

## **Qu'est-ce que le Web 2.0 ?**

---

**Tim O'Reilly**  
**(version française)**

- 1- Le web en tant que plate-forme*
- 2- Tirer parti de l'intelligence collective*
- 3- La puissance est dans les données*
- 4- La fin des cycles de release*
- 5- Des modèles de programmation légers*
- 6- Le logiciel se libère du PC*
- 7- Enrichir les interfaces utilisateur*

## **Qu'est ce que le web 2.0 : modèles de conception et d'affaires pour la prochaine génération de logiciels.**

L'explosion de la bulle internet en 2001 a définitivement marqué un tournant dans l'histoire du web. Beaucoup de gens ont à ce moment considéré que le web était une technologie surévaluée alors qu'au contraire, le fait qu'une bulle se forme puis éclate est un trait commun à toutes les révolutions industrielles. Ces soubresauts sont même caractéristiques du moment où une technologie ascendante est prête à entrer dans une nouvelle phase. C'est en effet le moment où les simples prétendants arrivent à bout de souffle tandis que les points forts des premiers gagnants apparaissent : et c'est seulement à cet instant que l'on commence à comprendre ce qui distingue les premiers des seconds.

Le concept de web 2.0 est apparu avec une conférence « brainstorming » entre O'Reilly et Medialive International. Dale Dougherty, pionner du web et membre d'O'Reilly notait alors que bien loin de s'être effondré, le web n'avait jamais été aussi important et que nombre de nouveaux sites et applications à caractère innovant apparaissaient avec une régularité déconcertante. De plus, les quelques sociétés qui avaient survécu à l'hécatombe semblaient avoir quelque chose de commun. Se pouvait-il que le crack des « dot com » eût révélé une nouvelle ère pour le web au point que l'expression « web 2.0 » ait un sens ? Nous tombèrent d'accord sur le fait que cela en avait un : la conférence web 2.0 était née

En un an et demi (*ndt : l'article est daté du 30/09/2005*), le terme « web 2.0 » s'est franchement popularisé avec plus de 9,5 millions de citations dans Google. Mais il reste encore de multiples points de désaccord sur sa signification exacte, un certain nombre de personnes décrivant ce qui ne leur semble être qu'un buzzword bien marketé, d'autres y voyant une nouveau modèle de pensée.

Cet article est donc une tentative de clarification du sens du terme « web 2.0 ». Dans notre brainstorming initial, nous avons exprimé notre ressenti du web 2.0 par l'exemple :

### **Web 1.0 // Web 2.0**

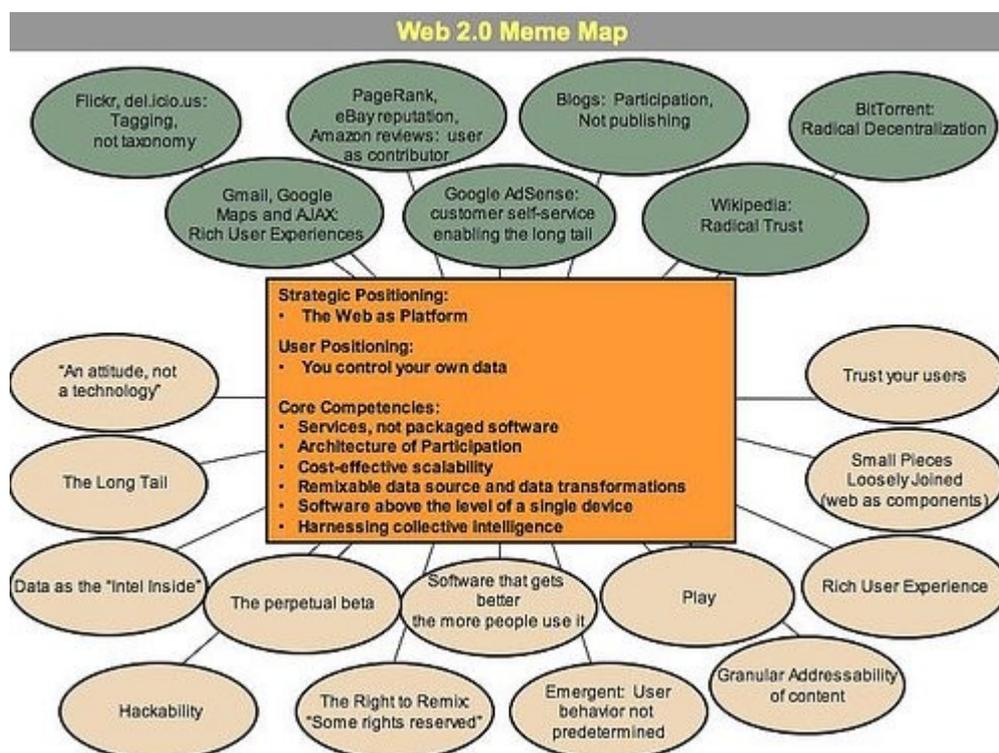
DoubleClick --> Google AdSense  
Ofoto --> Flickr  
Akamai --> BitTorrent  
mp3.com --> Napster  
Britannica Online --> Wikipedia  
sites perso --> blogs  
evite --> upcoming.org et EVDB  
spéculation sur les noms de domaines --> optimisation pour moteurs de recherche  
pages vues --> coût au clic  
« screen scraping » --> services web  
publication --> participation  
système de gestion de contenu --> wikis  
arborescence (taxonomie) --> tags ("folksonomy")  
rigidité du contenu --> syndication de contenu

Et la liste pourrait encore s'allonger... Mais cela ne nous dit toujours ce qui fait qu'une application ou un concept est web 1.0 et un autre web 2.0. (La question est particulièrement pressante dans la mesure où le terme « web 2.0 » s'est à ce point répandu que des entreprises se l'approprient bien souvent sans réelle compréhension de sa signification. C'est d'autant plus délicat que la plupart de ces start-ups friandes de termes à la mode ne sont en rien web 2.0 et que parallèlement à cela des applications que nous avons identifiées comme étant web 2.0, comme Napster ou BitTorrent ne sont même pas des applications web à proprement parler !). Nous avons donc commencé à tirer de ces

exemples les principes qui sont d'une manière ou d'une autre à la base des succès du web 1.0 mais aussi des applications récentes les plus remarquables.

## 1 - Le web en tant que plate-forme

Comme de nombreux concepts majeurs, le web 2.0 n'a pas de frontière claire mais plutôt un centre de gravité. Vous pouvez voir le web 2.0 comme un ensemble de principes et de pratiques qui, à la manière d'un système planétaire, verrait des sites exposant tout ou partie de ses préceptes graviter à des distances variables du centre en question.



