

IFT2905 Interfaces personne-machine

Sébastien Roy

Département d'Informatique et de recherche opérationnelle
Université de Montréal



9 janvier 2007
Université de Montréal

- Professeur : Sébastien Roy
- Bureau : 2391 (2384/2262 pour le Lab Vision 3D)
- Courriel : roys@iro.umontreal.ca
- Web : <http://www.iro.umontreal.ca/~roys>
- Web : http://www.iro.umontreal.ca/~roys/fr_ifft2905.shtml

Un examen intra (20), un examen final (20).

L'évaluation se fait aussi par les travaux pratiques.

%		Durée (semaines)
10	TP 1: à déterminer	2
10	TP 2: à déterminer	2
10	TP 3: à déterminer	2
10	Projet: à déterminer	2
20	Examen Intra	
20	Examen Final	

Est-ce que je peux utiliser du code de l'internet?

NON. Sauf si c'est explicitement demandé.

Code d'honneur du DIRO

<http://support.iro.umontreal.ca/Codehonneur.pdf>

Et si je ne satisfais pas aux conditions d'utilisation?

C'est **0** pour le travail au complet.

Est-ce que je peux utiliser du code de l'internet?

NON. Sauf si c'est explicitement demandé.

Code d'honneur du DIRO

`http://support.iro.umontreal.ca/Codehonneur.pdf`

Et si je ne satisfais pas aux conditions d'utilisation?

C'est **0** pour le travail au complet.

Le contenu de ce cours est fortement inspiré des cours

MIT OpenCourseWare

6.831 User Interface Design and Implementation

Robert Miller

<http://ocw.mit.edu/OcwWeb/>

[Electrical-Engineering-and-Computer-Science/
6-831Fall-2004/CourseHome](http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-831Fall-2004/CourseHome)

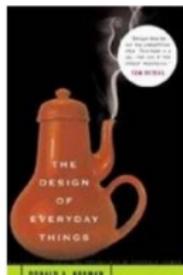
et

IICM (Institute for Information Systems and Computer Media)

Human-Computer Interaction

Keith Andrews

<http://courses.iicm.edu/hci/>



Livres obligatoires

- Donald A. Norman, *The Design of Everyday Things*, HarperCollins Canada / Basic Books, septembre 2002, ISBN 0465067107 (\$15)
- Jakob Nielsen, *Usability Engineering*, Morgan Kaufmann, septembre 1994, ISBN 0125184069 (\$31)
- Steve Krug, *Je ne veux pas chercher!*, Pearson Education France, 2e édition, mars 2006, ISBN 2744020303 (\$37)
- Steve Krug, *Don't Make Me Think: A Common Sense Approach to Web Usability*, New Riders Press, 2e édition (août 2005), ISBN 0321344758 (\$29)

Organisation du cours

Cours	Description
1	Introduction
2	Design orienté usagé
3	Architecture logicielle
4	Performances humaines
5	Modèles de sortie
6	Modèles et métaphores
7	Modèles d'entrée
8	Principes de design
9	Prototypes en papier
10	Contraintes et positionnement
11	Design graphique
12	Prototypes sur ordinateur
13	Outils pour les interfaces
14	Evaluation par heuristiques
15	Tests usagers
16	Conception et analyse des tests

Usability

The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.

— ISO 9241-11

- *effectiveness* \equiv efficacité
- *efficiency* \equiv efficience, rendement

Quelques définitions

Usability en français...

Utilisabilité

Capacité d'un système à permettre à ses utilisateurs de faire efficacement ce pourquoi ils l'utilisent. Afin que le travail soit fait, le système "utilisable" doit non seulement être facile à utiliser, mais aussi fiable et efficace.

— *Office québécois de la langue française*

Convivialité

Qualité d'un matériel ou d'un logiciel qui est facile et agréable à utiliser et à comprendre, même par quelqu'un qui a peu de connaissances en informatique.

