

# Interaction Homme-Machine

## Conception de sites Web

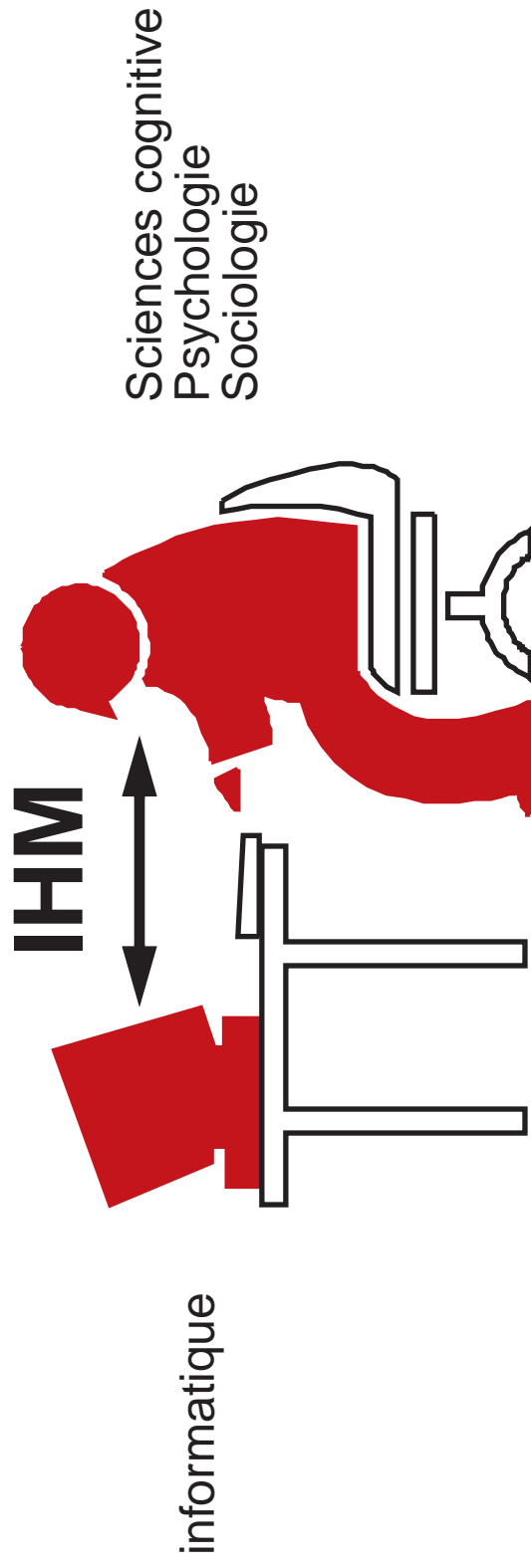
ENSIMAG (2ème année), Grenoble  
François Bérard



# Introduction

## QU'EST-CE QUE L'INTERACTION HOMME-MACHINE?

Discipline qui concerne l'étude des échanges entre les machines (ordinateurs) et leurs utilisateurs.

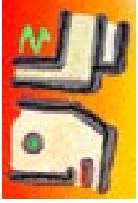




## **Que fait-on en IHM?**

- Observation du comportement humain
- Création de modèles du comportement humain
- Conception de systèmes adaptés au comportement humain

...afin d'améliorer l'interaction.



## Que cherche-t-on à améliorer?

- Utilité**
- Le champ d'action**  
*Permettre à l'utilisateur de faire ce qu'il a besoin de faire... et même plus.*
  - La simplicité d'apprentissage**  
*Le temps mis par un novice pour comprendre le fonctionnement du système.*  
*Le pourcentage de services du système mise en oeuvre par un novice au bout d'un temps donné.*
- Utilisable**
- L'efficacité des utilisateur**  
*Le temps mis par l'utilisateur pour accomplir une tâche donnée.*
  - La sécurité des actions**  
*Limiter les erreurs, faire en sorte qu'elles ne portent pas à conséquences.*
- ...



## DIFFICULTÉ FONDAMENTALE

### **Problème purement informatique**

*Recherche dans 2 milliards de documents en moins d'une seconde.*

L'homme s'adapte aux capacités de l'ordinateur. L'ordinateur est:

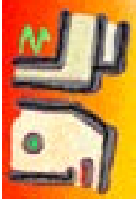
- déterministe,
- de faible complexité.

### **Problème d'IHM**

*Faire en sorte que tout utilisateur sache se servir du système en 5 minutes.*

La machine doit s'adapter au comportement de l'homme. L'homme est:

- non-déterministe,
- complexe,
- variable entre individus (tous les hommes sont différents),
- variable dans le temps (les individus évoluent).