Ce schéma montre une cartographie du web 2.0 telle qu'elle a été développée lors d'une session brainstorming pendant le FOO Camp, une conférence O'Reilly Media. C'est encore à l'état de brouillon mais cela montre la plupart des idées émanant du noyau du web 2.0.

Lors de la première conférence web 2.0 en octobre 2004, John Battelle et moi-même avons établi une liste préliminaire de principes pour notre discours d'ouverture. Le premier de ces principes était « le web en tant que plate-forme ». Or il se trouve que c'était déjà la devise d'un enfant chéri du web 1.0, Netscape, qui avait malgré cela succombé à sa fameuse bataille contre Microsoft.

De plus, deux de nos exemples web 1.0 de départ, DoubleClick et Akamai, étaient des pionniers qui avaient eux aussi abordé le web comme une plate-forme. Les gens n'y pensent pas souvent en tant que « services web » mais en réalité, les serveurs de publicité furent probablement les premiers « mashup » (*ndt : dans le sens ou ce sont des contenus diffusés directement au sein de sites clients*)

largement déployés sur la toile.

Chaque bannière publicitaire est en effet une forme de coopération entre 2 sites web dans un seul but : délivrer une page intégrée à un lecteur sur ordinateur de plus. De son côté, Akamai traite aussi le réseau en tant que plate-forme quand il élabore un cache ainsi qu'un système de distribution de contenu soulageant la bande-passante d'un réseau souvent congestionné.

Cependant ces pionniers apportent des contrastes intéressants avec des acteurs du web plus récents dans la mesure où ces derniers ont repris à leurs aînés des solutions aux mêmes problématiques tout en les poussant plus loin, et qu'ils expliquent ainsi de manière plus profonde la véritable nature de cette nouvelle plate-forme. DoubleClick et Akamai furent donc des pionniers du web 2.0, mais nous constatons déjà qu'il est possible d'exploiter plus efficacement ce nouveau concept à l'aide d'autres « design patterns » du web 2.0.

Je vous propose d'examiner plus précisément les trois cas qui suivent et qui nous permettront d'extraire les fameux traits différenciant le web 1.0 du web 2.0.

### **Netscape vs. Google**

Si Netscape est emblématique du web 1.0, Google est de manière encore plus certaine son équivalent web 2.0, ne serait-ce du fait de leurs entrées sur le marché boursiers qui marquèrent chacune l'aube d'une nouvelle ère du web. Commençons donc la comparaison de ces deux sociétés et de leur positionnement.

Netscape énonçait « le web en tant que plate-forme » dans les termes du paradigme du logiciel d'autrefois : leur produit-phare était le navigateur web, une application cliente, et leur stratégie était d'utiliser leur domination sur le marché du navigateur pour créer un marché à destination des serveurs haut de gamme. Le contrôle des standards d'affichage de contenu et des applications utilisant un navigateur aurait du en théorie donner à Netscape le même genre de pouvoir sur ce marché que celui que possède Microsoft sur celui des Pcs. Un peu à la manière de l'expression "horseless carriage" (*ndt* : « la charrette sans cheval ») qui popularisa l'automobile, Netscape promut l'expression « webtop » pour supplanter le desktop et prévit de remplir ce webtop d'informations et d'applications grâce à des apporteurs de contenus qui utiliseraient des serveurs Netscape.

Finalement, serveurs et navigateurs web devinrent de simples outils et la plus grande part de la valeur ajoutée du web se concentra dans les services diffusés par les plate-forme web.

Google au contraire, commença son existence en tant qu'application web native, jamais vendue ou packagée mais délivrée en tant que service, avec des clients payant, directement ou indirectement, pour utiliser ce service. Aucun des pièges de la vieille industrie logicielle ne pouvait s'appliquer à son modèle. Aucun planning de sortie de différentes version, juste une amélioration continue. Aucun système de vente ou de licence, simplement des utilisateurs. Aucun problème de portage sur différentes plate-forme de sorte que les clients puissent faire marcher le logiciel sur leur machine, uniquement une quantité massive de Pcs utilisant un système d'exploitation open source ainsi que quelques applications maison (que quiconque d'extérieur à l'entreprise n'a jamais pu voir...).

A la base, Google nécessitait une compétence dont Netscape n'avait jamais eu besoin : la gestion de base de données. Google n'est pas une simple suite d'outils logiciels, c'est une base de données spécialisée. Sans données, les outils ne servent à rien ; sans logiciel, les données sont ingérables. La gestion des licences et le contrôle des API – points cruciaux de l'ère précédente – n'avaient plus lieu d'être dans la mesure où les logiciels n'avaient plus besoin d'être distribués mais seulement utilisés

et où sans la capacité de collecter et de gérer des données, le logiciel n'est que de peu d'utilité. En fait, *la valeur d'un logiciel est proportionnelle à l'échelle et au dynamisme des données que cela permet de gérer.*

Le service offert par Google n'est ni un serveur – bien qu'il soit délivré par une quantité massive de serveurs web – ni un navigateur – bien que pour l'utiliser, un navigateur soit nécessaire. Leur service de recherche bien connu n'héberge même pas le contenu qu'ils permettent à leurs utilisateurs de trouver. A la façon d'un coup de téléphone où la valeur n'est pas dans les téléphones mais le réseau les mettant en relation, Google place sa valeur dans l'espace situé entre le navigateur et le serveur de contenu, comme un opérateur entre l'utilisateur et son usage du web.

Même si Netscape et Google peuvent être décrites comme des sociétés de logiciels, il est clair que Netscape appartient à la même catégorie que les Lotus, Microsoft, Oracle, SAP, et autres qui ont pris leur envol au début de la révolution logicielle du début des années 80 alors que les comparses de Google sont d'autres applications web telles que eBay, Amazon, Napster, et bien entendu, DoubleClick et Akamai.

### **DoubleClick vs. Overture et AdSense**

Comme Google, DoubleClick est un pur produit de l'ère internet. Il décline le modèle du logiciel en tant que service, a un coeur de métier dans la manipulation des données et, comme nous l'avons vu plus haut, a été un pionnier du service web bien avant même que le terme « web service » ait été créé. Cependant, DoubleClick fut finalement limité par son modèle économique. La société paria en effet sur la croyance très « 90's » que le web était fait de publications et non de participations ; que les annonceurs, et non les consommateurs, devaient être les acteurs moteur ; que la taille importait, et qu'Internet serait peu à peu dominé par les quelques sites webs du haut des classements de MediaMetrix et ses semblables.

Dans cette logique, DoubleClick annonce fièrement sur son site « plus de 2000 implémentations réussies » de sa solution. Pendant ce temps, Yahoo! Search Marketing (auparavant Overture) et Google AdSense possèdent déjà des centaines de milliers d'annonceurs...