— *Office québécois de la langue française*

Trois attributs mesurables de l'**utilisabilité** :

Efficacité

Degré de réalisation des objectifs à atteindre.

Effizienz

Rapport entre ce qui est réalisé et les moyens mis en oeuvre pour les atteindre.

Satisfaction

Absence d'inconfort, et attitude positive face à l'utilisation d'un produit.

On en ajoutera trois autres plus tard...



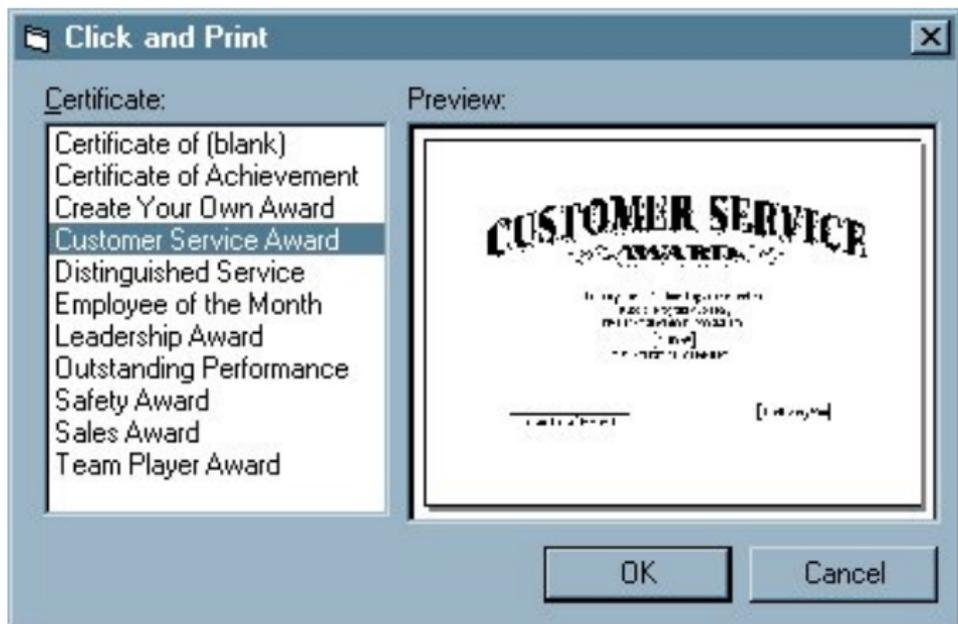
<http://homepage.mac.com/bradster/iarchitect/shame.htm>

On remarque...

- Pourquoi tant d'aide sur le coté?
 - *OKAY* et *OK* pas pareil
 - Pourquoi le texte est aligné à droite?
- Une barre de déroulement pour choisir un certificat?
 - Combien de certificats? Triés comment?
 - *Déroulement* inconsistant avec *Sélection*
 - Aucun raccourci possible

Peut-on faire mieux?

Exemple, revu et corrigé



First Launch Date: 09/09/97

Set Date

First Launch Time: 19:17

Set Time

Musée des horreurs

First Launch Date:	<input type="text" value="09/09/97"/>	<input type="button" value="Set Date"/>
First Launch Time:	<input type="text" value="19:17"/>	<input type="button" value="Set Time"/>

7:17 pm

am 11 12 1 pm

10 2

9 3

8 4

7 5

6

Left Mouse Button: Change Minute
Right Mouse Button: Change Hour

GIMP

- Pas de barre de menu
- Les menus apparaissent avec le bouton droit de la souris
 - Inconsistant.
 - Supposément plus rapide pour les experts...
- Problème de sous-menus

Emacs

- CTRL-S pour recherche incrémentale
 - réponse instantanée
 - réversible
 - feedback sur ce qui se passe
- Un problème: contrôles invisible
 - Comment débiter une recherche? Comment repartir à zéro?
 - Comment aller à l'envers? Pour tenir compte des majuscules?

