

# Séance 11

## Applications composites

---

INF 777

Automne 2017



## Plan de la séance

- 1. Applications composites (*Mashups et API*)
- 2. Langages et technologies : Ajax

## Web 2.0

- ❑ Réseaux sociaux, contenus générés par les utilisateurs, applications en ligne, blogs, podcasting...
- ❑ L'effet réseau
  - *Wikipedia* vs *Encarta* : BD s'enrichissent plus rapidement
  - Marketing et relations publiques : le point de vue des consommateurs
  - Applications Internet : Commentaires des utilisateurs s'immiscent dans l'assurance qualité habituelle (*beta* perpétuel, améliorations constantes, ex. *Facebook*, *Google*)

## Web 2.0

- ❑ Tendances
  - Localisation et mobilité
    - *Google Maps*, *Waze*, *Facebook*
  - Personnalisation
    - *Flipboard*
    - *Meteomedia.com*, *SAQ.com*, *Metro.ca*
    - *SoLoMo* : *Social / Local / Mobile*
    - *TripAdvisor*, *AirBNB* : Recommandations de nos contacts Facebook
      - <https://fr.airbnb.ca/help/article/198/what-is-the-social-connections-feature>

❑ Vidéo: The Machine is Us/ing Us

[http://youtu.be/NLIGopyXT\\_g](http://youtu.be/NLIGopyXT_g)

## Web 2.0

❑ Tirer enfin profit du HTML sémantique

- Un Web de **données** vs un Web de **documents**
- **Portabilité** des données : Séparer les données de la forme permet aux données de voyager :
  - Données ouvertes (*Open Data*) : Information brute, accessible, réutilisable et disponible publiquement
  - <https://www.donneesquebec.ca/fr/>
  - <http://donnees.ville.montreal.qc.ca/>
    - <http://donnees.ville.montreal.qc.ca/portail/applications/>
    - <http://villeintelligente.montreal.ca/>

## Web 2.0 : Applications composites (Mashups et API)

- ❑ Tendence à la réutilisation du code
- ❑ **Mashup (Application composite)**
  - Application Web qui combine des contenus provenant de plusieurs site ou applications Web
- ❑ **API : Application Programming Interface**
  - Ensemble de fonctionnalités qu'une application (ou programme) rend disponible pour une autre application (ou programme)
    - Ex.: Windows contient des centaines d'API qui permettent l'accès au disque dur, l'affichage à l'écran etc. Ces APIs sont ensuite utilisés par Word, Excel, etc.

## Web 2.0 : Applications composites (Mashups et API)

- ❑ Exemples de *widgets* : calendar.google.com, Twitter timeline
  - ! : Contenu indépendant de votre site
    - Sécurité
    - Disponibilité
    - N'est pas indexé
- ❑ Exemples de *mashups*
  - [Google Maps](#) (actuellement 50% des mashups <sup>(1)</sup> sur le Web)
    - Exemples : [googlemapsmania.blogspot.com](http://googlemapsmania.blogspot.com)
    - [www.bikemap.net](http://www.bikemap.net) / [www.landemtl.com](http://www.landemtl.com)
  - Amazon
    - [www.gettextbooks.com](http://www.gettextbooks.com)  
(API Amazon + Comparaison de prix de librairies)

<sup>(1)</sup> Source: [www.programmableweb.com](http://www.programmableweb.com)

## Web 2.0

### Questions soulevées

- Importance du contenu vs design ?
- Perte du « branding » en permettant aux données de voyager?
- Moins grande utilisation de la navigation à la source? Où situer l'ergonomie des interfaces dans un contexte où l'utilisateur est en contrôle?
- Perte ou gain d'audience?

### ☐ Réponse : l'**expérience** utilisateur au centre du développement

- Ce contenu doit tout de même être présenté et il faut y accéder...
- Comprendre les besoins et les tâches des utilisateurs
- UX vs UI

[http://www.digital-web.com/articles/web\\_2\\_for\\_designers/](http://www.digital-web.com/articles/web_2_for_designers/)

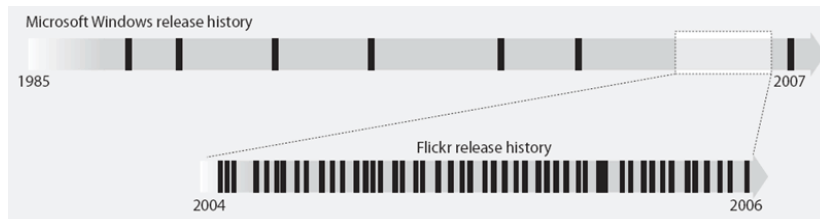
## Avantages du développement 2.0

- ☐ Mise en marché rapide
- ☐ Diminution du risque
- ☐ Relations avec les consommateurs
  - Ex.: Dell et Apple offrent une communauté d'utilisateurs en ligne qui échangent sur leurs problèmes... et prends en note au passage des idées pour les prochaines nouveautés de ses systèmes
- ☐ Données en temps réel permettant la prise de décision
- ☐ Réponse et vitesse d'adaptation accrues
  - Google Alerts / nexalogy.com
  - Ex. Première Moisson et YouTube

Web 2.0 Principles and Best Practices, Fall 2006, O'Reilly Media Inc.

## Web 2.0 : Meilleures pratiques de développement

- ❑ *Déployer tôt et souvent*
  - Développement Agile et itératif
  - Ex. MS Windows vs Flickr release history



- ❑ *Impliquer les utilisateurs en tant que co-développeurs et testeurs en temps réel*

Web 2.0 Principles and Best Practices, Fall 2006, O'Reilly Media Inc.

