Cours 6b:
ergonomie des interfaces
(utilisabilité, usabilité)

Anastasia.Bezerianos@lri.fr

(plusieurs slides sont basés sur des slides de N. Roussel, O. Chapius, T. Tsandilas, W. Mackay, M. Beaudouin Lafon, et S. Greenberg)

evaluation :
est-ce que ça marche?

pourquoi évaluer?

Phases initiales de conception
- Développer et évaluer des idées de conception initiales avec l'utilisateur (conception participative)

Conception itératif
- Le comportement du système correspondent aux besoins
- Résoudre des problèmes spécifiques
- Choix entre solutions alternatives

Les tests d'acceptation
- Vérifier que le système répond aux critères attendus

Idéal : évaluation avec des vrais utilisateurs

Evaluation d'un système

Techniques informels et rapides :

Heuristiques

Evaluation Heuristique d'usabilité

Design Walkthrough

Techniques formels :

Etude des alternatives aux utilisateurs

Expériences contrôlées

Quasi-expériences

Autres …

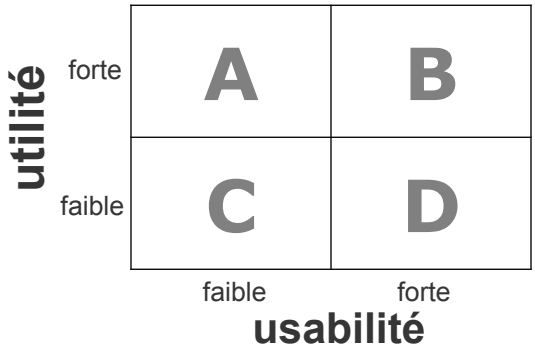
ergonomie des interfaces (utilisabilité, usabilité)

« Le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié.
(Norme ISO 9241)

Un système ergonomique est : facile à apprendre et mémoriser, efficace, visuellement agréable et rapide à reprendre des erreurs.

utilité

Atteindre des besoins spécifiques et soutenir des tâches réelles



D est meilleur que A ? Que pensez-vous ?

Avant de commencer

Source des critères et recommandations

De l'ergonomie du logiciel au design des sites web,
Jean-François Nogier, Dunod, 2002
Ten Usability Heuristics, http://www.useit.com/, Jakob Nielsen

Avantages

Ils permettent de corriger de nombreux problèmes d'utilisabilité
Ils sont faciles à retenir, et assez simples à appliquer
Ils peuvent servir à une évaluation simple et rapide d'un système

Inconvénients

Ce n'est pas une check-list
Il y a quelques redondances
Certains critères sont à manier avec précaution
Chaque règle peut avoir des exceptions

Catégories

1- Compatibilité
2- Guidage
3- Homogénéité
4- Flexibilité
5- Traitement des erreurs
6- Concision

Compatibilité

Compatibilité

Capacité du système à s'intégrer dans l'activité des utilisateurs

Recommandations :

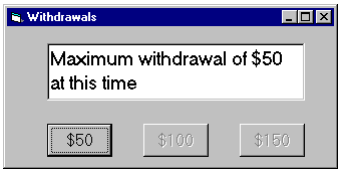
- parler le langage de l'utilisateur
  ex. système de PAO pour typographe vs. secrétaire
- présenter les informations de façon cohérente par rapport aux autres supports de travail
  ex. version électronique d'un formulaire papier
- l'accès aux commandes doit être compatible avec la tâche de l'utilisateur
  ex. commandes fréquentes au premier niveau des menus

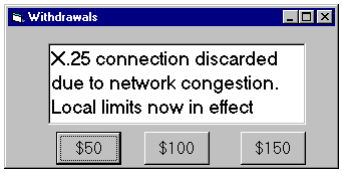
Nécessité d'étudier et d'analyser les pratiques de travail des utilisateurs visés

Compatibilité (suite)

Parler le langage de l'utilisateur

ex. retirer de l'argent d'un guichet automatique





Utilisez raccourcis et icônes faciles à mémoriser

ex. Enregistrer Fichier (File) / Enregistrer (Save)

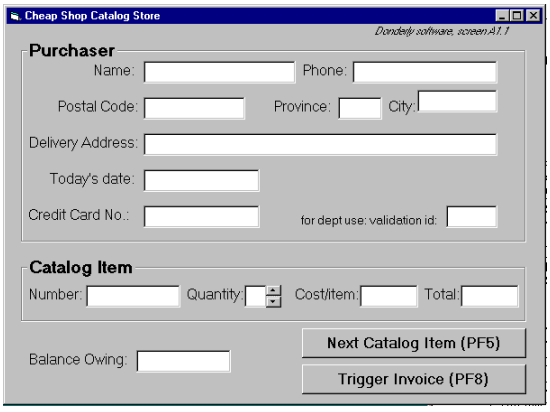
Alt+F+S (accès au menus)