Les interfaces usagers affectent la perception d'un logiciel

- Un logiciel convivial se vend mieux.
- Une page web inutilisable est abandonnée.

La perception est souvent superficielle

- Les usagers se blament pour les problèmes de l'interface.
- Ceux qui choisissent les logiciels ne sont pas toujours ceux qui les utilise.

- **Le temps des utilisateurs de logiciel coûte cher**
- **Plus simple de bien concevoir que de réparer les erreurs**
- **Les désastres sont possibles**
 - Thera-25 (machine pour la radio-thérapie), 1995, plusieurs morts
 - System radar Aegis, USS Vincennes, golf persique, 1988, 300 morts

Les Interfaces usagers (**IU**) sont difficiles à concevoir.

- Vous n'êtes pas l'utilisateur
 - Le génie logiciel s'intéresse surtout à la communication entre programmeurs.
 - Les IU s'intéresse à la communication avec les utilisateurs.
- L'utilisateur a toujours raison
 - les problèmes récurrents proviennent du système, pas de l'utilisateur.
- L'utilisateur n'a pas toujours raison
 - les utilisateurs ne sont pas des concepteurs de logiciel.
 - ils ne savent pas toujours *ce qui est bon pour eux*.

Facile à concevoir?

L'effort requis pour implanter des interfaces usagers est très important.

La programmation d'UI compte pour 50% de

- Temps de conception
- Temps d'implantation
- Temps de maintenance
- Taille du code

(étude de 74 logiciels, 1992)

Comment l'utilisateur réussit-il à utiliser les fonctionnalités disponibles?

Les dimensions (attributs) de l'utilisabilité

- Simplicité d'apprentissage
- Efficacité
- Efficience
- Mémorabilité
- Erreurs
- Satisfaction

En détail...

Dimensions de l'utilisabilité

Simplicité d'apprentissage

Facilité avec laquelle les usagers apprenent et utilisent l'interface.

Efficacité

Degré de réalisation des objectifs à atteindre. Quand et comment est-ce que l'utilisateur considère qu'il a réalisé la tâche souhaitée?

Efficience

Rapport entre ce qui est réalisé et les moyens mis en oeuvre pour les atteindre. À quelle vitesse peut-on accomplir la tâche souhaitée?

Dimensions de l'utilisabilité

Mémorabilité

Mesure la facilité de se rappeler de comment fonctionne l'interface.

Erreurs

Mesure de la quantité et de l'impact des erreurs de l'usagers.