## LIMITES DE L'IHM

**Y a-t-il des solutions aux problèmes d'IHM?**

Non:

il n'y a pas de solution systématique et complète.

**Est-ce que ça vaut la peine de s'en occuper ?**

Oui :

- des systèmes interactifs sont développés tous les jours,
- ils sont indispensables,
- il y en a de plus en plus.



## **RISQUES ASSOCIÉS À UNE MAUVAISE INTERFACE**

- Rejet pur et simple par les utilisateurs
- Frustration
- Coût d'apprentissage (formation, temps)
- Perte de productivité
- Utilisation incomplète
- Danger physique

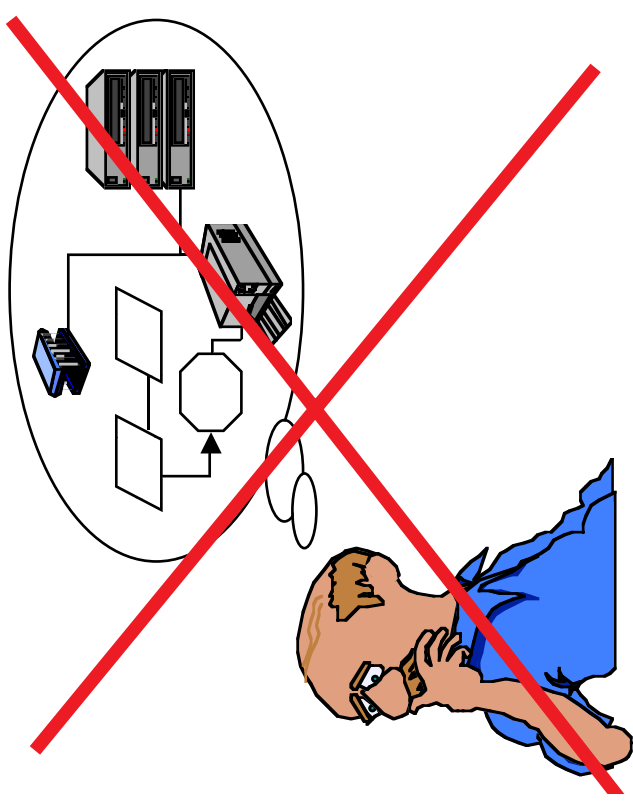
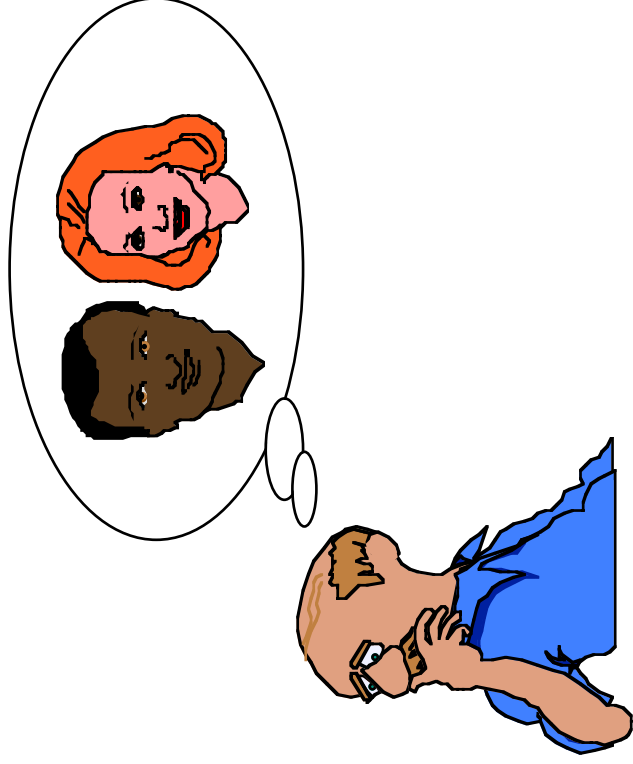
**Comment faire ?**



## OBJECTIFS DU COURS

# 1 Conception centrée sur l'utilisateur

▷ approche “top-down”







## OBJECTIFS DU COURS

### 2 Méthode de conception rationnelle

- Cycle de développement
  - ◇ Analyse
  - ◇ Conception
  - ◇ Réalisation
  - ◇ Évaluation
- Itérations des cycles de développement
- Mise en oeuvre de règles empiriques
- Traçabilité des choix



## OBJECTIFS DU COURS

### 3 Connaissances

- Modèles de l'utilisateur, de l'activité
- Règles ergonomiques
- User Action Notation (UAN)
- Modèle d'architecture
- Outils (TclTk / HTML / php...)