Le succès de Google et d'Overture provient de leur compréhension de ce que Chris Anderson nomme « la longue traîne » (*ndt : traduction de l'expression « the long tail » déjà popularisée dans la communauté francophone*) : la force collective des petits sites représente l'essentiel du contenu du web. L'offre de DoubleClick demande un contrat de vente très formel, limitant leur marché aux quelques milliers de sites les plus importants. Overture et Google ont montré comment il était possible d'ajouter virtuellement de la publicité sur n'importe quelle page existante. De plus, ils ont remplacé les publicités tape-à-l'oeil et agaçantes telles que les bannières et les « pop-up » par des messages peu intrusifs, sensibles au contexte de la page à laquelle ils s'intègrent et finalement plus appréciés des consommateurs.

*Leçon du web 2.0 : mettre au point un service simple d'accès et une gestion algorithmique des données pour toucher l'intégralité du web, jusque dans sa périphérie, pas seulement son centre, jusqu'au bout de sa longue traîne, pas seulement en son coeur.*

De manière peu surprenante, les autres « success stories » du web 2.0 démontrent le caractère gagnant de ce modèle. Ebay permet les transactions occasionnelles de quelques dollars entre individus en tant qu'intermédiaire. Napster (bien que fermé pour des raisons légales) ne s'est pas construit grâce à un immense catalogue de chansons mais à travers la mise au point d'un système faisant de chaque client un serveur renforçant lui-même le réseau.

## Akamai vs. BitTorrent

Tout comme DoubleClick, Akamai est optimisé pour faire des affaires avec un cœur de cible pas avec la périphérie du marché. Bien que ses services bénéficient à tous en facilitant l'accès aux sites à fortes audiences, ses revenus ne proviennent que d'un certain nombre de grands comptes du web.

BitTorrent, comme les autres pionniers du mouvement peer-to-peer, utilise une approche radicalement différente de la décongestion des flux. Chaque client est aussi un serveur, les fichiers sont découpés en fragments légers pouvant être téléchargés de plusieurs lieux à la fois, permettant ainsi à chacun des membres du réseau d'apporter un peu de bande passante et de données à chacun des autres utilisateurs : plus un fichier est populaire, plus rapide est son téléchargement.

BitTorrent illustre là un principe clé du web 2.0 : *le service s'améliore automatiquement quand le nombre de ses utilisateurs croît*. Alors qu'Akamai doit ajouter des serveurs pour améliorer son service, chaque utilisateur de BitTorrent apporte un peu de ses ressources à l'ensemble de la communauté. C'est implicitement une « architecture de participation », une nouvelle éthique de la coopération dans laquelle le service agit comme un intermédiaire intelligent, connectant chaque parcelle de la gigantesque banlieue du web à une autre et donnant le pouvoir aux utilisateurs eux-mêmes.

## 2 - Tirer parti de l'intelligence collective

Le principe central du succès des géants nés du web 1.0 qui ont survécu pour maintenant mener l'ère du web 2.0, semble être qu'ils ont su exploiter de manière efficace la force de l'intelligence collective que recèle potentiellement le web :

Les liens hypertextes sont le fondement du web. Au fur et à mesure que les utilisateurs ajoutent des contenus, celui-ci est intégré à la structure du web par d'autres utilisateurs qui les découvrent et placent des liens vers ceux-ci. Telles des synapses formant un cerveau où les associations se renforcent à force de répétitions et d'intensité, les connections au sein du web se multiplient organiquement à la mesure que leur dicte l'activité de l'ensemble des utilisateurs. Yahoo !, la première grande success story, est né d'un catalogue de liens, une agrégation du meilleur de milliers puis de millions d'internautes. Même si Yahoo! a depuis quelque peu changé de modèle d'affaire, son rôle en tant que portail vers l'oeuvre collective des utilisateurs du web reste sa principale valeur.

La percée de Google dans la recherche, qui en a fait rapidement et indiscutablement le maître du marché, était basée sur le PageRank (*ndt : littéralement classement de page*), une méthode basée sur la structure hypertexte du web plutôt que sur l'analyse des contenus.

Le produit d'eBay est l'activité collective de l'ensemble de ses utilisateurs ; comme le web lui-même, eBay grossit organiquement en réponse à l'activité de ses clients, le rôle de la société consistant simplement à fournir le contexte dans lequel cette activité peut se dérouler. De plus, les avantages comparatifs d'eBay viennent entièrement de la masse critique d'acheteurs et de vendeurs disponibles, tout nouveau concurrent étant inévitablement pénalisé de ce point de vue.

Amazon vend les mêmes produits que des concurrents tels que Barnesandnoble.com, ils reçoivent les mêmes descriptions de produits, images de couverture et contenus de la part des éditeurs. Mais Amazon a fait une science de l'implication du consommateur. Ils sont un cran au-dessus des autres en ce qui concerne les commentaires des acheteurs, les invitations à participer (de plusieurs

manières et sur quasiment toutes les pages) – et plus important encore, ils se servent de l'activité des utilisateurs pour affiner les résultats de leurs recherches. Alors que les recherches de Barnesandnoble.com poussent les utilisateurs vers ses propres produits ou vers des résultats sponsorisés, Amazon renvoie toujours vers « les plus populaires », une notion calculée en temps réel non pas à partir des ventes mais à partir de ce que les gens d'Amazon appellent le « flow » (*ndt : le bruit, le mouvement, le flux*) autour des produits. Il n'y a donc rien d'étonnant à voir la croissance des ventes d'Amazon dépasser celles de ses concurrents.

Ces temps derniers, plusieurs sites ont marqué le web de leur empreinte avec de tels principes et les poussent même peut-être un peu plus loin :

- Wikipedia, une encyclopédie en ligne basée sur le principe peu commun qu'une entrée puisse être ajoutée par n'importe quel utilisateur du web et modifiée par un autre est une expérimentation radicale du dicton de Eric Raymond (qui se référait à la base au développement open-source) qu'« avec suffisamment d'yeux, tous les bugs disparaissent » dans le domaine de la création de contenu.
- Les sites tels que del.icio.us et Flickr, 2 sociétés ayant bénéficié dernièrement d'une grande attention, ont été pionnières dans le concept que certains appellent « folksonomy » (*ndt : qu'on pourrait traduire par « classement par les gens » pour contraster avec « taxonomy », « classement standard »*), une sorte de catégorisation collaborative des sites utilisant des mots clés librement choisis, qu'on appelle en général « tag » (*ndt : trace, marque*). Le « tagging » permet un phénomène d'associations multiples et redondantes semblables à celles que le cerveau utilise lui-même plutôt que des catégories rigides. Exemple typique : une photo de Flickr d'un chiot peut être taguée par les termes « chiot » et « mignon » - permettant donc de retrouver la photo suivant un cheminement naturel grâce, une fois de plus, à l'activité des utilisateurs.
- Les systèmes anti-spam collaboratifs tels que Cloudmark agrègent les décisions individuelles des utilisateurs d'e-mail sur ce qui est et n'est pas du spam, et obtiennent de bien meilleurs résultats que les systèmes s'appuyant sur l'analyse des messages eux-mêmes.
- C'est enfoncer une porte ouverte que de dire que les plus grands succès du web ne font pas de publicité sur leurs produits. Leur adoption est guidée par le « marketing viral », les recommandations se propageant d'utilisateur unique à utilisateur unique. Vous pouvez parier sur le fait que si un site ou un produit se repose sur la publicité pour faire parler de lui, il n'est pas web 2.0.
- Une grande partie de l'infrastructure du web elle-même – Linux, Apache, MySQL et du code Perl, PHP ou Python sont utilisés sur la plupart des serveurs web – repose sur les méthodes de « production individuelle » (*ndt : par opposition à la production de masse de firmes privées*) de l'open-source qui sont, elles aussi, une émanation de l'intelligence collective.
- Il existe plus de 100.000 projets open-source listés sur SourceForge.net. N'importe qui peut y ajouter un projet, télécharger et utiliser le code, et les nouveaux projets migrent de la périphérie vers le centre à mesure que les utilisateurs les font vivre, contribuent, téléchargent... le tout reposant évidemment sur le marketing viral.