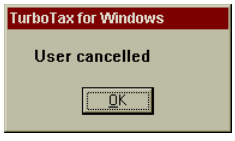
Ctrl + S (raccourcis)

(icône)

Compatibilité (suite)

Présenter les informations de façon cohérente par rapport aux autres supports de travail

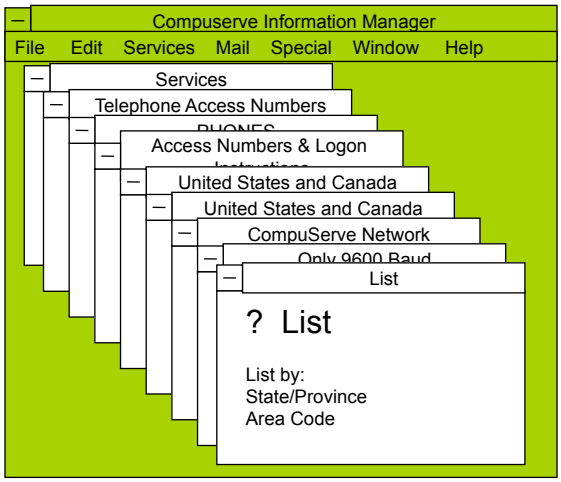




???

Compatibilité (fin)

L'accès aux commandes doit être compatible avec la tâche de l'utilisateur



Guidage

Guidage

Objectif : faciliter l'utilisation du système et son apprentissage

Mécanismes de retour (*feed-back* et *feed-forward*) pour

- réduire la charge cognitive
- prévenir des situations d'erreur
- rassurer (ex : progression d'une opération)

Guidage (suite)

Recommandations : incitation (feed-forward)

- griser les commandes non disponibles
- fournir la liste des saisies attendues
- donner le format de saisie des données
- donner des valeurs par défaut « intelligentes »

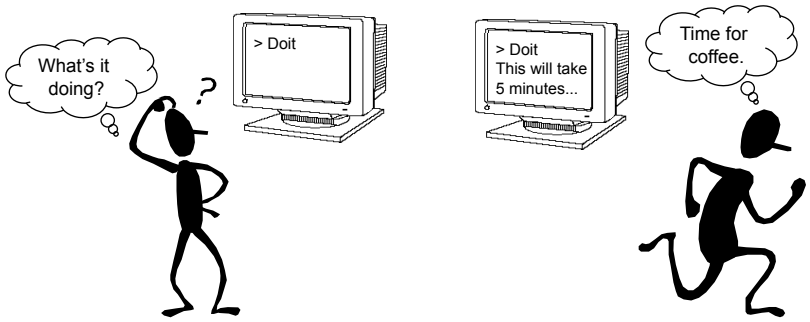
Recommandations : retour utilisateur (feed-back)

- à toute action de l'utilisateur doit répondre un changement de présentation de l'interface
- signaler les traitements longs par un message d'attente
- indiquer les modes de fonctionnement du système
- toujours faire apparaître les saisies de l'utilisateur
- rendre visibles les traitements réalisés par le système

Guidage (suite)

Traitements longs : la perception du temps …

- t <0.1 sec : perçu comme instantané
- t < 1 sec : le délai est perçu, mais ne perturbe pas l'utilisateur
- t > 10 secondes : l'utilisateur va vouloir faire autre chose en attendant



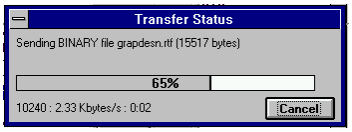
Guidage (suite)

Pour les longs délais :