## **Cas particulier de la conception de sites Web**

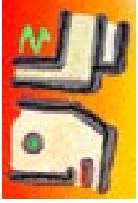
### **Motivation**

Plus de 38 million de sites en janvier 2001.  
(source Netcraft <http://www.netcraft.com/>)

Janvier 2001	27 M.
--------------	-------

Croissance exponentiel des sites et des applications.

Accès télématique universel.



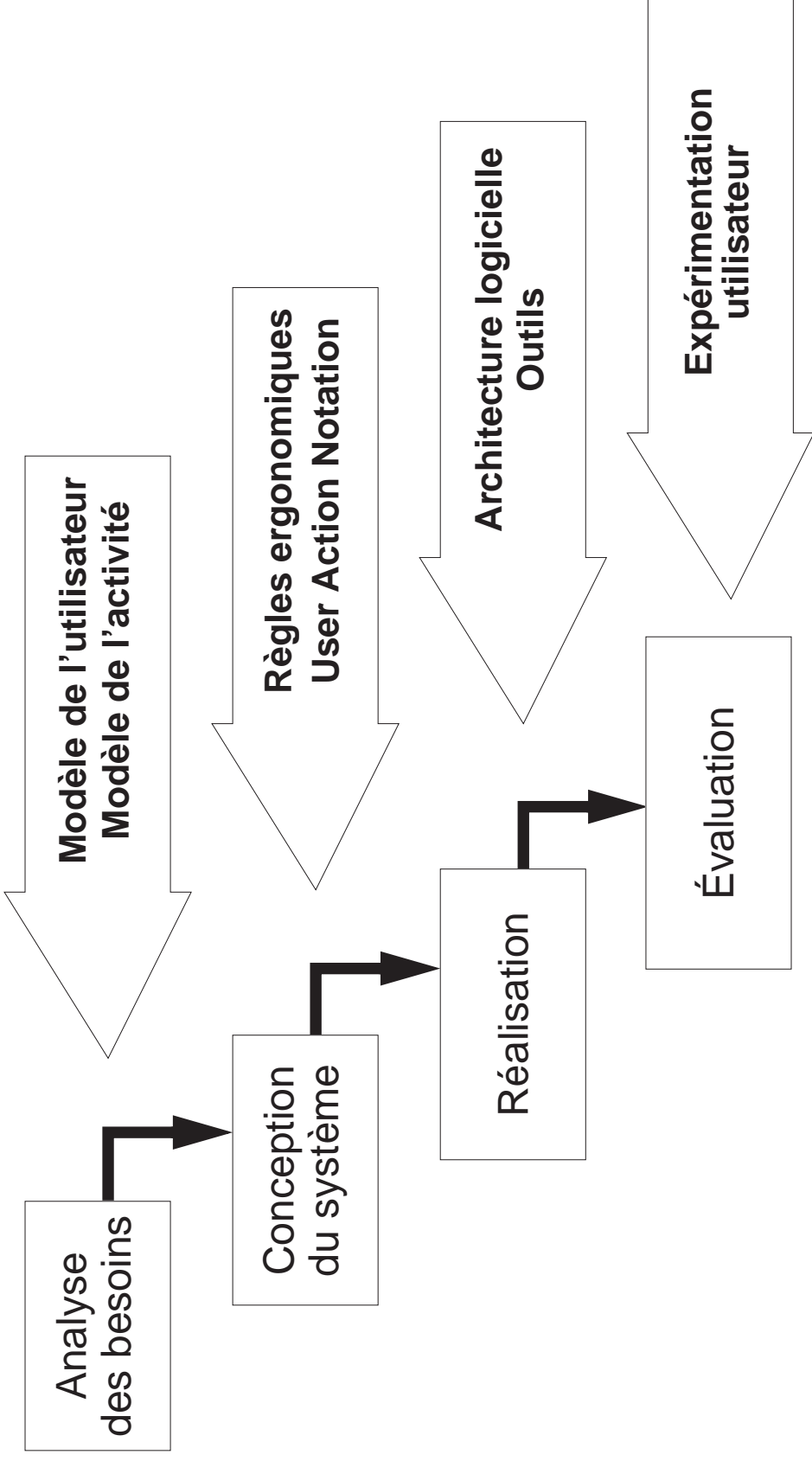
# SURVOL HISTORIQUE

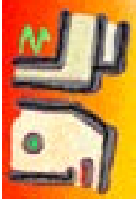
<b>Pré-histoire</b>	1946 ENIAC
