

Au-delà du navigateur

Redécouvrir le rôle du bureau dans un monde réseau-centré

Titre original anglais : Beyond the Browser

Richard Gaskin

Fourth World Media Corporation

netapps2010@fourthworld.com

Première ébauche publiée : 20 septembre 2001

Dernière révision : 17 avril 2009

Traduction de l'anglais par René Micout le 13 mai 2010

Texte original : <http://www.fourthworld.com/embassy/articles/netapps.html>

Copyright 2001–2010 Fourth World Media Corporation

© 2010 René Micout pour la traduction française

1. Introduction : Internet n'est pas le web

Internet est utilisé pour une grande variété de tâches, du transfert de fichiers à l'envoi d'e-mail. Ces tâches comprennent également la visualisation de textes, d'images et de contenus multimédia en utilisant le World Wide Web. Alors que le Web est un nouveau venu relativement récent dans la gamme des services Internet, sa popularité a éclipsé les autres services Internet. Aujourd'hui, de nombreuses personnes pensent que "Web" et "Internet" sont synonymes.

Compte tenu de l'omniprésence croissante du Web, les concepteurs de nombreuses applications ont estimé que la prochaine étape logique serait l'usage des applications de bureau depuis le Web au travers d'un navigateur. Certaines tâches sont bien adaptées à l'utilisation d'un navigateur et tirent avantage de la simplicité de l'environnement du navigateur.

Mais cette simplicité entraîne de nouveaux challenges, depuis l'interface jusqu'à la gestion de protocoles autres que HTTP qui sont parfaitement bien gérés dans des applications dédiées.

Les traitements de texte et beaucoup d'autres application informatiques «traditionnelles» impliquent des tâches qui sont isolées par nature ; les fichiers sont partageables et même modifiables par plusieurs utilisateurs, mais la tâche elle-même (utilisation d'un tableau, retouche d'une photo, rédaction d'une lettre) est en général réalisée par un seul utilisateur.

Il y a ensuite des tâches qui sont par nature réalisées en groupe et qui tirent avantage d'un traitement en réseau. Le courrier électronique, par

exemple, n'a que peu d'intérêt sans l'implication d'autres utilisateurs. C'est pour ces tâches informatiques «connectées» qu'Internet est particulièrement utile et c'est par l'intermédiaire d'un navigateur que ces tâches prennent toute leur valeur.

Le terme «Web App» (application Web) a été utilisé pour décrire les applications qui sont utilisées dans un navigateur. Cet article introduit la notion de «Net Apps» (application internet) pour décrire des applications connectées à Internet mises en œuvre par des processus indépendants avec leur propre interface utilisateur.

La figure 1 présente les différents types d'application. Dans une interface graphique moderne, toutes les applications sont des applications de bureau, le bureau agissant comme un espace dans le cadre de l'expérience-utilisateur. Un sous-ensemble d'applications sont des applications-réseau (Net Apps) qui utilisent des protocoles Internet pour fournir les prestations propres à l'informatique distribuée. Le navigateur Web est l'un d'entre eux mais il y en a beaucoup d'autres tels que le courriel, FTP, la messagerie instantanée, etc.

