sous-développée et abîmée. Il n'est donc pas étonnant qu'aux annonces d'une fin de l'ère industrielle au début des années 1970 suive l'apparition de néologismes comme « *prosumer* » [Toffler 1980] ou « consom'acteur » qui généralisent en quelque sorte la critique du concept de l'audience passive formulée déjà par des auteurs comme Richard Hoggart ou Roland Barthes dans les années 1960. Quand John Perry Barlow [1996] déclare « l'indépendance du cyberespace » il s'adresse par conséquent aux gouvernements « du monde industriel », ces « géants fatigués de chair et d'acier », en avançant comme différence ontologique entre ce monde industriel révolu et la nouvelle frontière numérique, le fait que sur Internet « tout ce que l'esprit humain est capable de créer peut être reproduit et diffusé à l'infini sans que cela ne coûte rien ». Cette idée d'une reconfiguration profonde des réglés du jeu économique par la technologie était au centre

même de la nouvelle économie et nous la retrouvons – déclinée mais intacte – dans maints discours autour du « Web 2.0 ». Des termes comme « *ProAms* » (amateurs professionnels) [Leadbeater, Miller 2004] ou « *produsage* » [Bruns 2008] célèbrent la figure d'un utilisateur *actif*, non seulement par son *interprétation* d'un artefact culturel mais surtout par le rôle croissant qu'il joue dans la *production* matérielle de ces objets. Or, comme le démontre Henry Jenkins [2006], les nouveaux espaces de création et de partage qui apparaissent en ligne s'imbriquent très souvent dans l'univers familier de la production culturelle établie. L'émergence de telles configurations hybrides entre « marché » (entreprises commerciales traditionnelles) et « communauté » (usagers producteurs) peut-être examinée de façon privilégiée à travers l'exemple des « *mashups* », des sites qui combinent des fonctionnalités ou données de plusieurs sources existantes en une seule application.

Dans cet article nous prenons comme cas exemplaire le service Web le plus utilisé dans la création de ces *mashups* : l'API[1] de *Google Maps*[2]. Notre but consiste à montrer que les rapports entre les fournisseurs des outils, qui permettent aux usagers de « faire eux-mêmes » et les communautés d'« usagers producteurs », qui se regroupent autour de ces outils sont complexes et intrinsèquement ambivalents. La nouvelle « culture participative » [Jenkins 2006] s'inscrit dans des contextes économiques, culturels et techniques particuliers où les marges de manœuvre des différents acteurs sont le résultat de processus de négociation permanents. La production amateur s'insère donc dans les configurations de production existantes tout en les transformant ; les constellations qui émergent nous imposent des ajustements conceptuels et nous renvoient à l'analyse empirique. Pour situer le phénomène il faut d'abord clarifier deux aspects de la création du logiciel.

1 UN CONTEXTE DE CREATION PARTICULIER

Bien que l'essor de la production amateur soit un phénomène embrouillé dont l'explication demande une approche multifactorielle, il nous semble que la composante technologique joue un rôle crucial de *facilitateur* qui est principalement dû à deux qualités du numérique.

1.1 La machine universelle et le réseau

L'ordinateur est une machine *universelle* qui peut se transformer en un nombre infini de machines *spécifiques* selon le programme qu'il exécute ; la création de programmes est principalement un travail d'écriture qui remplace, selon Turing [1948], le travail d'ingénieur de créer une machine physique pour chaque tâche par le « travail de bureau » de programmer la machine universelle. Pour devenir *producteur* de logiciel, il suffit d'un ordinateur, de temps libre et surtout des connaissances techniques nécessaires. L'investissement en capital et énergie étant négligeable, le domaine numérique est en quelque sorte prédestiné à l'émergence de formes alternatives de production.

A cette « légèreté »[3] du logiciel s'ajoute une infrastructure de communication, coopération et distribution redoutable : sur Internet des plateformes pour le développement logiciel

[1] En programmation, une API (*Application Programming Interface*) décrit un ensemble de fonctions qui peuvent être appelées par un autre logiciel. Pour créer une fenêtre sous *Microsoft Windows* par exemple, un développeur n'est pas obligé de le dessiner pixel par pixel sur l'écran ; il suffit d'appeler la classe *CreateWindow* qui fait partie de l'API de *Windows* et définit une fenêtre standard avec son comportement habituel.

[2] http://maps.google.com et http://code.google.com/apis/maps/

[3] Il faudrait ajouter qu'étant essentiellement *écriture*, le logiciel peut être copié, réutilisé et transformé avec très peu d'effort [Rieder, Schäfer 2008].

coopératif comme *SourceForge.net*, en combinaison avec toute une série de canaux de communication (*mailinglists*, *newsgroups*, forums, etc.), ouvrent la possibilité à des groupes géographiquement et socialement dispersés de communiquer, de travailler ensemble et d'échanger leurs œuvres. A travers le réseau se constituent des *usines virtuelles distribuées*.