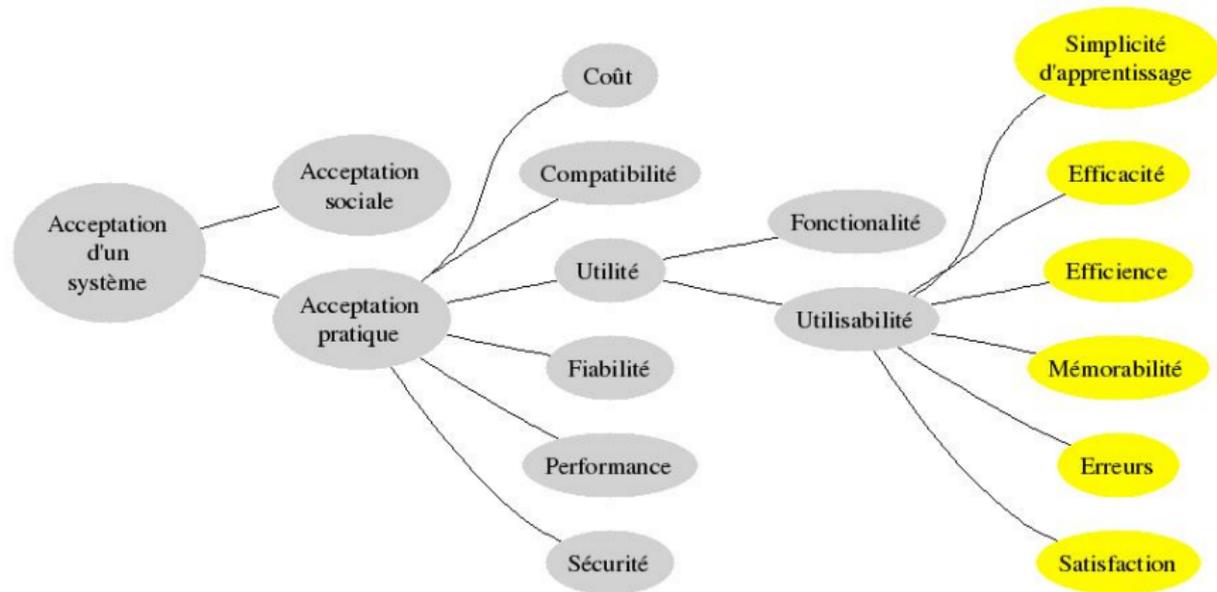
Satisfaction

Absence d'inconfort, et attitude positive face à l'utilisation d'un produit.

Ces attributs ne sont qu'une partie de ce qui rend un système acceptable...

Acceptabilité d'un système

L'utilisabilité n'est qu'un aspect du développement d'un système.



Il y aura des compromis à faire...

Les dimensions de l'utilisabilité ne sont pas toutes d'égale importance

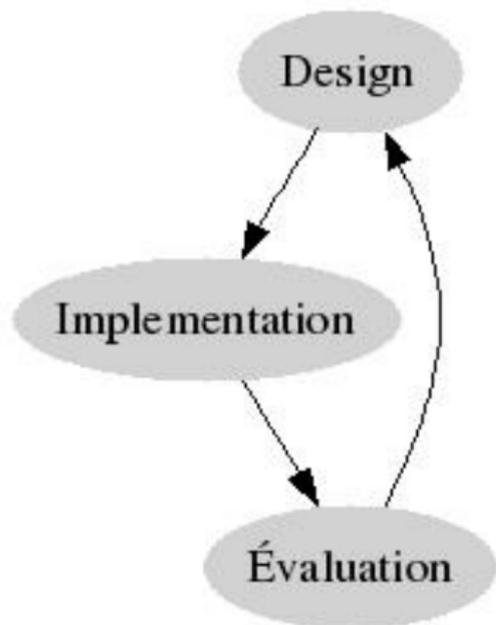
- Un usager débutant requiert de la *simplicité d'apprentissage*.
- Un usager occasionnel requiert de la *mémorabilité*.
- Un usager expert requiert de l'*efficience*.

Le processus de développement est itératif.

- Construire un prototype
- évaluer l'interface
- raffiner le design et corriger les problèmes

Spirale itérative

On répète tant qu'il reste du temps ou de l'argent!



Connaître l'utilisateur

Il faut en savoir le plus possible sur l'utilisateur éventuel.

- observation, interviews
- profils d'utilisateurs
- objectifs et attitudes des utilisateurs
- Comment l'utilisateur travaille-t-il?
- Scénarios d'utilisation

Des principes généraux peuvent aider à éviter les erreurs évidentes, mais ne donnent pas toutes les réponses.

- Prototypage
 - Simple, "jeter après usage"
 - Grossier: prototype en papier
 - Moins grossier: HTML, Visual basic, scripts
- Techniques de programmation d'interfaces
 - Modèles d'entrée et de sortie
 - Boîtes à outils (toolkits)
 - Constructeur d'interfaces

On évalue un prototype au moyen de différents tests.

- Évaluation d'un expert
 - Heuristiques et *walkthrough*
- Évaluation prédictive
 - on test avec un usager simulé *typique*
- Évaluation empirique
 - observation du comportement d'un usager en action

L'évaluation d'un prototype est la seule façon de mesurer l'utilisabilité.