*La leçon à retenir : Dans l'univers web 2.0, l'implication des utilisateurs dans le réseau est le facteur-clé pour la suprématie du marché*

## Les blogs et la sagesse des foules

Un des traits les plus remarquables de l'ère Web 2,0 est la montée du phénomène « blog ». Les pages personnelles étaient là depuis les débuts du web, les journaux intimes et le courrier des lecteurs depuis longtemps encore... de quoi s'agit-il alors ?

On pourrait dire très simplement que le blog est une page personnelle dans un format journal intime. Mais comme le souligne Rich Skrenta, l'organisation chronologique d'un blog « semble être une différence anodine, alors que cela conduit à un processus de création, à un système de signalement et à une chaîne de valeur radicalement transformés ».

Une des choses qui ont fait la différence est une technologie appelée RSS. RSS est l'avancée la plus significative dans l'architecture du web depuis que les premiers bricoleurs d'internet ont réalisé que les CGI pouvaient être utilisés pour créer des sites reposant sur des bases de données. RSS permet à quelqu'un de ne pas seulement attacher une page, mais de s'y abonner, avec un avertissement à chaque fois que le contenu de la page change. Skrenta appelle cela « le web incrémenté ». D'autres l'appellent « live web » (*ndt* : « *le web en temps réel* »).

Evidemment, « les sites web dynamiques » (c'est à dire générant dynamiquement un contenu à partir de base de données) ont désormais remplacé les pages web statiques depuis près de 10 ans. Mais ce qui est dynamique dans le « live web », ce ne sont pas seulement les contenus, ce sont les liens eux-mêmes. Un lien vers un blog pointe vers une page changeant périodiquement, avec des « permaliens » pour des entrées spécifiques, et un avertissement pour chaque changement. Un lien RSS est donc bien plus puissant que le favori ou le simple lien vers une autre page web.

RSS signifie aussi que les navigateurs web ne servent plus seulement à voir des pages. Certains agrégateurs de contenu, tels que Bloglines, sont basés sur le web, mais d'autres sont des applications résidentes et d'autres encore permettent à leurs utilisateurs de s'abonner à des flux RSS sur des dispositifs portables.

RSS n'est désormais plus seulement utilisé pour afficher les actualités des blogs, mais aussi pour toute sorte de données régulièrement mises à jour : cours de la bourse, météo, disponibilité de photos. Cette utilisation revient à l'une des origines du phénomène : RSS est né en 1997 de la rencontre de la « Really Simple Syndication » (*ndt* : *la syndication vraiment simple*) de Dave Winer, utilisée pour signaler les mises à jour de blogs, et de la « Rich Site Summary » (*ndt* : *sommaire de site riche*) de Netscape qui permettait à ses utilisateurs de créer des pages personnelles avec des flux de données régulièrement mis à jour. Netscape perdit de son intérêt pour cette technologie qui fut reprise par le pionnier du blog Userland, la société de Winer. Nous percevons cependant encore dans les dernières version de RSS l'héritage de ces deux parents.

Mais RSS est seulement une partie de ce qui rend le blog si différent d'une page web ordinaire. Citons la remarque de Tom Coates sur la signification des permaliens : « Cela peut sembler n'être aujourd'hui qu'un élément anodin, mais c'était en réalité ce qui a fait que les weblogs, jusqu'alors un simple moyen de publier facilement du contenu, sont devenus cette extraordinaire fatras de communautés entremêlées. Pour la première fois, il est devenu assez simple de pointer vers un article très spécifique et d'en débattre. Des discussions se sont créées. Des sessions de chat se sont déroulées. Et, naturellement, les amitiés se sont nouées et renforcées. Le permalien fut la première – et la plus réussie – des tentatives de bâtir des ponts entre les blogs. »

De bien des manières, la combinaison de RSS et des permaliens ajouta à HTTP, le protocole du web, des caractéristiques de NNTP, le protocole de Usenet. La blogobulle peut être vue comme une nouvelle voie de communication de particulier à particulier équivalente à celle qu'offre Usenet

depuis les débuts d'internet. Les blogueurs ne se contentent pas d'échanger des liens, ils peuvent aussi via le mécanisme des trackbacks (suivi des traces) voir qui les a liés à leur page et répondre soit par un lien réciproque soit par des commentaires.

Il est très intéressant de noter que les liens à double-sens furent un des buts des premiers systèmes hypertextes comme Xanadu. Les puristes de l'hypertexte ont salué l'arrivée des trackbacks comme une étape décisive sur le chemin de l'hyperlien à double-sens. Mais il faut noter que ces trackbacks ne sont pas à proprement parler à deux sens, ils s'avèrent plutôt être (potentiellement) des liens unidirectionnels symétriques créant un effet bidirectionnel. La différence paraît subtile mais en pratique, elle est considérable. Les réseaux sociaux tels que Friendster, Orkut ou encore LinkedIn, qui requièrent un accord de la personne sollicitée pour établir une connexion n'ont pas la souplesse du web. Comme le note Caterina Fake, cofondatrice du système de partage de photos Flickr, l'attention n'est réciproque que de manière accidentelle (Flickr permet donc de visualiser des listes, n'importe qui pouvant s'abonner au flux RSS de photos d'un autre utilisateur. L'accès est notifié mais n'a pas à être approuvé).

Si pour une part essentielle, le web 2.0 est une affaire d'intelligence collective (faire du web une sorte de cerveau global), la blogobulle est l'équivalente du dialogue mental permanent qui réside dans nos pensées superficielles, la voix que nous entendons tous en nous. Cela n'est certes pas nécessairement le reflet de la structure profonde de notre esprit qui est en grande partie faite d'inconscient, mais plutôt celui de notre pensée consciente. Et en tant que reflet de la pensée et de l'attention, la blogosphère a commencé à produire un effet des plus puissants...