<b>Clavier / langages</b>	1953 Fortran 1977 Apple 1
<b>Métaphore graphique / manipulation directe</b>	1968 <a href="#">Démo Doug Engelbart</a> 1980 Xerox STAR / Alto 1984 Apple Macintosh
<b>Futur?!</b>	1966 Réalité Virtuelle (Ivan Sutherland) 1980 Interfaces multimodales (Put That There, MIT MediaLab) 1993 Réalité Augmentée (P. Wellner)  Tangible Interfaces, Mobile / Ubiquitous computing...



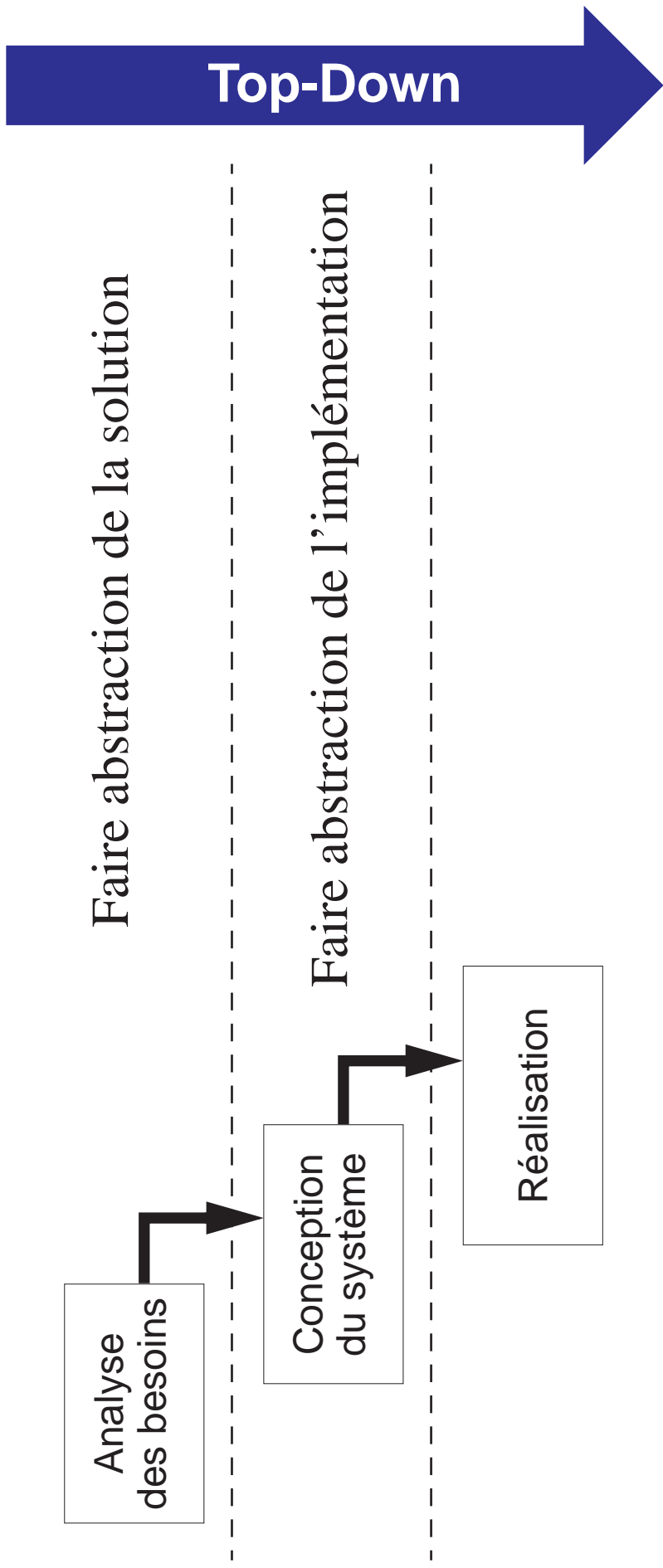
# Plan du cours

(Modèles cognitifs des utilisateurs)





