

Web 2.0, premier système

DÉVELOPPEMENT.

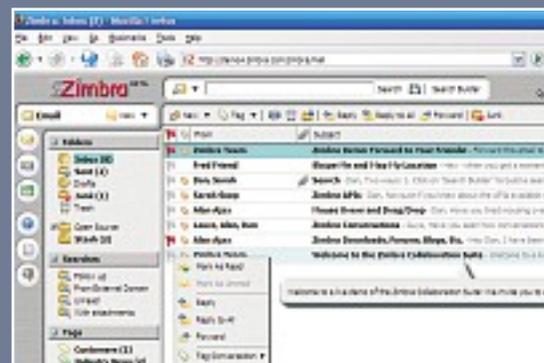
Depuis son navigateur, l'utilisateur accède désormais à des applications aussi riches et ergonomiques que celles installées en local. Une révolution est-elle en marche ?

On se croirait revenu en 1999 : le web est en pleine effervescence. Des start up se créent chaque jour, et les capitalistes sont à l'affût. On reparle même de révolution internet, la seconde celle du web version 2.0. Dans ce nouveau monde, l'internaute ne se contente plus de cliquer sur des liens hypertextes. Il organise, manipule et produit de l'information, exactement comme avec une application bureautique.

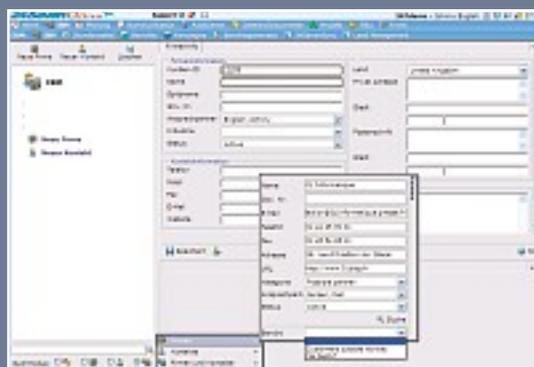
Ce fameux web 2.0 a déjà ses pionniers, ses héros et ses icônes. Il suffit de jeter un œil sur Zimbra, Flickr, Goowy ou 24Sevenoffice pour mesurer l'écart avec le web tel qu'on le connaît. Jusqu'ici, chaque clic effaçait la page et provoquait un aller-retour avec le serveur. Maintenant, l'affichage est instantané. On profite du glisser/déplacer, des menus déroulants, des raccourcis clavier, et même du clic droit.

L'ergonomie n'a rien à envier aux applications locales. Et pourtant, il s'agit toujours de pures applications web. Il n'y a rien à installer au préalable : ni plugin ni machine

On a du mal à croire qu'il s'agit d'applications web



ZIMBRA. Le clone d'Outlook. Complet, riche et rapide, Zimbra a poussé le raffinement jusqu'à inclure le clic droit et des infobulles. Un bien meilleur client de messagerie que nombre de logiciels locaux.



24 SEVENOFFICE. Il gère toute la relation client. Cette suite couvre le suivi des contacts jusqu'à la gestion de projet, en passant par le collaboratif. Ses performances pâtissent parfois du choix d'exécuter de nombreuses fonctions en local.

virtuelle. Tout fonctionne à l'identique quels que soient le navigateur et la plate-forme. Pour le directeur informatique, c'est le rêve. S'il pouvait appliquer ces techniques aux applications client-serveur, il réglerait d'un coup ses problèmes de déploiement. Les mises à jour seraient automatiques, sans procédure d'installation ni la nécessité de gérer des droits particuliers pour installer tel ou tel composant. Nul besoin non plus d'avoir des PC de compétition. Un navigateur suffit. Qu'il soit sous Linux, Mac OS ou Windows, l'application fonctionnera de la même façon. Aurait-on trouvé le chaînon manquant des applications web riches ?