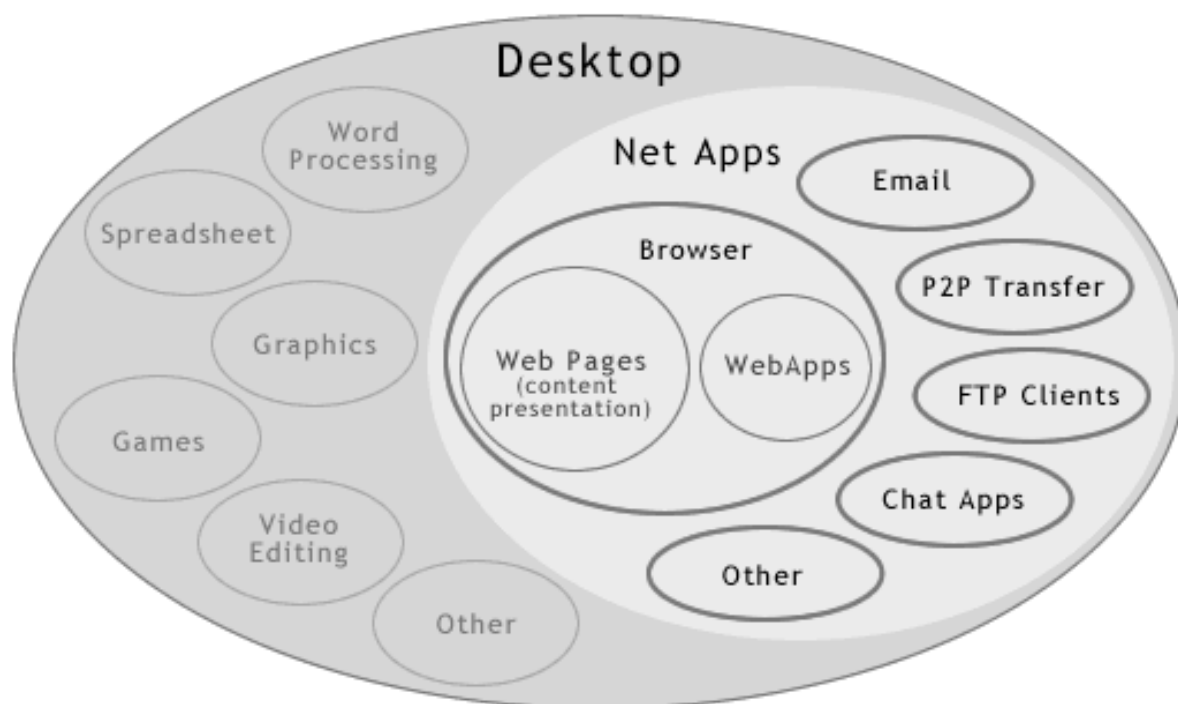


Fig. 1 : catégories d'applications logicielles et sous-catégories d'applications réseau (Net Apps)

Alors que le navigateur demeurera une option utile pour les applications réseau (Web Apps) la popularité croissante du peer-to-peer (P2P) tels des outils comme Napster et le réseau Gnutella ainsi que les outils de messagerie comme AOL Instant Messenger et ICQ, suggère que des applications dédiées seront plus appropriées pour certaines tâches.

2. Avec ou sans navigateur : affichage contre création

Au moment de choisir une alternative pour la mise en œuvre d'applications réseau, il est souhaitable d'étudier la nature des tâches à réaliser par l'application. Le navigateur présente des points forts et des points faibles et toutes les tâches ne sont pas prises en charge de manière optimale en considérant ses fonctionnalités de base. Cela s'explique en raison des tâches originelles pour lesquelles il a été conçu.

Le Web a été inventé pour permettre aux chercheurs de consulter des documents créés par d'autres chercheurs. Avec l'apparition du navigateur graphique ces documents ont pu également contenir des images et toutes sortes de médias. Les contrôles disponibles dans le navigateur reflète ce parti pris de visualisation passive axée sur la consultation de pages Web.

Plus tard, le langage JavaScript a été ajouté à Netscape Navigator et JScript à Microsoft Internet Explorer, permettant ainsi une meilleure interactivité avec les pages Web. Les langages de script permettent à la fois d'interagir côté client (une application «calculatrice» par exemple) mais également côté serveur comme le dépôt de formulaires nécessaires à une application CGI.

Les possibilités d'interaction du navigateur ont encore évolué avec l'apparition d'outils tels que Java et Macromedia Flash. Même si les éléments d'interface peuvent être placés dans une page Web ils ne sont pas complètement intégrés dans le navigateur, et la disparité avec laquelle les navigateurs réagissent avec les pages ne comportant que du HTML et celles comportant du Flash ou du Java peut causer de la confusion chez l'utilisateur. Par exemple, si l'on clique sur le bouton «précédent» du navigateur tout en regardant une animation Flash, la plupart du temps cela entraîne la disparition de l'élément Flash plutôt que de revenir à la page Web précédente. C'est pour cette raison que beaucoup de développeurs utilisant Java ou Flash disposent des fenêtres secondaires pour dissimuler les contrôles de navigation et rendent ainsi l'aspect du navigateur proche de celui d'une application de bureau.