La création en ligne n'est évidemment pas limitée au logiciel mais ce domaine joue le rôle d'avant-garde dans l'émergence d'une culture de production amateur. Pour comprendre pourquoi, il ne suffit cependant guère de nommer les facteurs techniques ; le développement historique de l'informatique pointe vers d'autres pistes explicatives.

1.2 Les trois crises du logiciel

Dans les premières décades de son existence, l'informatique était un domaine essentiellement expérimental. L'accès aux quelques ordinateurs en fonction était médiatisé par des techniciens experts et la programmation n'était guère l'activité interactive structurée qu'elle est aujourd'hui. Concernant la méthodologie, on s'orientait sur les mathématiques dont la façon d'opérer est peu formalisée sur le plan de l'organisation du travail. Or, cette approche bien adaptée aux calculs requis par les fins militaires rencontrait des problèmes importants au fur et à mesure de l'insertion progressive de l'informatique dans le monde commercial. En 1968, l'OTAN sponsorise donc une conférence de deux semaines en Allemagne pendant laquelle le « génie logiciel » (*software engineering*) – une méthodologie de travail très structurée s'inspirant des sciences de l'ingénieur – est défini et avancée comme remède aux grosses difficultés d'accommoder la complexité croissante du logiciel et la nécessité de travailler en groupe. La première crise du logiciel est ainsi avertie.

Dans les années 1980 l'arrivée des interfaces graphiques amène une nouvelle population vers l'informatique, des usagers résolument non experts dont l'accommodation n'est pas vraiment un problème technique à proprement parler. Cette deuxième « crise » est remédiée encore une fois par l'ouverture vers d'autres disciplines, par l'intégration notamment de la psychologie cognitive et des sciences du design, mais aussi des sciences sociales dans les processus de conception, développement et évaluation. Le *user-centered design* (conception orientée utilisateur) complète désormais le génie logiciel.

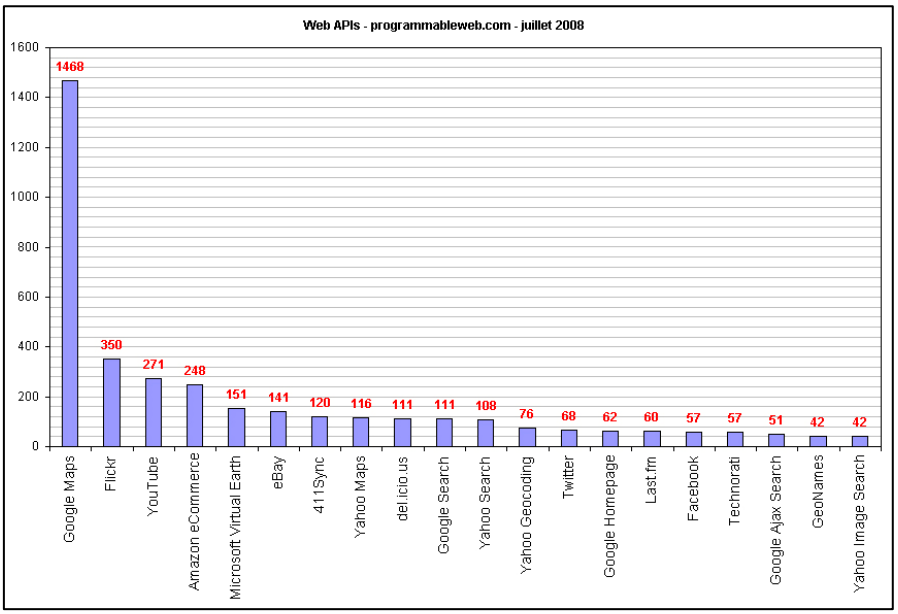
Avec l'arrivé d'Internet dans les années 1990, la problématique de la conception et création d'applications continue à se complexifier, principalement à cause de la diversification des usages qui accompagne l'imbrication de l'informatique dans les mailles fines de la vie quotidienne de toujours plus de personnes. Dans le cadre des applications en ligne il est très difficile de connaître les usagers et leurs besoins, et les méthodes prometteuses comme le design participatif sont très coûteuses. Le résultat est souvent un processus heuristique peu planifié où un système est mis sur le marché avec un nombre limité de fonctionnalités qu'on enrichit au fur et à mesure en observant les réactions des usagers. On parle donc de « bêta perpétuelle » (*perpetual beta*) – *Flickr* utilise le terme « gamma » – pour designer des applications Web qui n'arrivent jamais à un état final.