Premièrement, parce que les moteurs de recherche utilisent les liens pour proposer des pages pertinentes et que les blogueurs, la population la plus prolifique en liens, ont désormais un rôle presque disproportionné dans les résultats des recherches. Deuxièmement, parce que la communauté des blogueurs se réfère souvent à elle-même, les blogs renvoyant vers d'autres blogs et renforçant ainsi leur visibilité. La « caisse de résonance » du web tant décriée est donc aussi un amplificateur.

Si ils n'étaient qu'un amplificateur, les blogs seraient toutefois assez inintéressants. Mais à la manière de Wikipédia, ils exploitent eux aussi l'intelligence collective pour mieux filtrer le contenu. C'est là que ce que James Suriowecki appelle la « sagesse des foules » entre en jeu : tout comme le PageRank de Google produit des résultats meilleurs qu'une analyse individuelle des documents, l'attention collective de la blogosphère sait retenir des informations plus pertinentes que n'importe quel autre filtre.

Alors que les grands médias se contentent de voir chaque blog comme un concurrent, ils tardent à faire l'amer constat que la compétition se déroule pour eux contre la blogobulle dans son ensemble. Il ne s'agit pas juste d'une guerre entre des sites mais entre des modèles économiques. Le monde du web 2.0 est aussi le monde de ce que Dan Gillmor désigne par l'expression « nous, les médias », un monde dans lequel ceux qui n'étaient jusque là qu'auditeurs reprennent à quelques personnes réunies dans une arrière-salle le pouvoir de choisir ce qui est important ou non.

### **3 - La puissance est dans les données**

Toutes les applications web d'importance sont liées à une base de données spécialisée : la base d'exploration Google, l'index de Yahoo, la base de produits Amazon, la base d'utilisateurs d'eBay, la base de cartes de MapQuest, la base de chansons disponibles de Napster. Comme me le signalait Hal Varian l'année dernière, « le SQL est le nouveau HTML ». La gestion de base de données est le cœur de métier des sociétés du web 2.0, à tel point qu'on donne parfois à leurs applications le nom

d' « infoware » plutôt que software (*ndt* : « infogiciel » plutôt que logiciel).

Ces faits m'amènent à ce qui est pour moi la question centrale : qui possède les données ?

Dans l'univers d'internet, on a déjà vu un certain nombre de cas dans lesquels le contrôle des données amène dans un premier temps la domination du marché puis le profit. Le monopole sur l'enregistrement des noms de domaine qu'un décret gouvernemental (*ndt* : *il s'agit bien sûr du gouvernement des Etats-Unis*) avait offert à Network Solutions (plus tard repris par Verisign) fut par exemple l'une des premières sources de revenus de l'Internet. Puisque nous avons vu que l'avantage stratégique du contrôle des API n'avait plus vraiment de sens sur le web, cela signifie que l'élément de domination des marchés se trouve dans les données. C'est d'autant plus vrai lorsqu'elles sont difficiles à créer et susceptibles d'être entretenues par un réseau d'utilisateurs.

Regardez les licences d'utilisation de la base de données sur chaque carte servie par MapQuest, , maps.yahoo.com, maps.msn.com, ou maps.google.com, et vous verrez "Maps copyright NavTeq, TeleAtlas," ou avec le nouveau satellite d'imagerie, "Images copyright Digital Globe." . Ces sociétés ont fait des investissements considérables dans leurs données (NavTeq dit investir chaque année 750 millions de dollars dans leur base d'adresses et de directions. Digital Globe a dépensé 500 millions de dollars pour lancer son propre satellite et améliorer l'imagerie basée sur les satellites gouvernementaux). NavTeq est allé si loin pour imiter Intel que cela se retrouve jusque dans leur logo : les voitures embarquant des systèmes de navigations portent en effet l'inscription « NavTeq Onboard » (*ndt* : *à comparer avec le fameux « Intel Inside »*). Les données sont bel et bien le nouvel « Intel Inside », le composant essentiel de systèmes dont l'infrastructure est largement open source ou tout du moins collaborative.

La situation d'Amazon est cependant plus contrastée. Tout comme ses concurrents (BarnesandNoble.com par exemple), sa base de données d'origine provient du fournisseur de registre ISBN RR Bowker. Mais à la différence de MapQuest, Amazon améliore sans cesse ses données, en ajoutant du contenu provenant des éditeurs par exemple. Plus important encore, ils encouragent les utilisateurs à évaluer les données, de telle sorte qu'après 10 ans, c'est Amazon et non plus Bowker, qui est devenu la référence en matière ressources bibliographiques pour les universitaires et les libraires. Amazon a aussi introduit son propre système d'identifiant, le « ASIN », qui correspond à l'ISBN lorsqu'il est présent et crée un code équivalent lorsqu'il n'existe pas. En un mot, Amazon s'est véritablement doté d'une politique de gestion des données mais aussi des fournisseurs de celles-ci.

Imaginez que MapQuest ait fait la même chose : amener les utilisateurs à évaluer les cartes, ajouter d'autres niveaux d'informations. Il aurait certainement beaucoup plus difficile pour ses adversaires d'entrer sur le marché simplement en achetant une licence à un fournisseur de données.

L'arrivée récente de Google Maps fait de ce marché un véritable laboratoire de la compétition entre la vente d'applications et la vente de données. Le modèle de service léger de Google conduit à la création de nombreux services sous la forme de « mashups » reliant Google Maps avec d'autres services de la toile. Housingmaps.com de Paul Rademacher, qui combine Google Maps avec les données de Craigslist pour créer un outil de recherche de logement interactif est un excellent exemple de ces possibilités.

Evidemment, ces mashups ne sont pour le moment que des expérimentations mais les premiers projets professionnels devraient bientôt suivre. Malgré tout, on peut déjà dire que pour une partie des développeurs, Google a pris la place de Navteq en tant que fournisseur de données en se plaçant en tant qu'intermédiaire privilégié. On peut s'attendre à voir les premières batailles entre

fournisseurs de données et fournisseurs de logiciels éclater dans les années qui viennent, quand tous auront compris l'importance des données dans l'univers web 2.0.

La course pour la possession de données stratégiques a déjà commencé : positionnement, identités, calendriers d'événements, identifiants de produits... Dans bien des cas, là où il y a un coût pour construire une base de données, il y a l'opportunité de créer un support pour des services à valeur ajoutée avec une source unique de données. Souvent, le gagnant sera la société qui atteindra la première une masse critique de données par agrégation des utilisateurs et convertira cet avantage en services.