# Plan du cours



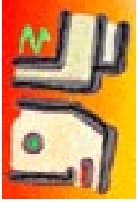


# Modèles cognitifs des utilisateurs

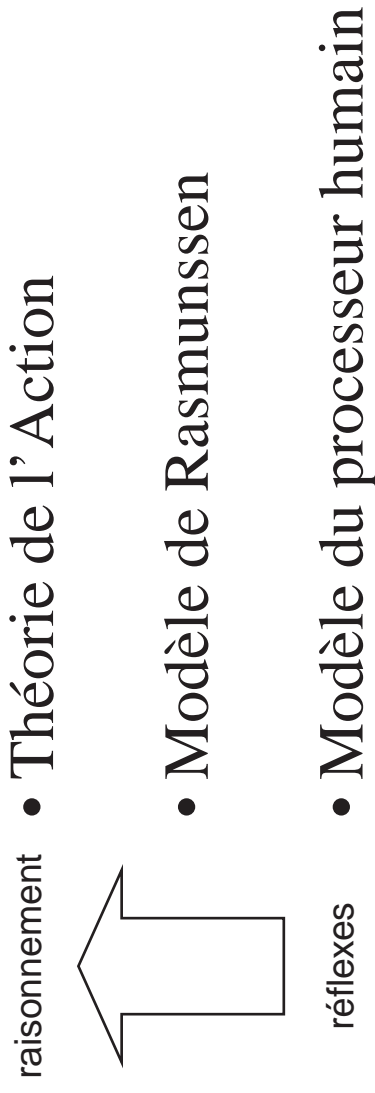
Relève de la psychologie cognitive. Nombreux modèles, pas d'unification.

Les modèles ne pourront **pas servir de preuve** (en règle générale c'est un problème trop difficile).

Les modèles doivent avoir un **pouvoir expressif**, et être les **supports du raisonnement**.



## Différents niveaux d'abstraction :







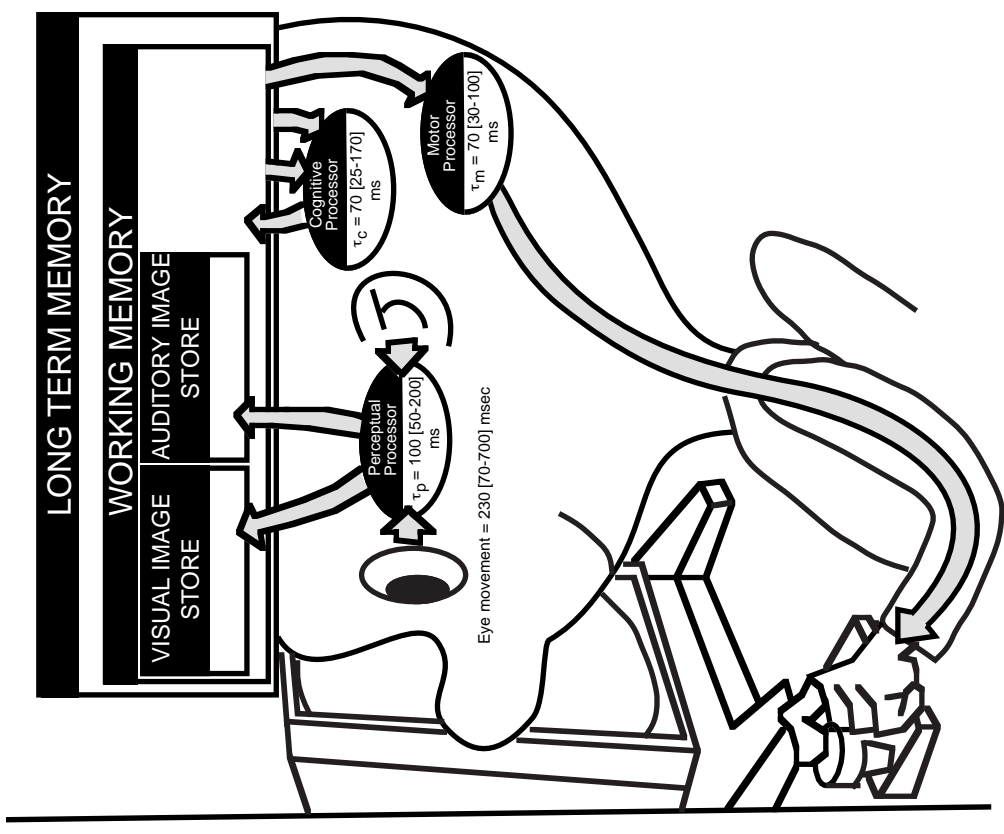
## MODÈLE DU PROCESSEUR HUMAIN

Card, Moran, Newel

“*The Psychology of Human-Computer Interaction*”

*Lawrence Erlbaum Associates, 1983*

- Bas niveau d’abstraction
- Estimations quantitatives
- Vérifications expérimentales aisées
- Orienté efficacité