Depuis quelque temps, une autre stratégie – similaire mais encore plus expérimentale – prend de l'importance : à la fin des années 1980, Eric von Hippel [1988] introduit le concept de « *user-driven innovation* » (innovation pilotée par les usagers) pour désigner une méthode de conception qui part de l'idée que les usagers savent mieux ce qu'ils veulent que n'importe quelle étude de marché pourrait le dégager. Il suffirait donc de leur donner les moyens d'être créatifs et incorporer les meilleures idées pour créer des produits novateurs. Chez von Hippel, l'usager devient une sorte de co-développeur non rémunéré dont la capacité à innover est supérieure à celle des entreprises souvent trop centrées sur leur fonctionnement interne.

L'intérêt des entreprises pour l'*open source* et leur disposition croissante à ouvrir leurs données et plateformes à des développeurs de *mashups* doivent être compris avec cet arrière plan : les caractéristiques du numérique ouvrent un espace de création inédit et les communautés qui peuplent cet espace représentent une immense ressource potentielle de créativité, de travail et de connaissances. L'API de *Google Maps* est un cas idéal pour démontrer et complexifier cet argument.

2 LA GOOGLE MAPS API ET SA COMMUNAUTE

Google Maps est un service de cartographie et de géolocalisation[4] en ligne, ouvert au public depuis février 2005. A peine quelques jours après son introduction, des programmeurs réussissent à « hacker » le service pour l'intégrer dans leurs propres applications Web. Contrairement aux réactions habituelles dans ce genre de cas, *Google* décide de ne pas poursuivre ces individus en justice mais, tout au contraire, d'ouvrir leur plateforme aux développeurs externes par le biais d'une interface de programmation ; la première version de cette API est rendue publique en juin 2005. Les conditions d'utilisations sont assez favorables : l'intégration du service dans un autre site est gratuite et illimitée pourvu que le site soit non payant pour les internautes – le placement de publicité n'est pourtant pas interdit.[5] Uniquement le service « *geocoder* » (accessible via l'API depuis juin 2006), qui permet de localiser une adresse ou un lieu sur une carte, est limité à 50K appels en 24 heures par application.[6] Les développeurs se jettent sur le système et *housingmaps.com* de Paul Rademacher, un site qui projette les annonces immobilières aspirées de *craigslist.org* sur une carte *Google Maps*, devient le prototype même du *mashup*. Début juillet 2008 le site *programmableweb.com* liste 1468 *mashups* utilisant l'API de *Google Maps* – de loin le plus grand nombre – mais nous estimons que le service est utilisé dans des dizaines de milliers de sites d'une façon ou d'une autre.



[4] La géolocalisation consiste à lier une information à des coordonnées dans l'espace, p.ex. une adresse, un lieu, des photos, etc.

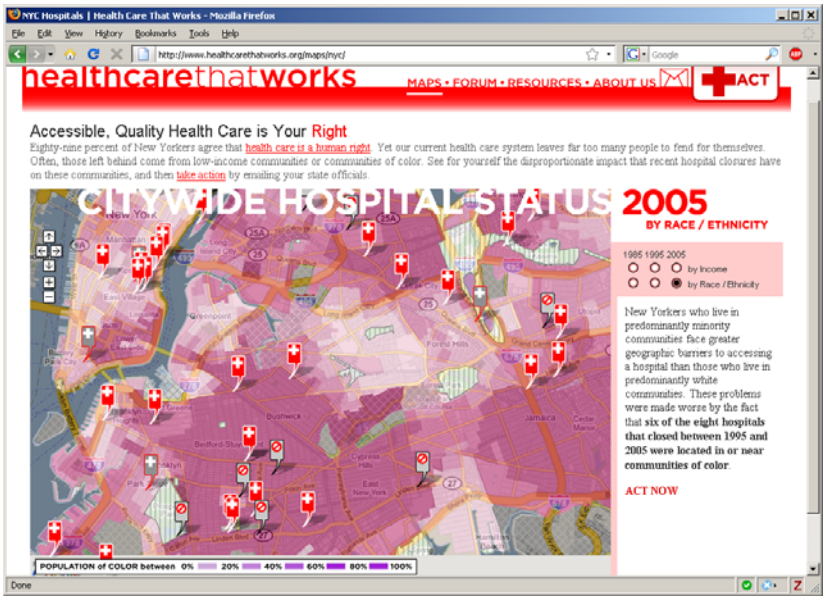
[5] Cf. http://code.google.com/apis/maps/terms.html

[6] Cette limite est désormais fixée à 15K par adresse IP ce qui revient effectivement à un élargissement du quota dans la plupart des cas.