Pour les transactions court

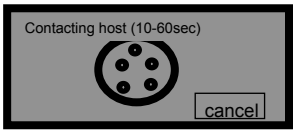
- changement de curseur

Pour les taches déterminées



- temps qui reste
- temps estimé
- avec barre de progression

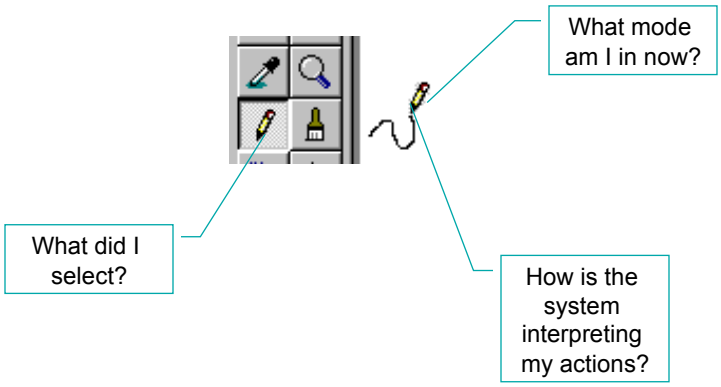
Pour des temps inconnus



- animation aléatoire

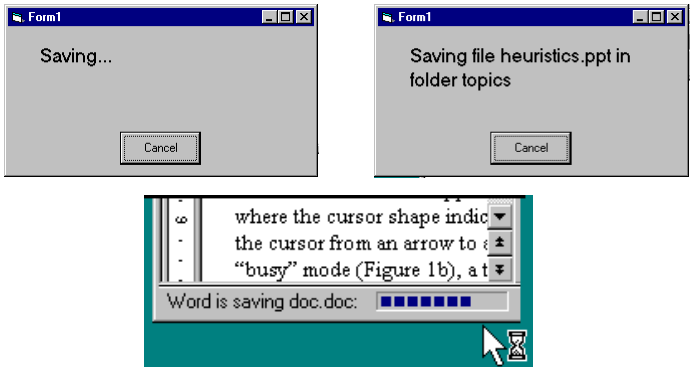
Guidage (suite)

Modes de fonctionnement



Guidage (suite)

Guidage spécifique est mieux

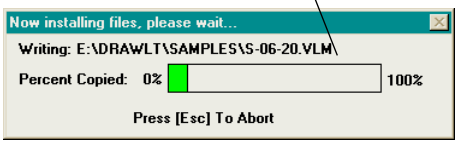


encore mieux en contexte de la tâche

Guidage (suite)

Guidage problématique

Multiple files being copied,
but feedback is file by file.



Drawing Board LT

Guidage (fin)

Recommandations : groupement/distinction (design graphique)

- regrouper les informations de même type par le format ou la position
- distinguer par une présentation différente les informations distinctes

Recommandations : lisibilité (typographie)

- utiliser une police droite
- employer des lettres sombres sur fond clair

Homogénéité

Homogénéité

Concerne la cohérence globale de l'interface

- interne : à l'intérieur d'une application
- externe : entre les applications, avec la métaphore du système

Principe : un système qui a l'air familier est perçu comme simple à utiliser

Intérêt : faciliter l'apprentissage et l'utilisation

Risque : freiner ou bloquer l'évolution (inertie des standards)

Homogénéité (suite)

Recommandations

- les fenêtres doivent suivre le même schéma d'agencement
  exemple : boîte de recherche en haut à droite
- la sémantique des boutons de la souris doit être constante
- le même vocabulaire doit être utilisé pour désigner les commandes du logiciel
  exemple : ouvrir / copier-coller / préférences / …
- la syntaxe des commandes doit être cohérente sur l'ensemble de l'interface

La consistance ne concerne pas que l'apparence visuelle
Autres exemples : syntaxe et effets des commandes

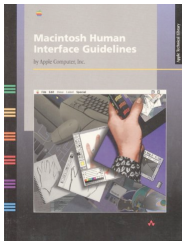
Homogénéité (suite)

Guides de style :

- publiés par les concepteurs d'interfaces graphiques
- décrivent le "look and feel" d'une plateforme
- sont souvent vicieux : faciliter la vie de ceux qui les suivent et compliquer celle de ceux qui ne les suivent pas...

Exemples :

- OSF Motif
- IBM CUA
- Apple Human Interface Guidelines
- MS Windows



Bons sur le principe, mais peuvent être difficiles à suivre
Implémentés (en partie) dans les boîtes à outils d'interface

Homogénéité (suite)

Motif style guide v1.1 : MessageDialogs should be used to convey a message to the user. They must not interrupt the user's interaction with the application. They should include a message, and one of the following button arrangements.