## Web 2.0 : Meilleures pratiques de développement

- ❑ *Créer de nouveaux produits et services de manière incrémentielle*
  - Ex. Google Maps : images satellites, calculateur de trajets...
- ❑ *Gestion des opérations : compétence cruciale*
  - Gestion au jour le jour des données et services
- ❑ *Utiliser des outils et langages dynamiques*
  - Développement Agile nécessite langages dynamiques procurant flexibilité, adaptabilité au changement et vitesse d'implémentation

Web 2.0 Principles and Best Practices, Fall 2006, O'Reilly Media Inc.

## Web 2.0 : Meilleures pratiques de développement

- ❑ Beta perpétuel : changement dans le cycle de développement logiciel habituel
  - La fin des « versions »
    - Sur le plan marketing (Google n'a pas de « versions »)
    - Dans les faits les itérations demeurent
  - L'arrivée de « services » en constant développement
  - Valider et raffiner les fonctionnalités auprès des utilisateurs OK ➔ mais ne remplace pas l'assurance qualité!
    - Un beta n'est pas ici une excuse pour un manque de stabilité ou de qualité
    - Risque de perdre des consommateurs

Web 2.0 Principles and Best Practices, Fall 2006, O'Reilly Media Inc.

## Web 2.0 : Meilleures pratiques de développement

- ❑ Attention aux excès
- ❑ Bien planifier les livraisons
- ❑ Assurer la disponibilité du service
  - Coûts et efforts de cette disponibilité (matérielle et humaine)
  - Ex.: Google Apps :
    - <http://www.google.ca/apps/>
    - <http://www.journaldunet.com/solutions/cloud-computing/panne-du-google-app-engine-octobre-2012-1012.shtml>
    - <http://www.google.com/appsstatus#hl=fr&v=status>
- ❑ Vie privé et sécurité des données
- ❑ Miser sur l'effet des premières impressions

Web 2.0 Principles and Best Practices, Fall 2006, O'Reilly Media Inc.

## Ajax

- ❑ **Asynchronous Javascript and XML** : Améliorer l'interaction *utilisateur – serveur*
- ❑ Requêtes de BD à un serveur nécessitent traditionnellement de recharger les pages Web
  - Ex. : Par un appel de formulaire (GET et POST)
- ❑ Ajax permet de faire une requête et d'obtenir une réponse du serveur sans recharger la page

## Exemples de sites avec Ajax

- ❑ **Ex.: Google Suggest**
  - [www.google.com](http://www.google.com)
  - Un javascript envoie la lettre saisie au serveur qui retourne une liste de suggestions
- ❑ **Google Maps**
  - [maps.google.ca](http://maps.google.ca)
- ❑ **Personnalisation d'interfaces**
  - [www.netvibes.com](http://www.netvibes.com)



## Ajax

- ❑ Solutions AJAX souvent disponibles à même les récents cadres sous forme de modules et de bibliothèques de scripts :

<http://www.smashingmagazine.com/2007/06/20/ajax-javascript-solutions-for-professional-coding/>

- ❑ JQuery

- <http://jqueryui.com>

Plus de détails sur l'objet javascript *XMLHttpRequest*

[www.w3schools.com/ajax/](http://www.w3schools.com/ajax/)

## Ajax

- ❑ Attention aux excès : garder à l'esprit les notions d'utilisabilité et d'ergonomie d'interface

- Surcharge d'informations ou de fonctionnalités
- Ex. : Cas d'un formulaire où au lieu de permettre la saisie de l'adresse postale, le code postal affiche une liste d'adresses à sélectionner
  - Problème de flexibilité / Manque de contrôle utilisateur / Problèmes si les données ne sont pas à jour

## Ajax

- ❑ Précautions et responsabilités pour un bon usage d'Ajax : Conserver les **principes du Web** (*Unobtrusive Ajax*):

```
<a href="#" class="ajaxtrigger">  
<a href="contenu.html" class="ajaxtrigger">
```

Permet une page fonctionnelle, avec ou sans javascript activé :

- Accessibilité / Référencement
- Contrôle-utilisateur de la destination du lien (clic pour une ouverture dans une nouvelle page ou onglet)

## Ajax

- ❑ Précautions

- Fournir la rétroaction (barre de progression pour les longs chargement)
- Possibilité d'arrêter une action (bouton « Arrêt »)
- Possibilité d'annuler une action (Retour en arrière)
  - Ex.: Boutons « Précédent » et « Suivant » du navigateur non-fonctionnels vis-à-vis des changements apparus sur la page (difficile de revenir en arrière dans une séquence de tâche)
- Messages d'erreurs appropriés (page ou information non-trouvée, délai trop long, demande d'authentification...)

# Ajax

## □ Limites et problèmes connus

- Difficulté de mettre en signet (Favoris) une page dont les contenus Ajax ont été modifiés
- Menus déroulants apparaissant à des endroits inattendus et difficiles d'accès si la souris quitte l'espace réactif
- Bogues fréquents pour les effets de surbrillance : un éléments pointé crée une surbrillance sur un autre
- Mise en page des écrans ne s'adapte pas aux variations de polices de caractères
- Défis de compatibilité de plateformes et de navigateurs
- Précautions techniques :
  - <http://www.smashingmagazine.com/2010/02/10/some-things-you-should-know-about-ajax/>
  - <http://www.alistapart.com/articles/userproofingajax>