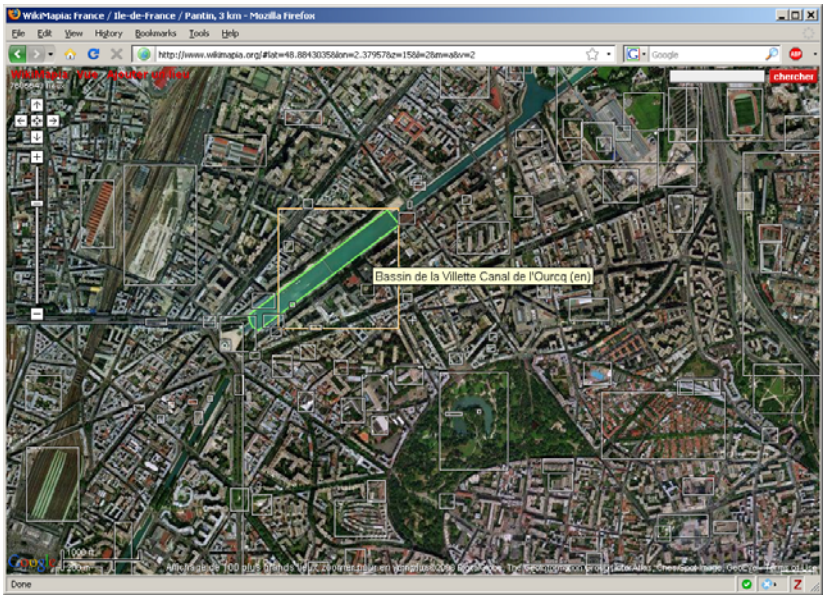
Autour de l'API s'est formée une importante communauté de développeurs qui utilisent le service dans leurs propres créations et participent activement dans l'évolution de l'outil même. Pour commencer à comprendre la logique hybride de cette culture participative, il faut spécifier en quel sens la communauté est *productive*. Cette productivité prend différentes formes mais peut être divisée en trois catégories : l'utilisation de l'API dans les *mashups*, l'évolution du service et l'élaboration de connaissances.

2.1 Les mashups

Le travail le plus visible de la communauté consiste évidemment en l'intégration du service *Google Maps* dans d'autres systèmes à travers l'API. Cela peut aller de l'insertion d'une simple carte sur la page « contact » d'un site jusqu'à la création de systèmes très élaborés. Le domaine de la géolocalisation est en pleine expansion et les expériences autour de *Google Maps* constituent un vecteur d'innovation important. L'API permet aux développeurs de lier des systèmes d'information intégrants une composante spatiale à un système de cartographie interactive pour un coût très faible. On voit l'émergence de principalement deux types d'applications. La première catégorie regroupe les systèmes de *visualisation* qui utilisent l'API pour projeter des informations sur une carte. Avec *Google Maps* cette pratique n'est plus réservée à des professionnels du domaine géographique et on peut aujourd'hui témoigner de l'émergence d'une véritable *rhétorique* de la carte. Le site *healthcarethatworks.org* par exemple réussit ainsi à démontrer de manière intuitive que la fermeture d'hôpitaux à New York ces dernières années touchait essentiellement des quartiers habités majoritairement par des personnes de couleur.

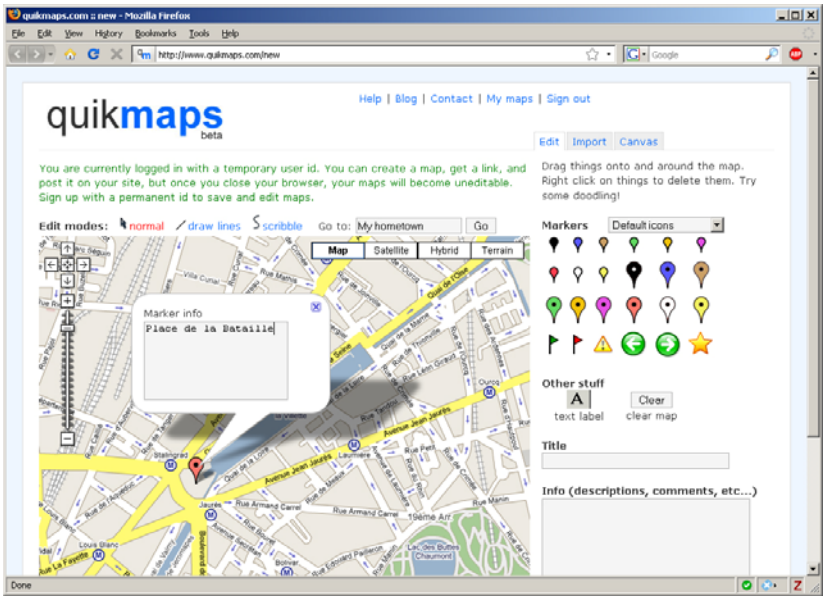


La deuxième catégorie est composée d'applications Web qui utilisent l'API pour *recueillir* (et puis présenter) des informations sur l'espace géographique. La carte fonctionne donc comme une interface visuelle pour lier une donnée à des coordonnées spatiales. Le site *wikipapia.org* par exemple contient 7,6 millions de lieux, intégralement apportés par les utilisateurs du site.