Dans le domaine de l'identité par exemple, Paypal, 1-click d'Amazon et les millions d'utilisateurs des systèmes de communication peuvent tous sembler légitimes pour construire une base de données d'identités à l'échelle du réseau (dans cette optique, la récente tentative de Google d'utiliser les numéros de téléphones en tant qu'identifiants dans les comptes Gmail est peut-être une étape vers l'intégration des téléphones). Pendant ce temps, des start-up comme Sxip explorent le potentiel d'identités fédérés, dans la quête d'une sorte de « 1-click distribué » qu'apporterait un sous-système du web 2.0 dédié à l'identité. Dans le domaine des calendriers, EVDB est une tentative de construire le plus grand calendrier partagé du monde à partir d'un système de participation de type wiki. Bien que le juge de paix reste le succès d'une approche ou d'une start-up particulière dans le domaine, il est clair que les standards et les solutions dans ces domaines qui sauront transformer des données brutes en des sous-systèmes de confiance du « système d'exploitation internet » rendront possibles une nouvelle génération d'applications.

Un autre point d'importance doit être précisé au sujet des données : les aspects de confidentialité et de droit des utilisateurs sur leurs données. Dans la plupart des premières applications web, le copyright n'était appliqué que de manière très approximative. Amazon, par exemple, dit respecter les droits de auteurs des critiques faites sur son site mais en l'absence d'application concrète, n'importe qui peut recopier une critique et la reposer ailleurs. Cependant les sociétés ayant pris conscience que le contrôle des données est un avantage stratégique décisif, on peut s'attendre de leur part à une réaction sur le sujet.

Comme la montée du logiciel propriétaire a conduit au mouvement du logiciel libre, il est envisageable de voir le mouvement « des données libres » s'opposer peu à peu à l'univers des données propriétaires. On peut en voir les premiers signes dans des projets ouverts comme Wikipedia, la licence Creative Commons ou encore dans des projets tels que GreaseMonkey qui permet à l'utilisateur de s'approprier un peu plus les données envoyées par les pages web en contrôlant l'affichage.

#### **4 - La fin des cycles de release**

Comme cela a été dit dans la discussion opposant Netscape à Google, une caractéristique définissant l'ère internet du logiciel est que celui-ci est proposé en tant que service et non en tant que produit. Ce fait amène nombre de changements fondamentaux dans le modèle d'affaire des sociétés de logiciels :

1- les traitements deviennent le coeur de métier. A l'expertise de Google ou de Yahoo ! Dans le développement dans leurs produits, doit correspondre une expertise dans les opérations du

quotidien. Le passage d'un logiciel-produit à un logiciel-service est à ce point fondamental que le logiciel ne peut plus fonctionner si il n'est pas maintenu à une échelle quotidienne. Google doit continuellement parcourir le web et mettre à jour ses indices, filtrer le « spam link » et autres tentatives d'influencer ses résultats, continuellement et dynamiquement répondre à des centaines de millions de requêtes asynchrones d'utilisateurs, tout en leur faisant correspondre des publicités adaptés à leur contexte.

Ce n'est donc pas par hasard que le système d'administration de Google, ses techniques de mise en réseau, d'équilibrage des charges sont des secrets peut-être mieux gardés encore que ses algorithmes. La réussite de Google dans l'automatisation de ces processus est probablement un facteur décisif dans leur avantage sur leurs concurrents en matière de coûts.

Ce n'est pas non plus par hasard que des langages de script tels que Perl, Python, PHP et maintenant Ruby jouent un rôle aussi importants dans les entreprises du web 2.0. Perl fut décrit de manière célèbre par Hassan Schroeder, le premier webmaster de Sun, comme le « robinet du web ». Les langages dynamiques (souvent appelés langages de script et méprisés par les ingénieurs de l'ère logiciel-produit) sont un outil de choix aussi bien pour les administrateurs réseaux et systèmes que pour les développeurs d'application élaborant des systèmes dynamiques en perpétuel changement.

2- Les utilisateurs doivent être traités comme des co-développeurs, en référence aux pratiques de l'open-source (même si le logiciel dont il est question ne doit pas forcément être open source). Le dicton de l'open-source « livrer tôt et livrer souvent » se transforme en une position encore plus radicale, « la bêta perpétuelle », dans laquelle le produit est développé avec de nouvelles fonctionnalités apparaissant de manière mensuelle, hebdomadaire voire quotidienne. Il n'y a donc rien d'étonnant à voir des services tels que Gmail, Google Maps, Flickr, del.icio.us, et autres porter un logo « bêta » depuis déjà un bon moment.

L'observation en temps réel du comportement de l'utilisateur dans le but de voir quelles sont les nouvelles fonctionnalités utilisées, devient donc un nouvelle compétence centrale. Un développeur web d'un grand service en ligne dit à ce sujet : « Nous mettons deux ou trois nouvelles fonctionnalités quelque part sur le site chaque jour, si elles ne sont pas utilisées, nous les retirons. Si elles sont appréciées, nous les implémentons partout ailleurs sur le site. »

Cal Henderson, le principal développeur de Flickr, a révélé récemment qu'il déployait une nouvelle version (*ndt : plus exactement un nouveau « build »*) de son application toute les demis-heures. C'est là un modèle de développement radicalement différent ! Même si toutes les applications web ne vont pas aussi loin que Flickr, la plupart a un cycle de développement tout à fait différent de l'ère du PC et du client-serveur. C'est pour cette raison qu'un récent éditorial de Zdnet concluait que Microsoft ne parviendrait pas à l'emporter face à Google :

« Le business model de Microsoft dépend de la remise à niveau que chacun fait sur son PC tous les deux ou trois ans. Celui de Google ne dépend que de la capacité des utilisateurs à se servir des nouvelles fonctionnalités qui leur sont proposées. »

Alors que Microsoft a démontré une capacité incroyable à apprendre de et finalement surpasser la concurrence, il n'est plus ici question de temps. La concurrence demandera cette fois à Microsoft (et par extension à toutes les sociétés de logiciel existantes) de devenir une société complètement différente. Les sociétés web 2.0 natives bénéficient d'un avantage naturel, celui de ne pas avoir de vieux schémas (avec les modèles d'affaires et les revenus correspondants) derrière lesquels s'abriter.

## 5 - Des modèles de programmation légers

Après que le principe des services web se fut popularisé, de grandes compagnies se sont lancés dans l'aventure en proposant des ensembles complexes de services conçus pour créer de véritables environnements de développement pour applications distribuées.

Mais tout comme le web avait réussi en passant outre une grande partie de la théorie de l'hypertexte, préférant le pragmatisme à une conception idéale, RSS devint peut-être le service web le plus déployé du fait de sa simplicité, alors que la complexité des web services des grandes sociétés condamna ceux-ci à un déploiement limité.