Moins d'échanges avec le serveur

Avant de répondre, tâchons de comprendre comment font Zimbra et consorts pour

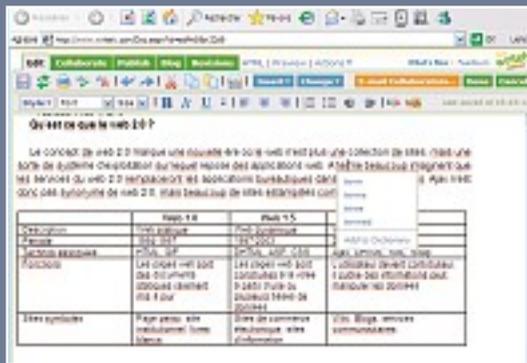
développer ces applications en ligne. Ils utilisent une technique simple, vieille de sept ans : le transfert asynchrone. Plutôt que de charger des pages entières à chaque clic, ils ne récupèrent en tâche de fond que les paquets de données nécessaires, décrits en XML (XMLHttpRequest). Ils s'appuient ensuite sur le langage Javascript du navigateur pour modifier tel ou tel élément de la page grâce au langage HTML dynamique. C'est aussi Javascript qui gère les menus déroulants, le glisser/déplacer ou l'affichage des boîtes de dialogue. Et comme XMLHttpRequest et Javascript existent dans tous les navigateurs modernes, l'application est, de facto, multi-plate-forme. En réalité, le développeur doit néanmoins composer avec les différences d'implémentation de Javascript.

La technologie a l'avantage de n'appartenir à personne. Comme Linux, elle ●●●

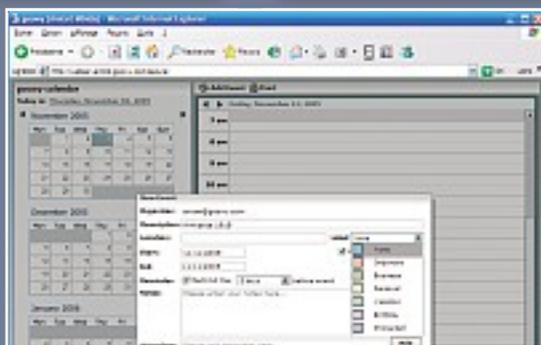
d'exploitation en ligne



MEEBO. Il réunit toutes les messageries instantanées. Elle se connecte aux réseaux ICQ/AIM, MSN, Yahoo, Jabber et Google Talk. Simple et légère, elle évite d'installer plusieurs clients de messagerie sur sa machine.



WRITELY. Un vrai traitement de texte qui couvre aussi bien les documents Word que HTML. Il gère les styles, les révisions, l'édition à plusieurs, les annulations multiples, corrige l'orthographe et enregistre au format Word.



GOOWY. Ce système d'exploitation en ligne propose un joli poste de travail avec une barre de tâches, un bureau, et toute une série d'applications. Une interface inspirée de Mac OS X.

Les avantages du client-serveur....

- Une interface riche.
- Il encaisse les traitements lourds.
- Ergonomique.
- Rapide.

.... sans les inconvénients

- Non lié à une plate-forme.
- Il ne requiert que peu de ressources machine.
- Pas compliqué à déployer.
- Peu cher à maintenir et à mettre à jour.

Qu'est-ce que web 2.0 ?

Le concept de web 2.0 marque une nouvelle ère. Le web n'est plus une collection de sites, mais une sorte de système d'exploitation pour applications web. A terme, les services de web 2.0 pourraient remplacer les applications bureautiques.

| | WEB 1.0 | WEB 1.5 | WEB 2.0 |
|------------------------|---|---|---|
| TYPE DE WEB | Statique. | Dynamique. | Collaboratif. |
| PÉRIODE | 1994-1997. | 1997-2003. | 2003. |
| TECHNOLOGIES ASSOCIÉES | HTML, GIF. | DHTML, ASP, CSS. | Ajax, DHTML, XML, Soap. |
| CARACTÉRISTIQUES | Les pages web sont des documents statiques rarement mis à jour. | Les pages web sont construites à la volée à partir d'une ou plusieurs bases de données. | L'utilisateur devient contributeur. Il publie des informations et manipule les données. |