OK	Cancel
OK Help	Cancel Help
OK Cancel	
OK Cancel Help	
Yes No	Retry Cancel
Yes No Help	Retry Cancel Help
Yes No Cancel	
Yes No Cancel Help	

Homogénéité (suite)

Macintosh Human Interface Guidelines

Figure 5-11 Examples of correct and incorrect window titles

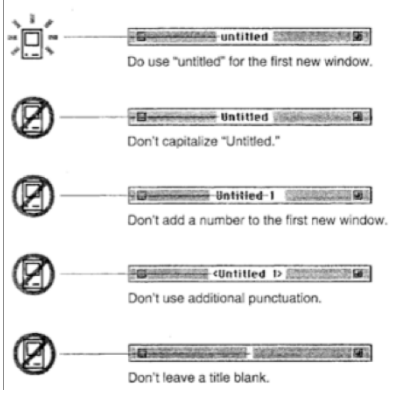


Table 10-1 Pointers

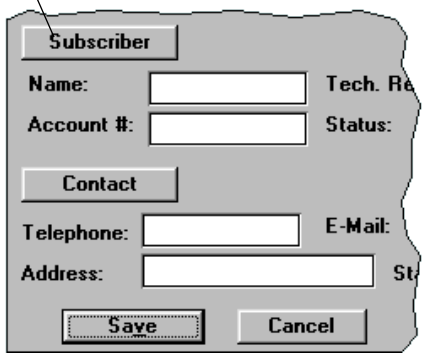
Pointer	Name	Used for
	Arrow	Scroll bar, other controls, size box, title bar, close box, zoom box, menu bar, desktop
+	Crosshairs	Drawing, shrinking, or stretching graphic objects
	I-beam	Selecting and inserting text
	Plus sign	Selecting fields in an array
	Wristwatch	Showing that a lengthy operation is in progress

Homogénéité (fin)

Homogénéité (et guidage) problématique

These are labels with a raised appearance.

Is it any surprise that people try and click on them?



Flexibilité

Flexibilité

Capacité de l'interface à s'adapter à différents contextes d'utilisation

Recommandations :

- permettre l'activation des commandes au clavier ou à la souris
- autoriser le déclenchement d'une commande fréquente depuis plusieurs fenêtres de l'application
- permettre à l'utilisateur de paramétrer le logiciel selon ses préférences
- fournir un moyen rapide d'accéder aux commandes des menus

Peut être contradictoire avec la concision (plus tard)

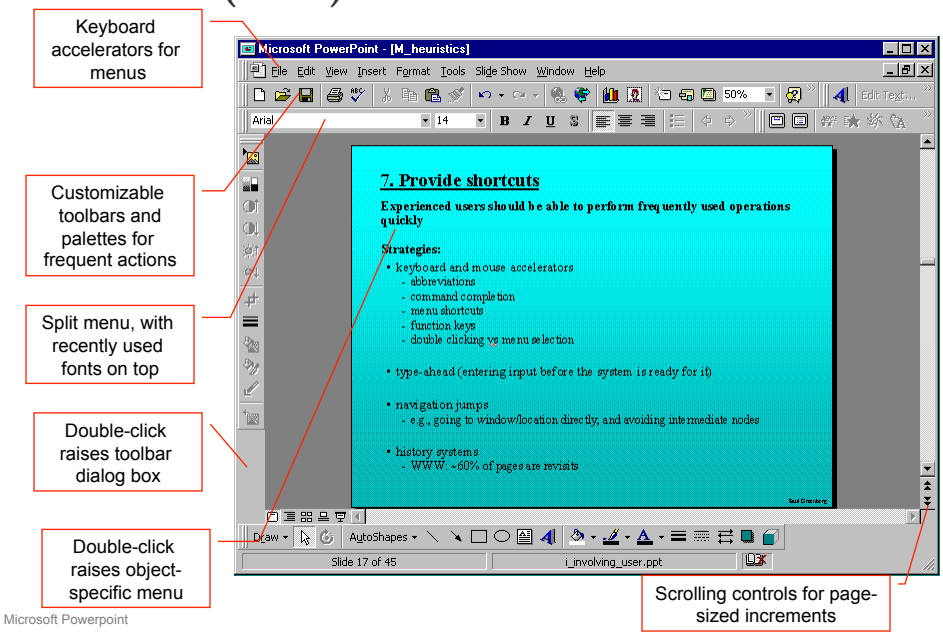
Flexibilité (suite)

Les utilisateurs expérimentés/experts doivent pouvoir se servir rapidement des opérations les plus courantes

Solutions :