2.2 L'évolution du service

A côté des *applications* proprement dites du service, la communauté est très engagée quand il s'agit de rendre le système lui-même plus performant et plus facile à intégrer. On peut encore distinguer deux niveaux d'implication. Le premier concerne les outils qui permettent à des non-développeurs de créer des cartes interactives pour les utiliser sur leurs sites ou qui facilitent la tâche des développeurs expérimentés. Des sites comme *quickmaps.com* par exemple proposent une interface simple pour créer et annoter une carte ; des bibliothèques de code comme le « *clusterer* » de Jef Poskanzer – un moteur pour afficher des milliers de marqueurs sur une carte – rendent certaines tâches de développement plus abordables.



Le deuxième niveau concerne l'évolution de l'API elle-même. Dans ce domaine, la stratégie de *Google* revient en quelque sorte à une application de certaines règles du développement *open source* avancées par Eric S. Raymond [1998] dans son article canonique *The Cathedral and the Bazaar*. Trois « thèses » extraites de cet article devraient illustrer ce constat.

Thèse 6 : *Traiter vos utilisateurs en tant que co-développeurs est le chemin le moins semé d'embûches vers une amélioration rapide du code et un débogage efficace.* Dans le cadre de *Google Maps*, la complexité du service et la difficulté notoire d'accommoder les différents

navigateurs Web et leurs nombreuses versions rendent la chasse aux erreurs difficile et coûteuse. La communauté composée de milliers de développeurs alertes devient donc une ressource cruciale dans le débogage, non seulement de l'API mais du système entier. Or, ces volontaires ne trouvent pas seulement des erreurs, ils proposent souvent des solutions dans le *Google Maps API Group*[7], une newsgroup améliorée, qui fonctionne comme relais principal entre *Google* et la communauté. Il s'agit donc d'une application du principe de « *crowdsourcing* », la sous-traitance aux internautes.

Thèse 7 : *Distribuez tôt. Mettez à jour souvent. Et soyez à l'écoute de vos clients.* Le système et son API sont constamment améliorés et les mises à jour se succèdent rapidement : entre février 2006, l'introduction de la version 2.0 de l'API, et fin juin 2008, le journal des modifications (*changelog*) compte 85 versions rendues publiques – il s'agit bien de la logique de la *bêta perpétuelle*. Outre les corrections d'erreurs, les nouvelles versions ajoutent des modifications de la syntaxe et de nouvelles fonctions, généralement suite à une demande de la part des développeurs externes.

Thèse 11 : *Il est presque aussi important de savoir reconnaître les bonnes idées de vos utilisateurs que d'avoir de bonnes idées vous-même. C'est même préférable, parfois.* Les expériences, outils et extensions venant de la communauté servent souvent de modèle pour l'évolution de l'API et de *Google Maps* en général. Selon Pamela Fox, ingénieur de service chez *Google* et responsable des relations avec la communauté, les développeurs externes ont l'avantage de ne pas être contraints aux mêmes standards de qualité que les employés de l'entreprise. Ils peuvent donc expérimenter de manière beaucoup plus libre et tester ce « qui marche ou pas et ce qui peut être utile » [Fox, entretien par email]. Ici, le principe de l'innovation « pilotée par les usagers » (*user-driven innovation*) trouve donc une application littérale parce que la communauté fonctionne comme une sorte de *laboratoire*.

2.3 L'élaboration de connaissances

Ce dernier élément pointe déjà vers un autre aspect du travail de la communauté, celui de la connaissance. Introduite de manière quelque peu précipitée, l'API était très mal documentée au départ et c'était essentiellement les volontaires qui s'occupaient des différents guides, tutoriaux, exemples commentés et répertoires d'applications. Le *Google Maps Mapki*[8] par exemple joue encore aujourd'hui le rôle d'une documentation supplémentaire qui remplit les lacunes et omissions des documents officiels. Le centre du « laboratoire » communautaire est pourtant le *Google Maps API Group*, ouvert le jour de la présentation de l'API, qui fonctionne comme lieu d'échange principal, comptant plus de 31K membres et 125K messages début juillet 2008. Un développeur qui pose une question au groupe peut compter sur une réponse compétente en moins d'une heure et, par le biais de la fonction de recherche, s'ouvre une archive de connaissances redoutable. Il s'agit finalement d'une « communauté de pratique » [Lave, Wenger 1991] dont l'intelligence collective produit et stabilise des connaissances et *best practices*. La possibilité d'intégrer le rang des membres actifs ouvre ainsi un double chemin de socialisation et d'apprentissage.