Ces tentatives pour fournir l'interactivité propre aux applications de bureau atteignent rapidement leurs limites. En plus de la disparité entre les fonctionnalités du navigateur et celles de l'applet résidant en son sein, le concepteur est confronté aux considérations suivantes :

Contrôles – Présentation

Un exemple de limitations dans la conception d'applications dans un navigateur (Web Apps) est la gestion dynamique de contrôles dont les

dimensions se modifient automatiquement pour s'ajuster à celles d'une fenêtre redimensionnée par l'utilisateur. Les gestions dynamiques sont difficiles à mettre en œuvre dans un navigateur alors qu'elle sont très courantes dans les applications de bureau. Le formulaire «Nouvel Email» d'AOL en est un bon exemple comme le montre la figure 2.

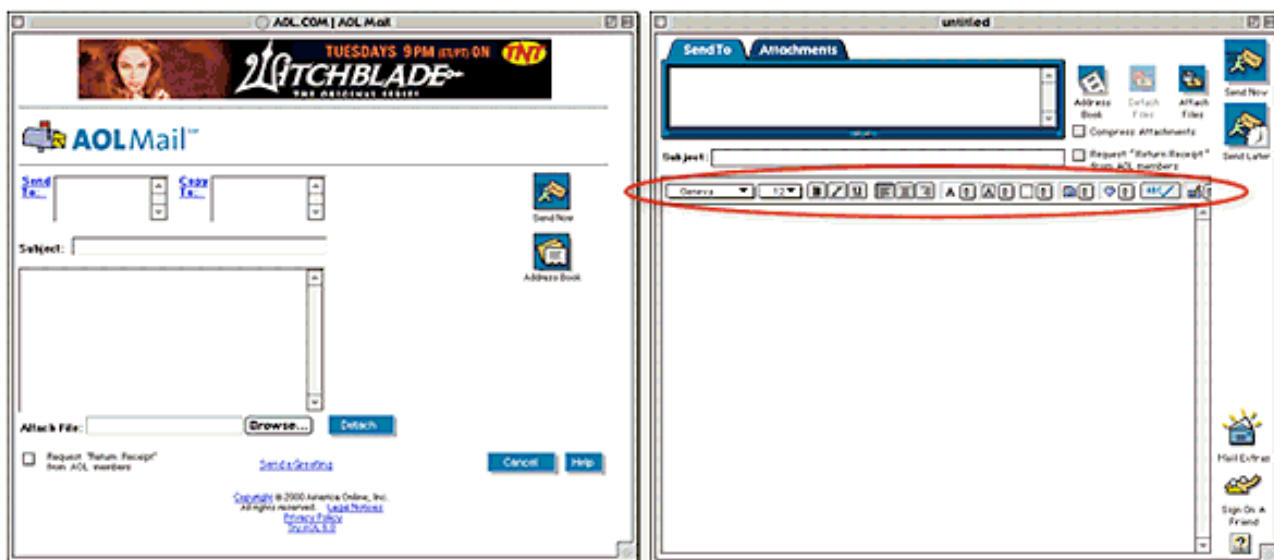


Fig. 2 : formulaire «Nouvel Email» d'AOL, version Web (à gauche) et dans le logiciel de bureau.

Les limitations du langage HTML ne permettent pas à la zone de saisie du message d'être redimensionnée de manière à correspondre à la taille souhaitée par l'utilisateur.

Notez également le peu de place occupée à l'écran par la partie véritablement fonctionnelle de l'interface. La plus grande partie de l'espace est utilisée par des éléments non fonctionnels tels la publicité et autres liens éloignant l'utilisateur de la tâche à accomplir.

Bien que participant de l'expérience du Web, de nombreux éléments sont une distraction pour l'utilisateur et deviennent en quelque sorte des «anti-fonctions».