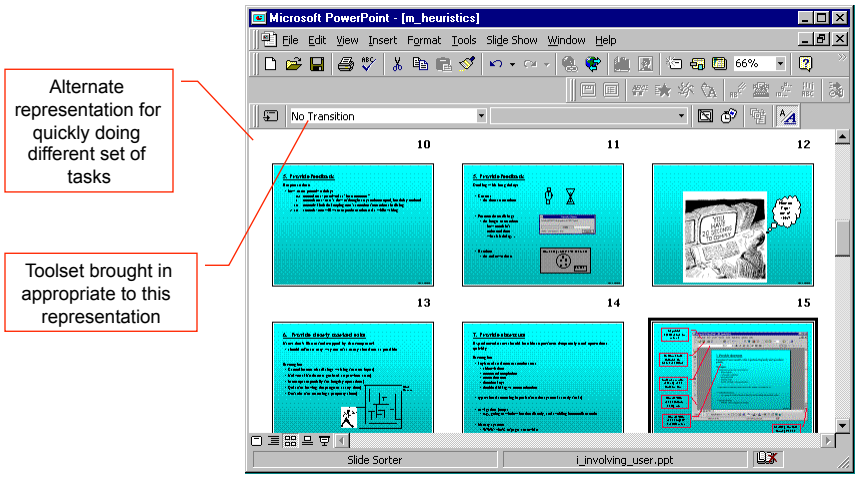
- icônes
- abbréviations / raccoursis (Ctrl-s)
- mnémoniques (Alt-f-s)
- touches spéciales (Xerox Star: move, copy, properties...)
- complétion automatique, frappe anticipée (*type ahead*)
- mécanismes de navigation, historiques
  - WWW: ~ 60% des pages sont des revisites
- redo

Flexibilité (suite)



Microsoft Powerpoint

Flexibilité (suite)



Microsoft Powerpoint

Flexibilité (fin)

Adaptabilité : personnalisation explicite par l'utilisateur
- lexique
- préférences de présentation

Attention à l'interface de configuration des préférences :
d'autant plus difficile que l'application est générale

Attention à ne pas transformer l'utilisateur en concepteur

Adaptativité : personnalisation dynamique sans action explicite de l'utilisateur

L'adaptativité est très controversée...
- non prédictibilité
- rupture du principe de cohérence

Traitement des erreurs

Traitement des erreurs

Regroupe les différents moyens visant à protéger l'utilisateur des erreurs et à corriger celles-ci

Recommandations générales :
- éviter les erreurs (!)
- détecter les erreurs au plus tôt
- permettre une correction aisée des erreurs
- faciliter l'exploration et l'apprentissage du système

Traitement des erreurs (suite)

Les utilisateurs vont faire des erreurs (c'est certain!)

Exemples :
- *habitude :* créer un nouveau document à partir d'un document existant, et l'enregistrer sans lui donner un nouveau nom
- *similarité :* l'action ressemble à d'autres qui sont possibles
- *fausse manipulation :* cliquer sur la commande voisine de celle que l'on vise
- *inattention* : oublier notre objectif en train de faire une séquence d'actions
- *incompréhension :* croire que l'on déplace une copie d'un objet alors qu'on déplace l'objet
- *exploration :* essayer un bouton que l'on ne connait pas

Erreurs causées par actions intentionnelles et non-intentionnelles

Traitement des erreurs (suite)

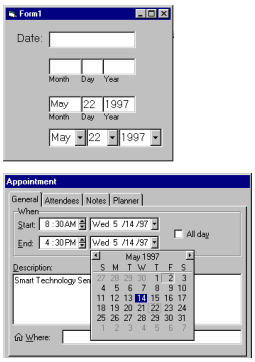
On peut tirer partie de la formation des habitudes
- la répétition et l'entraînement développent des habitudes
- *The ideal humane interface would reduce the interface component of a user's work to benign habituation. Many of the problems that make products difficult and unpleasant to use are caused by human-machine design that fails to take into account the helpful and injurious properties of habit formation. (J. Raskin)*

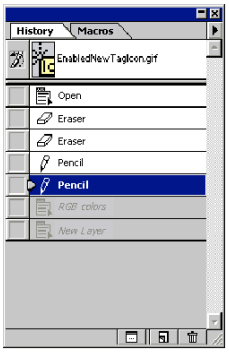
Réduire le nombre de choix peut favoriser le développement d'habitudes

Les habitudes sont sources d'erreurs :
- valider sans réfléchir une boîte de confirmation

Traitement des erreurs (suite)

Solutions (entre autres) :
- empêcher les manipulations qui produisent des erreurs
- faciliter la détection et la correction d'erreur par
  - le feed-back
  - un historique (*défaire - refaire*)