De la même manière, les services web d'Amazon furent fournis sous 2 formes : la première adhérant au formalisme de SOAP (Simple Object Access Protocol), la seconde se contentant de fournir un flux de données XML via HTTP, un modèle plus simple désormais connu sous le terme de REST (Representational State Transfer). Alors que les connexions B2B (comme celles reliant Amazon et ToysRUs par exemple) utilise SOAP, Amazon rapporte que 95% des usages de ses web services se font via REST.

C'est la même quête de simplicité qui peut être vue dans d'autres services web « organiques ». La récente sortie de Google Maps en constitue un bon exemple. L'interface de Google Maps bâtie sur les principes d'AJAX a rapidement été décryptée par quelques hackers, qui ont alors utilisé ces données pour bâtir de nouveaux services.

Les données relatives à la cartographie était déjà disponibles via quelques fournisseurs tels qu'ESRI, MapQuest ou encore Microsoft MapPoint. Mais Google Maps a bouleversé cet univers du fait de sa simplicité. Alors que les expérimentations d'utilisation de web services demandait jusque là un contrat entre les intéressés, le fait de laisser les données aisément accessibles a permis à quelques bidouilleurs de réutiliser ces données de manière créative.

Il y a là plusieurs enseignements à en tirer :

1- *Mettre en place des modèles de programmation légers permettant la création de systèmes faiblement couplés.* La complexité des services web « corporate » est faite pour des systèmes fortement couplés. A contrario, la plupart des applications web les plus intéressantes demeure faiblement couplé, voire fragile. La façon de penser web 2.0 est très différente des visions classiques des systèmes d'informations !

2- *Pensez syndication, pas coordination.* Les services web simples tels que RSS ou ceux basés sur l'architecture REST cherchent avant tout à laisser des données à disposition, pas à contrôler ce qui se passe à l'autre bout de la connexion. Cette idée est fondamentale vis-à-vis du paradigme même d'Internet dans la mesure où c'est une transposition du principe du « end-to-end » (*ndt : principe fondamental de l'Internet Protocol stipulant que l'interprétation des données a lieu à chaque extrémité de la chaîne d'informations*).

3- *Une conception faite pour être « bidouillable » et « remaniable ».* Les systèmes tels que le web, RSS et AJAX ont un point commun : les barrières pour les réutiliser sont très faibles. La plupart des logiciels utiles sont open source, et même quand ils ne le sont pas, ils sont assez peu protégés du point de vue de la propriété intellectuelle. L'option « voir la source » des navigateurs a rendu possible à n'importe qui la possibilité de copier la page de quelqu'un d'autre ; RSS a été conçu pour renforcer les possibilités de l'utilisateur de voir le contenu qu'il désire quand il le désire et non selon le bon vouloir d'un fournisseur d'informations ; les services web les plus puissants sont ceux qui

permettent le plus facilement de les utiliser dans des buts pour lesquels ils n'ont pas été conçus. La phrase « certains droits réservés » qui a été popularisé par la licence Creative Commons pour s'opposer à la célèbre « tous droits réservés » est assez représentative de cet esprit.

### **L'innovation est dans l'assemblage**

Les modèles d'affaires « légers » ont une affinité naturelle avec les modèles de programmation et de connections léger. L'état d'esprit web 2.0 n'hésite pas à réutiliser l'existant : un nouveau service tel que housingmaps.com a été élaboré simplement en reliant 2 services existants. Housingmaps.com n'a pas (encore) de modèle d'affaires, mais plusieurs services à petite échelle, Google AdSense (ou peut-être Amazon ou même les deux) lui apporte l'équivalent d'un modèle de rémunération.

Ces exemples donnent un aperçu d'un autre principe du web 2.0, que nous appelons « innovation par l'assemblage ». Quand les composants de base deviennent abondants, il est possible de créer de la valeur en les assemblant de manière nouvelle ou plus efficace. Tout comme la révolution du PC a apporté de nombreuses opportunités d'innovation dans l'assemblage du hardware et a permis à des sociétés comme Dell faisant une science de cet assemblage de vaincre des sociétés dont le modèle d'affaires reposait sur l'innovation dans la production de composants, nous pensons que le Web 2.0 apportera des opportunités pour des sociétés d'entrer dans la compétition simplement en intégrant et en assemblant des services fournis par d'autres.

## **6 - Le logiciel se libère du PC**

Une autre des caractéristiques du web 2.0 qui mérite d'être mentionnée est le fait qu'il n'est plus limité à la plate-forme PC. Le dernier conseil que le développeur Dave Stutz donna à Microsoft fut : « Les logiciels utiles qui se libéreront d'une plate-forme spécifique seront des vecteurs de fortes marges pour un bon moment ».

Bien entendu, n'importe quelle application web peut être vue comme un logiciel indépendant d'une plate-forme spécifique. Après tout, même la plus simple des applications web implique au moins deux ordinateurs : l'un hébergeant le serveur web, l'autre le navigateur. Or comme nous l'avons déjà expliqué, le développement du web en tant que plate-forme pousse cette idée jusqu'à des applications synthétisant des services apportés par de nombreux ordinateurs.

Là enore, comme de nombreuses parties du web 2.0, l'aspect « 2.0 » n'est pas dans la nouveauté, mais plutôt dans la pleine réalisation du véritable potentiel de la plate-forme web, celle-ci devant nous guider pour comprendre comment concevoir applications et services.

A ce jour, iTunes est le meilleur exemple de ce principe. Cette application va sans cesse d'un appareil portable à un système web massif, le PC servant uniquement de mémoire locale et de station de contrôle. Il y avait déjà eu plusieurs tentatives de mettre du contenu venu du web dans des dispositifs portables, mais le couple iPod/iTunes est la première conçue pour être véritablement multi-plate-forme. TiVo est un autre bon exemple de ce phénomène.

Itunes et TiVo offrent également la démonstration de quelques autres principes du web 2.0. Elles ne sont pas des applications web en elles-mêmes, mais elle tire parti de la puissance de la plate-forme web, en faisant une partie permanente et pratiquement invisible de leurs infrastructures. La gestion de données est très clairement le coeur de leur offre. Ce sont des services, pas des applications packagées (bien que dans le cas d'iTunes, cela puisse être utilisé en tant qu'application packagée

pour manipuler des données locales). De plus, TiVo et iTunes montrent une utilisation naissante de l'intelligence collective, même si dans chacun des cas, leurs expérimentations sont entrées en guerre avec les lobbies de la propriété intellectuelle. La seule limite se trouve dans l'architecture de participation d'iTunes, bien que la récente apparition du podcasting change quelque peu la donne de ce point de vue.