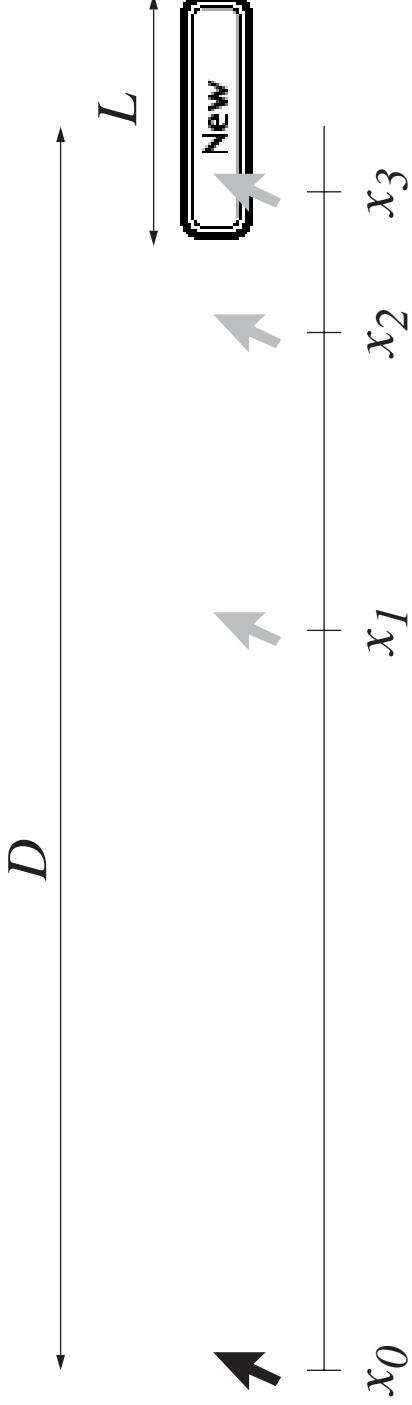
## Temps de cycle

	Processeur Perceptuel	Processeur Cognitif	Processeur Moteur
FastMan	50	25	30
MiddleMan	100	70	70
SlowMan	200	170	100

Capacité de la mémoire à court terme:  $7 \pm 2$  “chunk”



## Loi de Fitts



$$X_i / X_{i-1} = \varepsilon \quad \varepsilon < 1$$

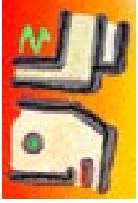
$$X_2 = \varepsilon X_1 = \varepsilon(\varepsilon D) = \varepsilon^2 D$$