Un exemple d'application web 2.0 : Google Maps

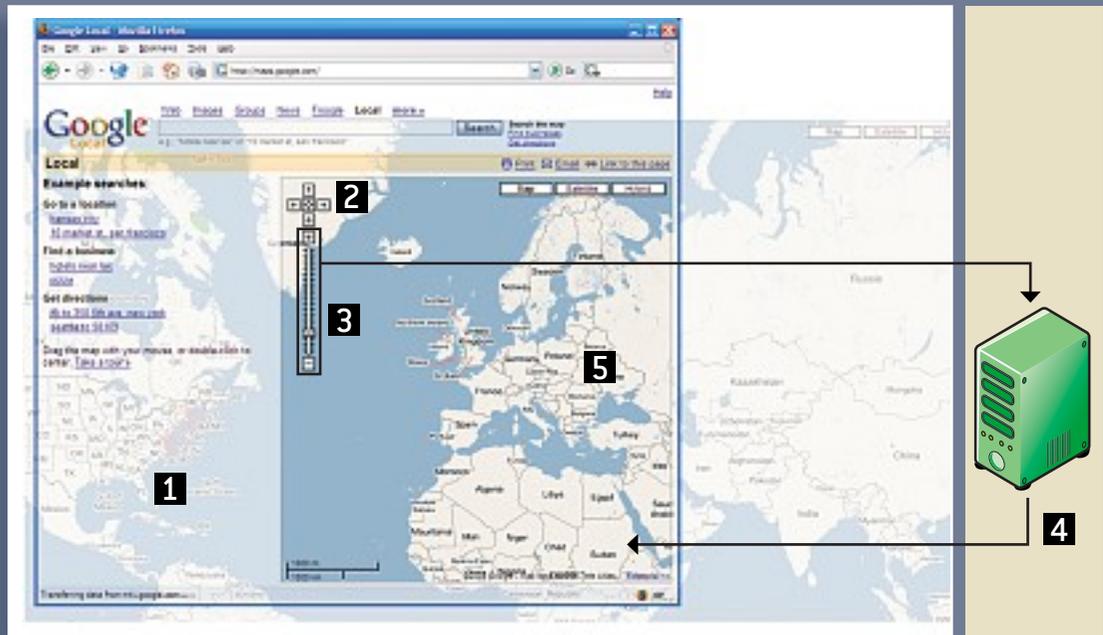
1 Le navigateur commence par charger une carte beaucoup plus grande que celle affichée, ainsi que le code Javascript qui servira à la manipuler.

2 Quand on clique sur les boutons pour déplacer la carte, aucune requête n'est envoyée au serveur. Le code Javascript déplace la grande carte stockée en mémoire en utilisant les feuilles de style (DHTML).

3 Quand l'internaute zoome sur la carte ou qu'il la déplace dans une zone limitrophe, le code Javascript envoie une requête au serveur via la fonction XMLHttpRequest pour recevoir les prochains morceaux de carte à afficher. Il invoque alors un service web sur le serveur.

4 Ce dernier renvoie les morceaux manquants dans un format JSON, une sorte de Javascript binaire. Il pourrait aussi les récupérer en XML, mais il faudrait alors les décoder. Ce qui prend du temps.

5 Les données reçues sont interprétées et affichées par Javascript dans la fenêtre du navigateur. Si l'on repart en arrière, l'affichage est immédiat, car les données sont déjà dans l'antémémoire.



●●● a évolué hors du sillage des fournisseurs. Outlook Web Access l'a inaugurée le premier en 1998, mais c'est Google qui l'a rendue célèbre avec Google Maps au début de l'année. Puis tout s'est accéléré au mois de mars, quand un consultant du cabinet Adaptive Path a formalisé l'ensemble et lui a donné le nom ronflant d'Ajax. Depuis, le terme s'est diffusé comme une traînée de poudre. Provoquant un battage sans précédent chez les développeurs.