3 ENTRE COMMUNAUTE ET COMMERCE

Ce travail fourni par la communauté soulève un ensemble de questions qui nous semblent au centre de la problématique plus générale de la « culture participative » et de la « production

[7] http://groups.google.com/group/Google-Maps-API/

[8] http://mapki.com

amateur ». Nous allons donc rapidement examiner de plus près les caractéristiques et motivations de la communauté, ses relations avec *Google* et finalement la question du pouvoir telle qu'elle se pose au sein de cette configuration complexe.

3.1 Une communauté ouverte jusque où ?

Qui sont les personnes qui participent à ces différentes activités et pourquoi investissent-elles leurs temps et savoir-faire sans rémunération ? A travers d'entretiens et d'une lecture des différents forums d'échange on peut retrouver des motivations similaires à celles avancées au sujet des développeurs *open source* [Weber 2004] : le plaisir de faire partie d'une communauté intéressée, la stimulation intellectuelle et créative de la programmation, la volonté de montrer et d'améliorer ses connaissances dans un domaine émergent et l'intérêt de se bâtir une réputation comme programmeur capable. Il faut prendre en compte qu'une partie importante des individus qui participent dans le travail communautaire autour de l'API de *Google Maps* sont des développeurs Web professionnels ou souhaitent le devenir. L'engagement volontaire est pour eux une façon d'exercer leurs compétences dans un domaine qui ne leur a pas été imposé, tout en étant économiquement prometteur. S'ajoute à cela une fascination très prononcée pour le domaine de la géolocalisation qui est vue comme un nouveau continent dont les possibilités ne sont pas encore identifiées. Les normes de comportements qui règlent les interactions entre les membres du groupe ressemblent donc fortement à celles que l'on trouve dans le contexte de l'*open source* : le ton est parfois dur mais toujours respectueux, les débats restent proches du sujet et le meilleur argument technique gagne.

Il faudrait pourtant préciser que le taux de participation des différents membres peut varier considérablement. Le fait que 27% des 125K messages dans l'*API Group* aient été écrits par seulement dix individus démontre qu'un nombre limité de personnes est responsable d'une grande partie du travail communautaire mentionné plus haut. Au fil du temps on peut également remarquer une sorte d'« ossification » de la communauté. Avec l'accumulation de connaissances et l'évolution du service vers toujours plus de complexité il devient de plus en plus dur pour un nouveau participant de rejoindre le « noyau dur » de la communauté couronné par le titre « *Maps API Guru* » derrière le pseudonyme ; l'émergence d'une oligarchie risque pourtant de freiner la croissance de la communauté et de nuire à son statut comme interlocuteur de *Google*.

Il est certain que la plupart des membres actif préféraient travailler dans un contexte *open source* mais le cas de la cartographie en ligne montre très nettement les limites de cette approche. Certes, la programmation d'un système de cartographie peut être fait sans problème par des volontaires et il existe effectivement des systèmes comparables à *Google Maps* entièrement libres, mais ces systèmes n'ont pas la possibilité d'accéder légalement aux cartes et images satellites fournies par des entreprises comme *Tele Atlas* ou *Navteq* qui ont investi des milliards d'euros dans la création de ces contenus. Google paye des sommes considérables pour le droit de les utiliser. Sans ces subventions, l'espace de création ouvert par l'API ne pourrait pas exister. Au moment où le monde matériel « lourd » entre en jeu, l'économie « légère » de l'*open source* rencontre ses limites.