$$X_n = \varepsilon^n D$$

$$\varepsilon^n D \leq 1/2 \cdot L$$

$$n = -\log_2(2D/L) / \log_2 \varepsilon$$

$$T = -(\tau_p + \tau_c + \tau_a) / \log_2 \varepsilon \cdot \log_2(2D/L)$$

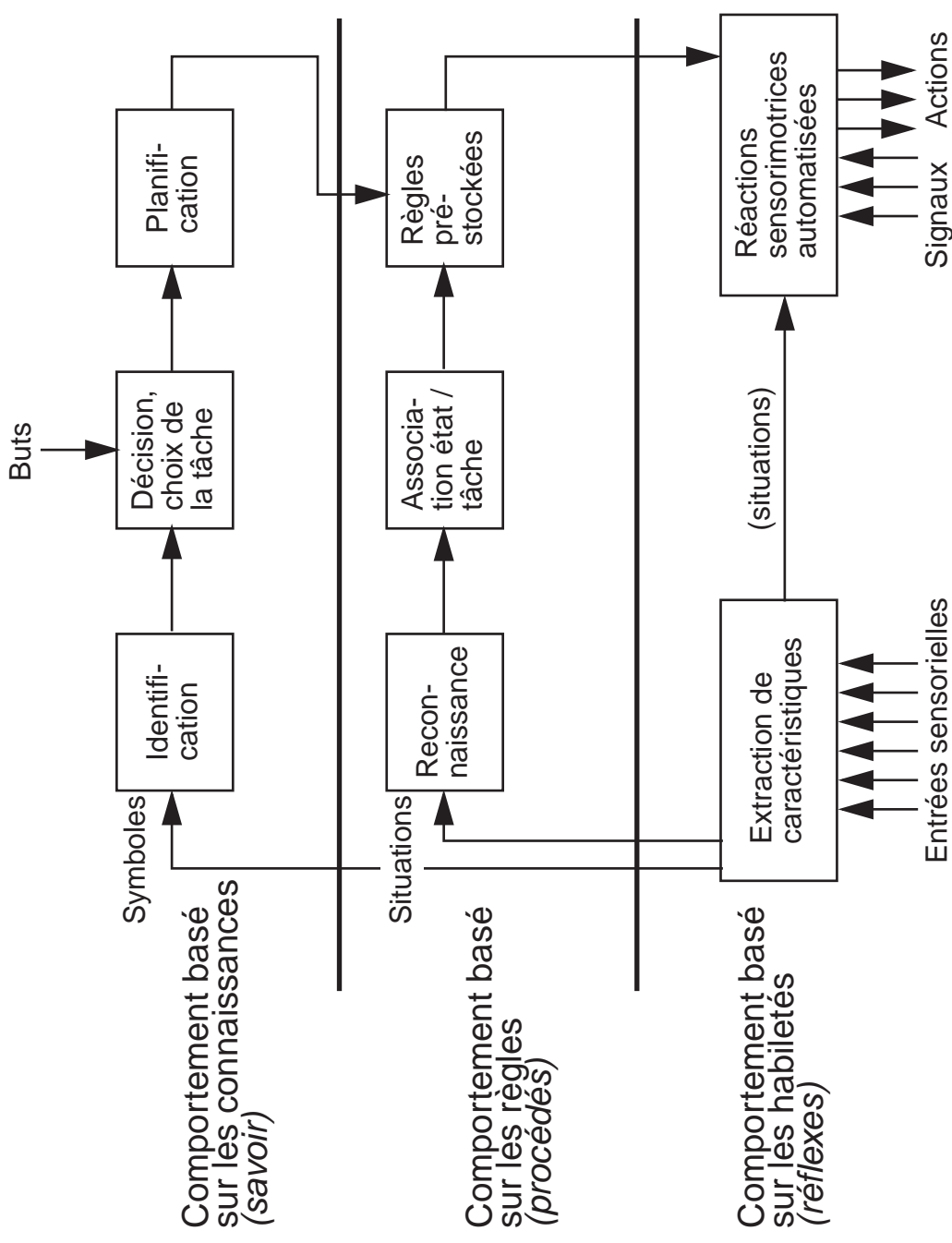


# MODÈLE DE RASMUNSEN

Rasmussen

*“Information Processing and Human-Machine Interaction, an Approach to Cognitive Engineering”  
North-Holland, 1986*

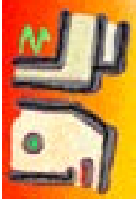
- Plusieurs niveaux d’abstraction
- Modèle très simplifié





## Exemples

- Déplacer un paragraphe d'un texte.
- Sélectionner un mot.
- Frappe de texte au clavier.



## THÉORIE DE L'ACTION DE D. NORMAN

D. Norman “*User centered design*”, 1986

- L'utilisateur modélise le monde en termes de variables psychologiques  $\psi$
- Le monde réel se manifeste en termes de variables physiques  $\varphi$

L'image (l'IHM) est le pont entre le monde psychologique le monde physique.



## **Exemple : Remplir une baignoire**

Deux robinets indépendants : eau chaude - eau froide.

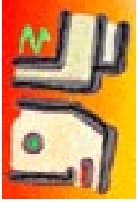
Variables **psychologiques** :

- *d* : débit de l'eau
- *t* : température du bain

Variables **physiques** :

- *dc, tc*: eau chaude (débit et température)
- *df, tf*: eau froide (débit et température)

Commandes physiques : les robinets liés à *dc* et *df*.



Relations entre les variables physiques et psychologiques

$$d = df + dc \quad (1)$$

$$t = \frac{(dc \cdot dt + dt \cdot df)}{(df + dc)} \quad (2)$$

**Inadéquation entre variables physiques et psychologiques**

- Comment augmenter la température du jet ?

**Mise en correspondance entre variables physiques et dispositif**

- Quel est le robinet d'eau chaude ?
- Dans quel sens tourner ?