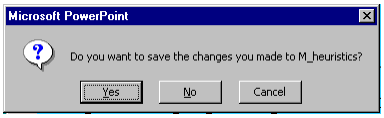




Traitement des erreurs (suite)

Recommandations : protection contre les erreurs

- feed-forward:
  - griser les commandes non disponibles
  - fournir la liste des valeurs possibles, et de bonnes valeurs par défaut
- minimiser les saisies clavier
- détecter les erreurs au plus tôt
  - faire des « checks » des valeurs raisonnables
- limiter les pertes de données (sauvegarde automatique, journal)
- valider explicitement les commandes irréversibles



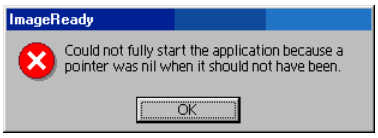
Traitement des erreurs (suite)

Recommandations : correction des erreurs

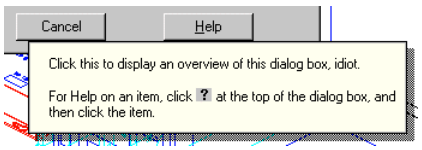
- placer le message d'erreur là ou l'utilisateur est censé regarder
- mettre en évidence le champ erroné
- afficher des messages d'erreur explicites, brefs, non réprobateurs et auto-suffisants

Traitement des erreurs (suite)

Les messages d'erreur sont importants

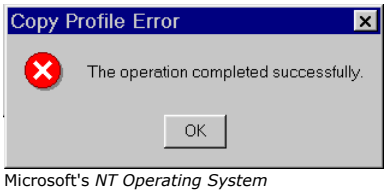


Adobe's *ImageReady*



AutoCAD Mechanical

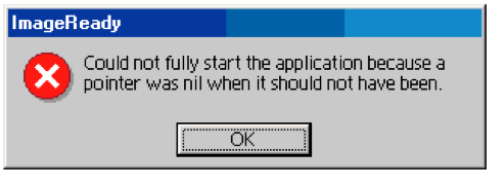
Windows Notepad



Microsoft's *NT Operating System*

Traitement des erreurs (suite)

Les messages d'erreur sont importants



Le message doit être rédigé dans des termes compréhensibles par l'utilisateur, si possible reliés à la tâche qu'il essayait d'accomplir

Eviter de donner l'impression à l'utilisateur qu'il est stupide

Essayer d'inclure une façon de corriger l'erreur

- Exemple : manque d'espace-> vider la corbeille

Traitement des erreurs (suite)

L'utilisateur n'aime pas se sentir "piégé »

- Engagement prématuré : forcer à prendre une décision trop tôt
- Essayer de toujours lui laisser une porte de sortie clairement identifiée

Solutions :

- bouton Annuler (*cancel*)
- bouton Reset/Default pour annuler des changements
- commande Défaire (*undo*), avec historique, et Refaire (*redo*)
- commande Interrompre (et/ou Pause)
- commande Quitter

Traitement des erreurs (suite)

Prévoir un **système d'aide**

Remarque : le système d'aide n'est pas là pour aider à faire passer une interface mal conçue

La plupart des utilisateurs ne lisent pas la documentation

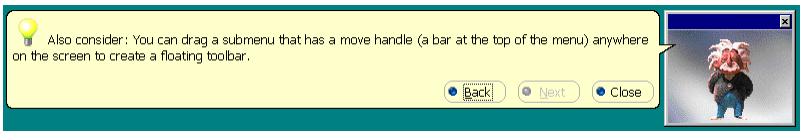
Situation la plus courante : le « mode panique »

- besoin ponctuel, spécifique et urgent
- nécessite une documentation en ligne avec des mécanismes d'indexation

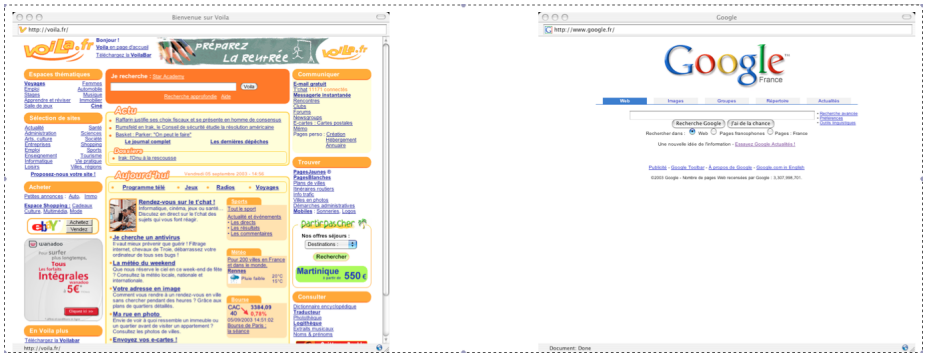
Traitement des erreurs (suite)