Il s'agit de la partie du web 2.0 dont nous attendons les changements les plus importants, dans la mesure où de plus en plus d'appareils pourront être connectés à la nouvelle plate-forme que constitue le web. Quelles applications deviendront possibles quand nos téléphones et nos voitures ne se contenteront plus d'utiliser des données mais en émettront ? Le suivi du trafic routier en temps réel, les « flash mobs » (*ndt : un groupe de personnes qui ne se connaissent pas, se réunit pendant quelques minutes -grâce à un message SMS ou email- pour accomplir au même moment une action dénuée de sens, puis se disperse*) et le journalisme citoyen ne sont que quelques uns des signes précurseurs des possibilités du web de demain.

## 7 - Enrichir les interfaces utilisateur

Dès le navigateur « Viola » de Pei Wei en 1992, le web a été utilisé pour délivrer des « applets » et d'autres types de « contenus actifs » à l'intérieur du navigateur. L'introduction de Java en 1995 était aussi motivée par cet objectif. JavaScript puis le DHTML permirent d'apporter de manière plus légère intelligence côté client et richesse d'interface. Il y a de cela plusieurs années, Macromedia lança le terme « Rich Internet Applications » (qui a été repris par le projet Open Source Lazlo) pour mettre en lumière les possibilités de Flash vis à vis des applications (et plus seulement des contenus multimédias).

Cependant le potentiel de richesse d'interface du web ne fut jamais exploité par les grandes applications, jusqu'à ce que Google lance Gmail, rapidement suivi par Google Maps : des applications web possédant un niveau d'interactivité équivalent à un logiciel PC classique. La technologie utilisée par Google pour cela fut baptisée AJAX dans un article de Jesse James Garrett de l'entreprise de design web Adaptive Path. Il écrivait alors :

« Ajax n'est pas une technologie, il s'agit de plusieurs technologies, se développant chacune de leur côté, combinées ensemble pour donner des résultats aussi nouveaux que puissants. Ajax comporte :

- une présentation basée sur les standards XHTML et CSS
- un affichage dynamique et interactif grâce à DOM (Document Object Model)
- un système d'échange et de manipulation de données utilisant XML et XSLT
- un mécanisme de récupération de données asynchrone utilisant XMLHttpRequest

JavaScript pour lier le tout »

Ajax est aussi un élément clé des applications web 2.0 tels que Flickr (qui appartient désormais à Yahoo!), les applications de 37signal basecamp et backpack et bien entendu les applications de Google comme Gmail ou Orkut. Nous entrons dans une période sans précédent d'innovation dans l'interface à mesure que les développeurs deviennent capables de réaliser des applications web aussi riches que les applications locales classiques.

Il est intéressant de noter que bien des possibilités explorés aujourd'hui existaient depuis maintenant quelques années. A la fin des années 90, Microsoft et Netscape avaient une vision des possibilités qui sont désormais exploitées mais leur bataille au sujet des standards rendit difficile la création

d'applications multi-navigateurs. C'est seulement lorsque Microsoft eut gagné cette guerre que le seul navigateur restant devint de facto le standard et que ces applications devinrent possibles. Heureusement, lorsque Firefox relança la concurrence sur le marché des navigateurs, les blocages au sujet des standards ne se reproduisirent pas dans les mêmes proportions.

Nous nous attendons à voir apparaître de nombreuses applications web au cours des prochaines années, certaines seront entièrement nouvelles, d'autres seront des réimplémentations d'applications PC existantes. C'est d'autant plus inévitable que chaque changement de plate-forme crée aussi l'opportunité d'un changement de leadership pour les acteurs d'un marché logiciel donné.

Gmail a déjà apporté plusieurs innovations intéressantes dans les applications e-mail en combinant les forces du web (accessible de n'importe où, capacités de recherches avancées) avec une approche ergonomique proche des interfaces PC habituelles. Les applications clientes essaient donc de grignoter des parts de marché en ajoutant de nouvelles fonctionnalités telles que la messagerie instantanée. Sommes nous encore loin du client de communication complètement intégré incluant le meilleur de l'e-mail, de la messagerie instantanée et du téléphone portable tout en utilisant la voix sur IP pour ajouter encore plus de richesse à l'interface du nouveau client web ? En tout cas, le compte à rebours est lancé.

Il est assez facile d'imaginer comment le web 2.0 réimplémentera le carnet d'adresses. Un carnet d'adresses web 2.0 pourra d'une part traiter un ensemble de contacts mémorisés comme le fait par exemple un téléphone. Parallèlement à cela, un agent de synchronisation web, dans le style de Gmail, se souviendra de chaque message envoyé ou reçu, de chaque adresse e-mail ou numéro de téléphone utilisé et s'appuiera sur des heuristiques de réseaux sociaux pour décider quelles alternatives offrir en cas de recherches infructueuses. Si il ne trouve pas de réponse, le système élargira alors le périmètre du réseau social dans le lequel il effectue sa requête.

Le traitement de texte du web 2.0 permettra l'édition de données collaborative de type wiki et plus seulement celle de documents indépendants. Mais il pourra aussi offrir l'édition de document riche comme Microsoft Word ou OpenOffice savent le faire. Writely est un excellent exemple d'une telle application, bien que sa portée soit encore assez limitée.

Mais la révolutions web 2.0 ne s'arrêtera pas là. Salesforce.com a montré comment le web pouvait être utilisé pour délivrer le logiciel comme un service, à l'échelle d'applications d'entreprises telles que la CRM.

L'opportunité pour de nouveaux arrivants d'exploiter pleinement le potentiel du web 2.0 entrainera la création d'applications qui apprennent de leurs utilisateurs en s'appuyant sur une architecture de participation et qui se démarqueront non seulement par leur interface mais aussi par la richesse des données partagées qu'elles offriront.

### **Le coeur de métier des sociétés du web 2.0**

A travers les 7 principes évoqués plus haut, nous avons souligné quelques uns des principaux traits du web 2.0. Chaque exemple abordé démontre un ou plusieurs principes clé, mais laisse en général les autres de côté. Terminons donc en résumant ce que nous pensons être le coeur de métier des sociétés du web 2.0 :

**- des services, pas un package logiciel, avec des possibilités d'économie d'échelle**

- **un contrôle sur des sources de données uniques, difficiles à recréer, et dont la richesse s'accroît à mesure que les gens les utilisent**
  
- **considérer les utilisateurs comme des co-développeurs**
  
- **tirer partie de l'intelligence collective**
  
- **toucher le marché jusque dans sa périphérie à travers la mise en place de service « prêt à consommer »**
  
- **libérer le logiciel du seul PC**
  
- **offrir de la souplesse dans les interfaces utilisateurs, les modèles de développements ET les modèles d'affaires**

La prochaine fois qu'une société clame « ceci est web 2.0 », confrontez-la à la liste ci-dessus. Plus elle marque de points, plus elle est digne de cette appellation. Rappelez-vous néanmoins que l'excellence dans un domaine vaut mieux quelques paroles pour chacun des sept.