Contrôles – Comportement

Les deux implémentations de l'interface Email AOL illustrent une autre limitation dans la conception d'applications dans un navigateur (Web Apps) : les contrôles ainsi que de nombreuses fonctionnalités sont difficiles voire impossibles à créer en utilisant HTML et JavaScript.

L'équipe Web d'AOL réalise une bonne mise en page des éléments de contrôle et des fonctionnalités essentielles de l'application de bureau en y intégrant des outils d'attribut de style (cf. fig. 2 contour rouge) alors que ceux-ci sont absents de la version «navigateur».

La possibilité de joindre des pièces à un envoi, de les compresser ou encore d'enregistrer un message sans l'envoyer sont autant de fonctionnalités présentes dans la version «bureau» que l'on ne retrouve pas dans la version Web.

La délimitation des zones cibles des boutons radio et des cases à cocher dans le navigateur est un bon exemple d'un détail inhabituel que l'on rencontre partout dans les pages Web. Dans les applications de bureau, ce sont la commande elle-même et son étiquette qui réagissent au clic de l'utilisateur alors que la plupart des navigateurs limitent la zone réactive à la seule commande comme nous le montre la figure 3. Cette limitation nuit à son efficacité selon la loi de Fitt ([Fitt's Law](#)) : «le temps d'atteindre une cible est fonction de la distance et de la taille de la cible». Seuls quelques navigateurs récents permettent de supprimer cette limitation, toutefois malgré une mise à jour du navigateur les pages doivent être réécrites pour que le problème soit résolu.

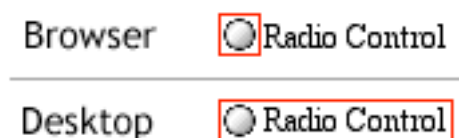


Figure 3 : contrôles «Radio», la zone cible est indiquée par le liseret rouge (browser = navigateur, desktop = application de bureau).

Menus

Les utilisateurs sont habitués à trouver les commandes dans la barre de menu. Dans un navigateur la barre de menu est intégrée au navigateur lui-même. Sans la possibilité pour les applications Web de modifier la barre de menu du navigateur, les concepteurs ont trouvé des solutions détournées pour créer des menus déroulants en JavaScript. Ces solutions utilisent un large éventail d'éléments interactifs souvent sans cohérence entre les différentes applications et ne se comportant presque jamais selon le mode d'utilisation normale des menus déroulants.

Ces incohérences sont encore aggravées par des différences dans l'implémentation du langage JavaScript comme le relate tel ou tel utilisateur UNIX ou Mac OS à propos de sites développés exclusivement pour une plateforme.

Fichier E/S

La capacité à lire et à écrire des données dans des fichiers est essentielle pour toute application nécessitant la persistance des informations entre les sessions. Dotés d'un faible niveau de sécurité, les navigateurs ne

permettent par aux pages Web, ni même à des plug-ins, de lire et d'écrire sur disque. En revanche le navigateur offre, par l'intermédiaire de cookies, la possibilité de stocker quelques données permettant la recherche de données plus importantes stockées sur le serveur. L'utilisation des cookies nécessite de multiples transactions HTTP pour chacun des éléments délivrés à l'interface, alors qu'une application de bureau peut avoir accès, en quelques millisecondes, à d'importantes quantités de données stockées localement.

Réactivité

En plus de la latence générée par l'absence de support de fichier E/S, les Web Apps offrent encore moins de réponse que leurs homologues de bureau car elles nécessitent le chargement de l'interface à chaque fois que celle-ci est utilisée. En revanche, le client AOL fournit un excellent équilibre entre les éléments locaux et téléchargés en stockant d'importants éléments d'interface utilisateur sur la machine cliente et en téléchargeant de nouveaux éléments d'interface et du contenu au fur et à mesure des besoins.

Protocoles

Alors qu'Internet offre une grande variété de protocoles pour différentes tâches, le navigateur ne gère que HTTP en mode natif avec un support FTP limité. D'autres protocoles comme SMTP (courriel), NNTP (groupes de discussion), Gnutella (partage de fichiers), IRC (chat) et d'autres nécessitant Java ou un plug-in plus souvent pris en charge dans des applications dédiées.