L'aide peut prendre différentes formes :
- tutoriels, démonstrations (« comment créer un tableau »)
- manuel de référence (table des matières, index, etc.)
- mémento (cartes de référence, tip of the day, tooltips, glossaire)
- aide contextuelle (qu'est ce que c'est ?)
- assistants

Exemples d'aide contextuelle :
- où suis-je ?
- où est...?
- et maintenant ?
- que s'est-il passé ?
- pourquoi est ce que ça ne...?



Concision



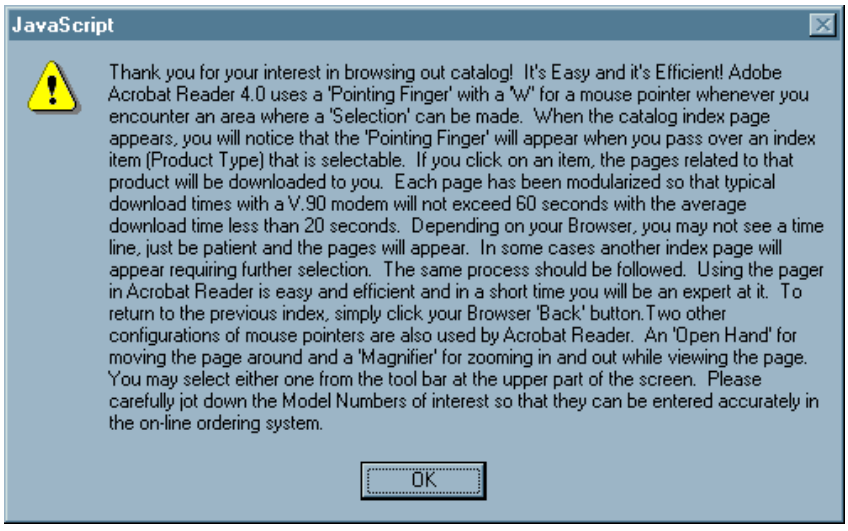
Concision

Ensemble des moyens visant à réduire la charge perceptive et mnésique de l'utilisateur

Recommandations :

- n'afficher que les informations pertinentes
  (mais cela dépend de ce que veut faire l'utilisateur)
- réduire le nombre d'actions élémentaires pour atteindre un objectif donné
- minimiser les saisies/lectures
- éviter les textes trop verbeux
- ne pas demander de saisir des informations
  qui peuvent être déduites
- éviter à l'utilisateur d'avoir à se souvenir d'informations
- ne pas l'obliger à faire des calculs qui peuvent être automatisés

Concision (suite)



Concision (suite)

Présenter uniquement les informations nécessaires

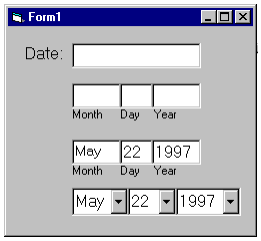
- « less is more » : limiter le nombre de widgets et de fenêtres qui introduisent des manipulations suplémentaires
- cacher ou supprimer les informations non essentielles (cf. progressive disclosure du Star)
- utiliser si possible l'ordre naturel des choses

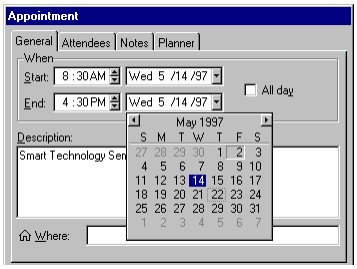
Préférer les techniques de manipulation directe aux techniques nécessitant la saisie de commande ou d'arguments (*point and click instead of remember and type*)

Concision (suite)

Préférer un petit nombre de commandes génériques

Expliciter les contraintes existantes :





Reconnaître est plus facile que se souvenir, mais bien souvent, en terme de représentation graphique, *less is more*...