3.2 Une configuration symbiotique ?

Afin de mieux comprendre les relations entre *Google* et la communauté de développeurs il est utile d'examiner plus précisément comment chaque partenaire tire profit de l'autre. Pour la communauté, l'avantage principal est effectivement la mise à disposition quasiment libre du service lui-même (et notamment des cartes et images satellites), mais aussi de l'infrastructure informatique très performante qui héberge *Google Maps*. Cette mise à disposition ouvre un champ de *production potentielle* qui, malgré les points d'interrogation dont nous allons parler

plus bas, est un espace largement ouvert à l'expérimentation et même à l'exploitation commerciale. Du côté de *Google*, une telle exploitation est explicitement souhaitée : un développeur qui conçoit un produit autour de *Google Maps* et qui souhaite l'exploiter commercialement peut le faire principalement en plaçant de la publicité sur son site ; si ce développeur est déjà bien familier des produits de l'entreprise, le système de publicité choisi sera très probablement *AdSense*, principale source de revenus du géant californien. En ouvrant son service et en soutenant les développeurs, *Google* arrive à *socialiser* les développeurs, à les fidéliser à ses produits. Cette stratégie semble rencontrer un fort succès, notamment lorsqu'on prend en compte que *Yahoo* et *Microsoft* proposent des services similaires et parfois plus performants que *Google Maps* sans avoir réussit à créer une diffusion comparable. « Virtual Earth de MSN est assez cool et je pense que la qualité de leurs images satellites et BEAUCOUP mieux, mais je trouve la communauté de développeurs et le support de l'API plutôt médiocre » dit l'utilisateur Eric[9] dans une discussion à propos des différents services de géolocalisation disponibles, et ce commentaire montre bien l'importance et le succès de la politique communautaire de *Google*.

Nous avons déjà décrit tout le travail que fournit la communauté au niveau des techniques et connaissances, mais *Google* en profite encore à d'autres niveaux : les développeurs qui s'engagent autour de l'API ne constituent non seulement une ressource externe mais potentiellement aussi une ressource interne. Paul Rademacher par exemple, le créateur de *housingmaps.com*, travaille désormais à Mountain View et n'est pas le seul employé ainsi recruté. Comme le note Steven Weber [2004], l'engagement volontaire est un moyen formidable pour un bon programmeur de se faire remarquer parce que les entreprises peuvent directement juger son travail, ce qui dans le domaine informatique est bien plus significatif qu'un diplôme et démontre l'enthousiasme du candidat.

Les relations entre *Google* et la communauté semblent donc assez fusionnelles et la lecture des discussions dans l'*API Group* témoigne effectivement d'une coexistence plutôt harmonieuse. Cela est loin d'être évident lorsqu'on prend en compte la dimension de la distribution de pouvoir.

3.3 Propriété et pouvoir

Sur ce point, la notion de *propriété* nous semble centrale parce qu'elle marque un départ important par rapport à la culture du tout ouvert du mouvement *open source*, que Steven Weber [2004] définit comme une expérience sociale regroupée autour d'une conception alternative de ce que veut dire être propriétaire de quelque chose. Parce que légalement, *Google* reste le détenteur de tous les droits de *Google Maps* et l'entreprise peut contrôler l'ensemble des conditions d'usage du service ; en principe, rien ne les empêche de le rendre payant, d'ignorer les souhaits de la communauté ou tout simplement de l'arrêter. Nous sommes donc très loin de l'esprit des licences *open source* comme la GPL (*GNU General Public License*) qui, selon Weber, redéfinissent la propriété intellectuelle comme *droit de distribuer* et non comme *droit d'exclure*.

En même temps, le pouvoir ne se réduit pas à sa seule dimension légale, il émane d'« une situation stratégique complexe » [Foucault 1976]. En permettant l'émergence d'une communauté d'« amateurs professionnels » autour de l'un de ses produits, une entreprise prend des risques : selon Jeff Bezos, CEO d'*Amazon.com*, « le bouche à oreille est très fort sur Internet et si vous rendez un consommateur content il peut le dire à 5000 autres personnes.

[9] Billet dans le *Google Maps API Group* du 25 octobre 2005.

Et si vous le rendez mécontent il va certainement le dire à 5000 autres personnes. »[10] Une communauté peut être une ressource formidable mais maltraitée, elle peut se transformer en un ennemi redoutable [cf. Jenkins 2006]. Les développeurs de *Google Maps* peuvent non seulement passer tout simplement à un service concurrent, mais une communauté en colère peut également facilement nuire à l'image d'une entreprise dont la devise est toujours « ne soit pas méchant ! » Le processus de socialisation va donc en quelque sorte dans les deux sens : les développeurs sont socialisés dans la culture *Google* mais *Google* est contraint par des normes communautaires, qui s'inspirent largement de la culture *open source*. Le fonctionnement final est le résultat de processus de négociation permanents. Par conséquent, l'ouverture vers les développeurs externes peut donc réduire la marge de manœuvre de l'entreprise. Il n'est donc guère étonnant que *Google* agisse de manière très prudente. Aucun des abus de l'API – l'extraction des cartes est notamment assez fréquente – n'a été réglé devant un tribunal, et les employés de l'entreprise portent beaucoup d'attention à ne pas paraître sourds ou arrogants dans leurs échanges avec la communauté. Tout changement doit être justifié et argumenté techniquement ; et, bien que les conditions d'utilisations réservent explicitement le droit à *Google* de placer de la publicité directement sur les cartes et images satellites, cela n'a pas été fait jusqu'ici. Finalement, une partie croissante de l'API est placée sous une licence *open source* – ce qui ne change pas grande chose mais constitue un acte symbolique important.