Compte tenu de ce qui précède, nous pouvons nous rendre compte que certaines applications peuvent fonctionner correctement dans la fenêtre d'un navigateur alors que pour d'autres tâches, ce type de fonctionnement se révélera maladroit voire même impossible en dehors d'un environnement dédié. De nombreux articles, vantant les mérites d'applications basées sur un navigateur, ont été publiés, mais peu d'entre-eux, s'il y en a, ont été rédigés dans un navigateur, la quasi-totalité l'ayant été dans un traitement de texte.

Des tâches comme le traitement de texte, la production d'images et d'autres créations multimédia établissent une distinction utile dans le domaine des applications dédiées : elles impliquent la création de données. En revanche, le succès des applications consultables depuis un navigateur sont plus souvent orientées vers l'affichage de données.

3. Considérations sur la sécurité

Un des avantages perceptibles dans le choix de mettre en œuvre une application dans un navigateur est la sécurité. En limitant les E/S et par le biais de transmissions cryptées via des protocoles tels que HTTPS, les navigateurs offrent un niveau de sécurité qui est généralement perçu comme acceptable même pour les grandes entreprises.

Toutefois, il existe peu de restrictions de sécurité pour les objets embarqués tels que les contrôles ActiveX et les Xtra Director, objets qui ne sont pas rares sur le Web. Dans la plupart des navigateurs, l'utilisateur doit accepter explicitement le téléchargement et l'installation de ces objets. Mais remettre en cause la question de la sécurité du navigateur pour un cas isolé, peut ébranler la confiance que peut avoir l'utilisateur dans le site fournissant cet objet.

Étant donné la prévalence du code exécutable véhiculé sur Internet, illustrée par certains virus spécifiques à certains systèmes d'exploitation tels que Code Red ou SirCam, la capacité du navigateur à assurer la sécurité est somme toute limitée. Bien que de nombreuses entreprises choisissent des Web Apps pour des raisons de sécurité, la plupart d'entre-elles ont aussi fait le choix des systèmes d'exploitation Windows sur lesquels l'exploitation croissante des failles de sécurité, propres à cette famille d'OS, coûte à ces mêmes entreprises 3,6 milliards de dollars US pour débarrasser leurs systèmes des seuls virus Code Red et SirCam.

En terme de sécurité effective, l'utilisateur averti devrait avoir autant confiance en téléchargeant une application à partir d'un fournisseur identifié que lorsqu'il télécharge n'importe quel autre objet codé comme un contrôle ActiveX. Les Net Apps peuvent utiliser les mêmes protocoles que les applications consultables depuis un navigateur, y compris HTTPS, de manière à fournir un niveau de sécurité équivalent pour les données sortantes.

En termes pratiques, les risques liés à l'exécution d'une Net App autonome ne sont pas significativement plus élevés que ceux liés à de nombreux objets généralement intégrés dans les pages Web.

4. Options offertes aux Net Apps

À partir du moment que nous commençons à chercher des opportunités de développement en dehors du navigateur, nous constatons que la presse spécialisée dans ces solutions se focalise exclusivement sur le partage de fichiers peer-to-peer (P2P). Il est vrai que Napster et Gnutella ont effectivement beaucoup attiré l'attention, mais il existe sur Internet beaucoup d'autres choses que le partage de fichiers.

L'une des catégories dont la croissance a été la plus forte ces dernières années a été la messagerie instantanée, avec des outils comme AOL Instant Messenger (AIM), Microsoft Messenger, Yahoo Messenger, Hotline, ICQ et autres. En plus de permettre l'échange de texte en temps réel entre utilisateurs, la plupart de ces applications permettent aussi les discussions de groupes, le partage de fichiers et d'autres services.

Par ailleurs, la technologie «Push» avec laquelle les utilisateurs peuvent choisir de recevoir des informations spécifiques au lieu d'aller les chercher sur le Web, semble revenir en force. Alors que ses promoteurs initiaux, comme Marimba, étaient axés sur des pratiques marketing, la technologie «Push» peut s'avérer plus utile comme outils permettant à une entreprise de diffuser de manière opportune des informations multimédia à son personnel.