Seulement voilà. Ajax est encore loin d'être mûr. Rappelons que ces techniques, quoique anciennes, ne sont formalisées que depuis quelques mois. Elles ont démontré leur intérêt pour de petites applications en ligne. Mais qu'en est-il pour les grosses applications d'entreprise, celles écrites en Powerbuilder, Visual Studio, Delphi et autres Windev ? Là, les avis sont partagés.

Il y a ceux qui n'y croient pas du tout. Leurs plus grands reproches concernent Javascript. Pour eux, ce n'est même pas un langage, mais un bricolage inventé pour effectuer quelques traitements locaux sur le navigateur. Il n'est pas orienté objet, ses varia-

bles ne sont pas typées, et il lui manque des notions de haut niveau. Résultat : le développeur peut écrire n'importe quoi, n'importe comment, car le langage offre une trop grande liberté.

Un calvaire pour les chefs de projet

Développer en Ajax peut aussi devenir très compliqué. Pour s'en convaincre, un coup d'œil sur le code de Google Maps suffit. Tout y est codé à la main : la transaction HTTP avec le serveur, la manipulation des données XML, les différences entre les navigateurs. Ici, pas de développement visuel. Les programmeurs débutants peuvent s'absentir. Pire : comme les outils d'aide au développement, au test, à la maintenance ou à la documentation manquent encore, les délais deviennent totalement flous. Difficile, dans ces conditions, d'industrialiser le développement. Bâtir des applications professionnelles dans un tel contexte est perçu au mieux comme une solution à courte vue, au pire comme une hérésie. Sans compter que

le concept multi-plate-forme relève souvent du rêve. Même la référence Google Maps rencontre encore des problèmes avec Safari, le navigateur des Mac.

Pour ces sceptiques, Ajax est avant tout une lubie, qui autorise les développeurs frustrés par HTML à s'exprimer à nouveau avec un outil qui ressemble au client-serveur. Tant que les outils ne seront pas au niveau des ateliers de développement du marché – et cela prendra du temps – « toute autre solution sera toujours plus efficace qu'Ajax ». En attendant, le seul intérêt qu'ils lui reconnaissent est d'éviter de recharger plusieurs fois les mêmes éléments d'une page. Ajax est donc parfait pour économiser de la bande passante. Mais pour le client-serveur, il repassera.

Des bibliothèques masquent la complexité

Faux ! rétorquent les enthousiastes. Ajax à beau être jeune, son écosystème s'étend à une vitesse phénoménale. L'intérêt est tel que, depuis cet été, il ne se passe pas une

Ce qu'ils pensent d'Ajax

LE SCEPTIQUE



Laurent Bédé,
directeur technique
chez BNP-Paribas

« Il ne faut pas enterrer le client lourd »

« Le déploiement en client-serveur n'est plus un problème. Avec un nombre fini d'utilisateurs, des outils comme Java Web Start assurent un déploiement transparent. Fiducia AG, par exemple, a déployé automatiquement une application Java/Swing de 120 Mo sur plus de 100 000 postes. Derrière Ajax, il y a un bricolage incroyable. Même si les outils s'améliorent, les meilleures bibliothèques d'aujourd'hui atteignent à peine le niveau de ce que faisait Visual Basic il y a dix ans. Quel intérêt y a-t-il à rester centré sur le navigateur ? C'est le nivellement par le bas. Le client du futur, le téléphone, a depuis longtemps opté pour un modèle avec des applets. »

LE PRAGMATIQUE



Thierry Nivelet,
fondateur
de l'éditeur Abaque

« Un bon moyen de reprendre les applications existantes »

« Au lieu de redévelopper l'application Windows en Ajax, nous la faisons tourner sur un serveur. Les écrans sont ensuite passés dans une moulinette pour ne reproduire que les modifications sur le client web. L'application Ajax ne se charge que de l'affichage et du renvoi des frappes clavier et des clics souris. Ils sont interprétés par notre serveur et renvoyés à l'application d'origine, qui retournera les changements d'affichage vers le client. Cela fonctionne un peu comme Citrix. A la différence qu'il n'y a pas de flux échangés, juste des événements graphiques. Résultat : l'application d'origine est peu modifiée, et elle fonctionne à distance. »