Il reste cependant d'autres lieux de frictions potentielles. En s'inspirant des *mashups* et extensions pour l'évolution du site *Google Maps* grand public, ces créations sont parfois rendues obsolètes. Avec l'introduction de *My Maps* (personnaliser et sauvegarder des cartes), *Map Maker* (créer et annoter des parcours) et la possibilité d'afficher des photos et articles de Wikipédia, *Google* se met en concurrence directe avec ses développeurs externes. Pour l'instant cela n'a pas encore produit de conflits très visibles mais le problème est bien réel et est un indicateur de l'asymétrie qui caractérise, malgré tout, les rapports entre l'entreprise et la communauté.

4 CONCLUSIONS

Le cas de *Google Maps* n'est pas un *pars pro toto* de l'ensemble de cette « culture participative » en train d'émerger, mais nous pensons qu'il est finalement plus représentatif que celui du mouvement *open source*, dont la redéfinition de la notion de propriété reste une île – certes en croissance – dans un océan peuplé d'hybrides qui combinent logique de marché et logique de communauté, et où la production amateur ne donne pas toujours lieu à la création d'un « bien commun » (*common*). Nous aurions donc tort de regarder la situation actuelle dans une logique binaire *ouvert* / *fermé*. En regardant de plus près, nous voyons que la production amateur brouille les frontières en créant des mélanges dynamiques et instables. Ces formes posent un certain nombre de questions nouvelles parce qu'à cause des intérêts différents les relations entre entreprises et communautés sont forcément ambiguës. Mais même dans une situation comme celle de *Google Maps* où l'acteur commercial détient tout les droits légaux, nous sommes loin des consommateurs passifs de Horkheimer et Adorno. La participation des usagers infiltre la logique traditionnelle de la production culturelle et affecte la marge de manœuvre de entreprises. L'exercice du pouvoir doit être proprement *foucaldien* : productif au lieu de répressif, micro plutôt que macro, subtil et flexible. Les normes communautaires désavouent l'usage de force brute et contraignent l'exercice de la loi comme outil de pouvoir.

[10] Interview avec *Business Week* du 16 mars 1999.

Pour le moment, la tâche pour la recherche ne consiste pas à porter un jugement moral sur une éventuelle culture participative mais à examiner les différentes formes qu'une telle culture peut prendre. Et c'est en dégageant les pratiques et jeux de pouvoir que nous découvrons un nouvel espace de création dont les chances et dérives restent en mouvement.

BIBLIOGRAPHIE

BARLOW John Perry

1996, *A Declaration of the Independence of Cyberspace*, http://homes.eff.org/~barlow/Declaration-Final.html, traduction française : http://www.freescape.eu.org/eclat/1partie/Barlow/barlowtxt.html

BRUNS Axel

2008, *Blogs, Wikipedia, Second Life, and Beyond: From Production to Produsage*, Peter Lang

JENKINS Henry

2006, *Convergence Culture : Where Old and New Media Collide*, NY University Press

LAVE J., WENGER E.

1991, *Situated Learning : Legitimate Peripheral Participation*, Cambridge University Press

FOUCAULT Michel

1976, *La volonte de savoir*, Gallimard

HORKHEIMER M., ADORNO T. W.

1947, *Dialektik der Aufklärung*, Querido

LEADBEATER C., MILLER P.

2004, *The Pro-Am Revolution : How Enthusiasts Are Changing Our Society and Economy*, Demos

RAYMOND Eric S.

1998, *The Cathedral and the Bazaar*, First Monday, Vol. 3, No. 3, traduction française : http://seddisoft.kelio.org/cathedrale-bazar.htm

RIEDER B., SCHÄFER M. T.

2008, *Beyond Engineering: Software Design as Bridge over the Culture/Technology Dichotomy*, in VERMAAS P. E., KROES P., LIGHT A., MOORE S. A., *Philosophy and Design. From Engineering to Architecture*, Springer

SHIRKY Clay

2008, *Here Comes Everybody : The Power of Organizing Without Organizations*, Allen Lane

TOFFLER Alvin

1980, *The Third Wave*, Bantam Books

TURING Alan M.

1948, *Intelligent Machinery*, National Physical Laboratory Report

VON HIPPEL Eric

1988, *The Sources of Innovation*, Oxford University Press

WEBER Steven

2004, *The Success of Open Source*, Harvard University Press