Lors de la conception de Net Apps sur mesure, vous pouvez tirer parti de toutes ces technologies d'une manière beaucoup plus souple que vous ne pourriez le faire dans un navigateur. Le tableau 1 présente une brève liste des protocoles Internet les plus courants et leur prise en charge par les navigateurs les plus utilisés :

Protocole	Description	Support navigateur
HTTP	HyperText Transfer Protocol (Web)	Oui
FTP	File Transfer Protocol (transfert de fichier)	Limité
SMTP	Simple Mail Transfer Protocol (Courriel)	Non
NNTP	Network News Transport Protocol (Newsgroups)	Non
IRC	Internet Relay Chat (Chat)	Non
GNUtella	(Transfert de fichiers, calcul partagé)	Non
Custom	Nouveau protocole à développer selon nécessité	Non

Tableau 1 : protocoles internet les plus courants

5. Outils de création de Net Apps

L'un des principaux facteurs ayant contribué au succès du Web est que celui-ci est indépendant de la plateforme utilisée. Par conséquent tous les outils utilisés pour la création de Net Apps doivent également pouvoir fonctionner sur le plus grand nombre de plateformes possible.

Même les Intranets d'entreprise, où c'est souvent le personnel de l'entreprise qui détermine les configurations, peuvent bénéficier de l'indépendance de plateforme. Il n'est pas rare que dans de grandes sociétés ayant, pour la plupart, des systèmes basés sur Windows, on trouve des serveurs UNIX et des clients Macintosh dans certains services.

Étant donné les nombreuses et significatives différences entre les systèmes d'exploitation, le développement d'applications multi-plateformes en langage traditionnel comme le C++ est trop onéreux pour la majorité des entreprises. C'est pourquoi sont apparus de nouveaux langages indépendants de la plateforme tels que Java.

Mais pour les raisons exposées par John K. Osterhaut dans son article [Scripting: Higher Level Programming for the 21st Century](#), il y a de nombreux avantages à utiliser des langages de quatrième génération (4LG) pour le développement de Net Apps. Un certain nombre d'outils multi-plateformes existants permettent le développement très rapide de ces applications.

Le tableau 2 présente une brève liste des outils de développement disponibles permettant d'accéder aux protocoles Internet et assurant le déploiement sur au-moins deux plateformes :

Outil de développement + lien	UNIX/Linux	Plateforme Win32	Mac OS
Java Sun Microsystems	Oui	Oui	Oui
Adobe AIR Adobe Systems	Oui	Oui	Oui
Tcl/Tk Scriptics	Oui	Oui	Oui
REALBasic REAL Software, Inc.		Oui	Oui
MetaCard MetaCard Corporation	Oui	Oui	Oui
RunRev (Revolution) Runtime Revolution Ltd.	Oui	Oui	Oui
Macromedia Flash MX Macromedia, Inc.	Oui (Player)	Oui	Oui

Macromedia Director Macromedia, Inc.		Oui	Oui
iShell Tribeworks		Oui	Oui
Rebol REBOL Technologies	Oui	Oui	Oui
Ruby Yukihiro Matsumoto	Oui	Oui	

Tableau 2: outils et langage de programmation Net App

Si vous utilisez un outil ou un langage de développement de Net App ne figurant pas dans la liste ci-dessus, merci de m'en informer à <mailto:netapps@fourthworld.com>.

6. Ressources

Ci-dessous, une liste de sources complémentaires d'informations liées aux applications Internet (Net Apps). Vous pouvez suggérer des ajouts à cette liste en envoyant un courriel à netapps2007@fourthworld.com.