Evaluation Heuristique - Nielsen (1990)

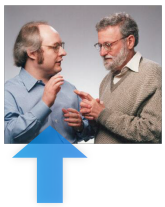
L'inspection systématique d'interface, en utilisant les catégories d'usabilité (ergonomie)

Procédure

- 3-5 inspecteurs (experts d'usabilité, les utilisateurs finaux)
- Inspecter l'interface (environ 1-2 heures pour les interfaces simples)
- Comparer leurs notes après

Marche pour storyboards, prototypes, et vrais systèmes

Evaluation Heuristique - Nielsen (1990)



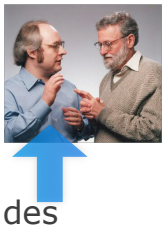
Non - guidée

- Exploration du système, sans tâche dirigée
- Aide à explorer les divers aspects de l'interface

Guidée par des Scénarios

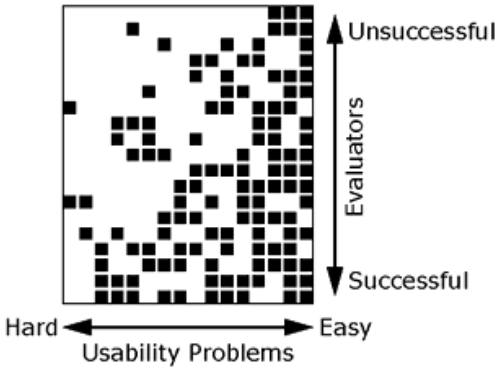
- Utiliser des tâches représentatives des utilisateurs
- Problèmes identifiés dans les parties pertinentes
- Fonctionnalités d'intérêt sont évaluées
- Mais des problèmes peuvent échapper

Evaluation Heuristique - Nielsen (1990)



Les évaluateurs/inspecteurs peuvent manquer des problèmes (faciles et difficiles à trouver)

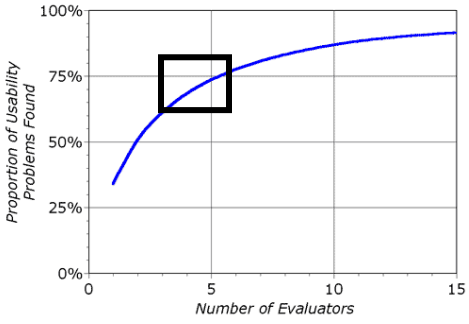
Les « meilleures » évaluateurs peuvent manquer des problèmes faciles
Les «pires» peuvent découvrir des problèmes difficiles
(ex. d'une évaluation)

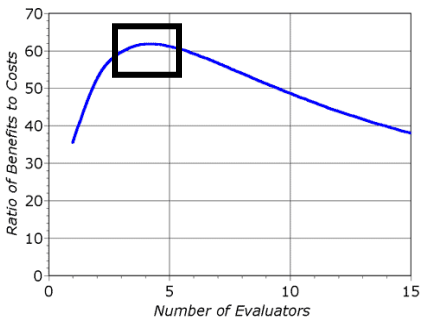


Evaluation Heuristique - Nielsen (1990)

3-5 évaluateurs trouvent 66-75% des problèmes d'utilisabilité

différentes évaluateurs trouver des problèmes différents (si ils travaillent seuls)





Design Walkthrough

Un groupe évalue un aspect d'un document spécifique, pas-a-pas :

source d'un programme	trouver les 'bugs'
conception d'un système	comprendre la structure
écrans graphiques	obtenir un feedback des utilisateurs
texte (articles scientifiques)	verifier la structure et la compréhensibilité
expérience	vérifier la méthode et les détails

Design Walkthrough

But :

Aider à identifier les problèmes informellement et rapidement, en utilisant des critères d'évaluation (à vous à preciser)

Procédure

- Choisir un petit groupe avec des rôles et des expertises différentes
- Fixer une durée, 1 heure au maximum
- Un présentateur déroule le scénario (storyboard, prototype video)
- Choisir le niveau des critiques
- Le groupe identifie autant de problèmes que possible
- Utiliser des règles pour aider à trouver les problèmes (ex. Principes de conception, spécifications, critères d'usabilité)

Design Walkthrough : Types de commentaires

Specifique
- Il faut trois étapes pour faire un recherche simple

Fonctionalité manqué
- Pas d'aide, besoin de recherche

Bugs
- L'import de fonctionalité X ne fonctionne pas

Suggestions
- Une vue generale sur tous les données crées

Generale (le moins utilile)
- Difficile à utiliser, Trop d'icônes

Evaluation d'un système

Techniques informels et rapides :
- Heuristiques
- Evaluation Heuristique d'usabilité
- Design Walkthrough

Techniques formels :
- Etude des alternatives aux utilisateurs
- Expériences contrôlées
- Quasi-expériences
- Autres …