L'ENTHOUSIASTE



Dominique Derrien,
développeur et architecte
chez Oracle

« Une technologie qui va dans le sens de l'histoire »

« Le seul critère bloquant pour basculer des applications client-serveur en Ajax reste l'accès transparent au disque local. Pour le reste, Ajax n'apporte que des bénéfices : c'est le client web facile à déployer et à maintenir. Il n'y a pas de protocoles cachés ou bloqués par les pare-feux, il est rapide et flexible. Les critiques formulées sur Ajax sont valides, mais les choses s'améliorent tous les jours. Nous verrons bien où l'on en sera dans cinq ans. Quant à ceux qui dénigrent Javascript, ils ont tort. Certes, il a besoin d'être manipulé par des développeurs expérimentés. Mais on pourrait bientôt imaginer un Javascript++, comme C++ avec C. »

semaine sans l'annonce de nouvelles bibliothèques de développement. Le niveau de connaissance augmente, et la plupart des problèmes de jeunesse sont réglés ou en passe de l'être. Des différences existent entre les navigateurs ? Des bibliothèques viennent masquer cette complexité. XML est trop verbeux pour certains transferts de données ? Préférez-lui JSON. Ses objets Javascript binaires plus compacts réduisent les traitements côté client. C'est la méthode adoptée par Google Maps. Mais quid des outils de test et des ateliers de développement visuel ? Ils arrivent doucement chez Tibco (General Interface) et Backbase. Quant à JSUnit, HTTPUnit et Selenium, ils assurent le débogage et JSDoc la documentation. On attend aussi beaucoup de Microsoft, qui vient de présenter Atlas, son framework Ajax pour Visual Studio. Bien sûr, ces outils ne sont pas encore au niveau de ceux que l'on trouve dans d'autres environnements. Mais ils progressent vite. L'idéal serait de standardiser les bibliothèques pour ne pas avoir à réapprendre une interface de programmation à chaque fois que l'on change de fournisseur.

Une plus grande liberté de manœuvre malgré ses défauts

Se pose aussi le problème du multimédia. Sans extension, le navigateur ne sait produire ni son ni animation. Un sérieux handicap face au client-serveur. Alors certains ont eu l'idée d'utiliser Flash. Il est quasiment installé sur tous les navigateurs, et rien n'empêche de le piloter avec Javascript au lieu de l'Actionscript de Macromedia. C'est d'ailleurs la technique qu'a retenue Goowy pour son bureau virtuel. Le résultat est saisissant : on se croirait sous Mac OS X !

A voir certaines réalisations, cloner une application client-serveur en Ajax ne poserait

aucun problème. Jackbe a ainsi reproduit à l'identique une application de gestion Visual Basic. Multifenêtrage, boutons avec icônes, raccourcis clavier, tout y est. Ajax laisse tout de même une grande liberté de mise en œuvre. Certains choisissent d'effectuer un maximum de traitements sur le serveur, n'abandonnant au navigateur que la partie présentation. D'autres optent pour un maximum de traitements sur le client. Mais attention à l'overdose de Javascript dans les pages. Les temps de chargement s'allongent, et l'application est ralentie. Tout réside dans un savant dosage.

La guéguerre entre partisans et réfractaires à Ajax risque finalement d'être arbitrée par les utilisateurs. Quand ils auront goûté à des applications web 2.0 comme Gmail, ils auront du mal à revenir en arrière. Ils ne comprendront pas pourquoi leur application web d'entreprise est si lente et si peu ergonomique. Ce sont eux qui pousseront à adopter Ajax.

Restera tout de même un problème insoluble : le travail hors ligne, sur un portable, quand il n'y a pas de connexion internet. Car même si elle a l'air d'une application locale, une application web restera toujours une application web. ● ANICET MBIDA

Pour en savoir plus

- Qu'est-ce que le web 2.0 ?
- Qu'est-ce qu'Ajax ?
- Les meilleurs sites Ajax
- Les bibliothèques et les ateliers Ajax

Retrouvez toutes les adresses sur :
www.01blog.fr/1835