Articles

[Good Riddance to Browser-Based Apps](#)

Visual Studio Magazine, 23 juin 2005

[Forms vs. Applications](#)

Jakob Nielsen's Alertbox, 19 septembre 2005

[Beyond Browsing: Syntek and Groove Networks get P2P ready to go to work](#)

James Karney, InternetWorld, septembre 2001

[Internet Perspective](#)

Bruce Tognazzini, AskTog.com, mai 2001

[www.p2p.edu: Rip, Mix & Burn Your Education](#)

by Thom Gillespie, Maître d'Igital , été 2001

[Ready for Web Apps?](#)

Jim Seymour, PC Magazine, octobre 1999

[The Death Of The Web Is Inevitable, According To Forrester Research](#)

Mariko Zapf, Forrester Research, Inc., mai 2001

[My View: X Internet](#)

George F. Colony, Chairman of the Board and CEO, Forrester, octobre 2000

[ZDNet Special Report: Instant Messaging](#)

ZDNet News

[Why 'Push' is now ripe \(again\)](#)

Charles Cooper, ZDNet News, février 2001

[Instant Messaging Goes to Work](#)

David Legard, IDG News Service, novembre 2000

[SETI and Distributed Computing](#)

Garrett Moritz 1998

[Web services a savior?](#)

Bob Lewis, InfoWorld, août 2001

[Web services on trial](#)

Bob Lewis, InfoWorld, août 2001

[SirCam worm still a serious threat](#)

Robert Lemos, CNET News.com, septembre 2001

[Code Red cost estimated at \\$2.6 billion](#)

Reuters, août 2001

[Stubborn AOL may have last laugh](#)

Jim Hu, CNET News.com, juin 2001

[P2P and XML in Business](#)

Brian Buehling July, 2001

[The misery of Web applications](#)

D.F. Tweeney, The Tweeney Report, mars 2000

[Web services](#)

Tom Sullivan, InfoWorld, mars 2001

[Freenet developer to create commercial apps](#)

Ed Scannell, InfoWorld, avril 2001

[Groove ships p-to-p platform](#)

Ed Scannell, InfoWorld, avril 2001

[The Death of TCP/IP](#)

[Why the Age of Internet Innocence is Over](#)

Robert X. Cringely, août 2001

[Internet Winter](#)

[Why Internet Security is an Oxymoron](#)

Robert X. Cringely, juillet 2001

[Response Times: The Three Important Limits](#)

Jakob Nielsen, extrait du chapitre 5 de son livre «Usability Engineering»

[Scripting: Higher Level Programming for the 21st Century](#)

J. Osterhaut, Scriptics, mars 1998

[The Case Against Strained Carrots](#)

Scott Raney, MetaCard Corporation 1996

[Drop-Down Menus: Use Sparingly](#)

Jakob Nielsen, Alertbox novembre 2000

[The Network is the User Experience:](#)

[Microsoft's .NET Announcement](#)

Jakob Nielsen, août 2001

[Runtime Revolution Embassy](#)

Référentiel «Fourth World» des ressources RunRev et base de RevNet.

Sites Web

[The Free Network Project](#)

[OpenP2P.com](#)

[.Net Dev Center](#)

[OreillyNet](#)

[Gnutella](#)

[MacOrchard](#)

[OSF Distributed Computing Environment](#)

[Distributed Computing Projects](#)
[Peer-to-Peer Working Group](#)
[InfoAnarchy](#)
[Wired's Guide to Global File-Sharing](#)
[Peertal](#)
[Napster](#)
[SETI@Home](#)
[Gnutella Clients](#)
[Distributed.Net](#)
[Jabber](#)
[MSN Messenger](#)
[ICQ](#)
[Yahoo! Messenger](#)
[AOL Instant Messenger](#)
[ACM Special Interest Group for Computer-Human Interaction](#)

Principes d'élaboration d'interface

[Microsoft Windows Human Interface Guidelines](#)
[Apple Mac OS X Aqua Human Interface Guidelines](#)
[Motif Style Guide](#)
[Irix User Interface Guidelines](#)
[KDE User Interface Guidelines](#)
[Java Look and Feel Design Guidelines](#)

Note : la plupart des propos tenus dans cet article sont basés sur des anecdotes, mais, et ce n'est pas faute d'avoir cherché, il semble y avoir pénurie de publication de recherche sur les différences de convivialité entre le Web et les applications de bureau. Si vous en avez la possibilité, vous pouvez mettre à jour cet article, je vous en serais reconnaissant. Si vous connaissez des liens intéressants ayant un rapport avec le sujet, merci de les poster sur NetApps2010@fourthworld.com