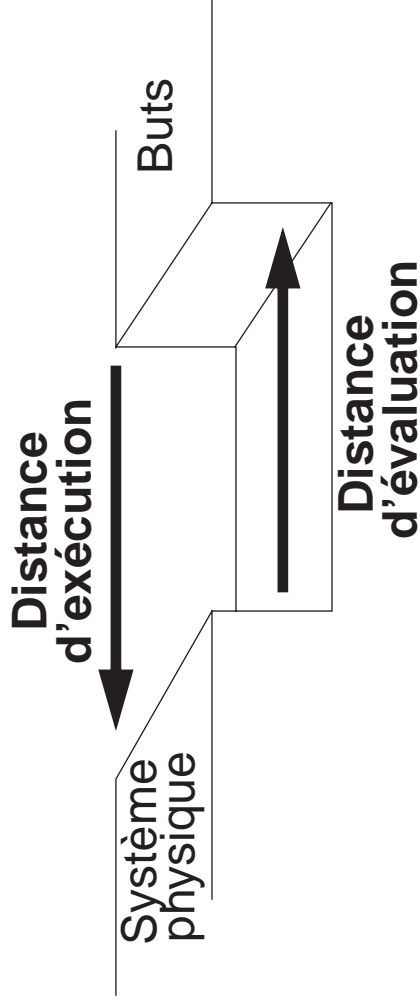
**Évaluation du résultat**

- Cas de deux robinets





## Distances d'exécution et d'évaluation

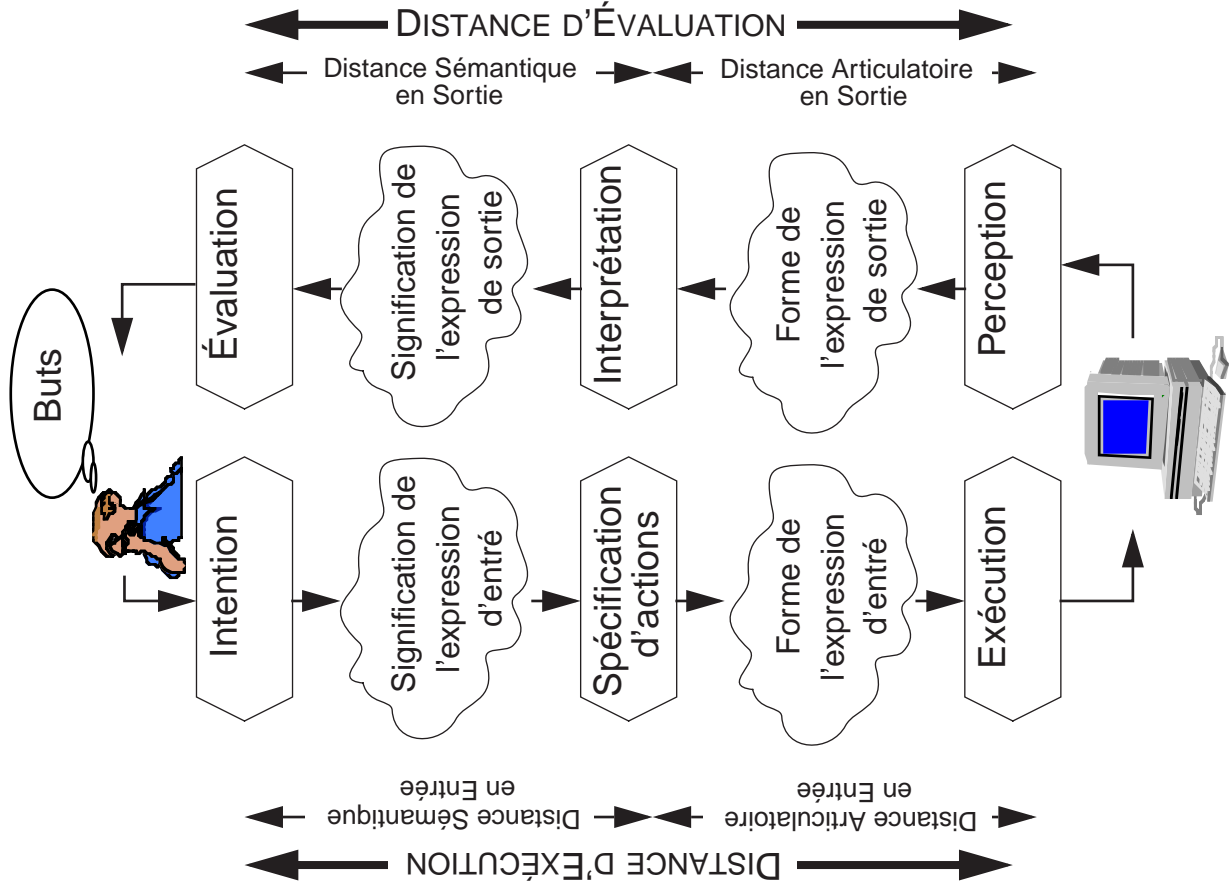


L'objectif du concepteur et du réalisateur :  
réduire les distances mentales par le biais de l'image  
du système.



## MODÈLES COGNITIFS DES UTILISATEURS

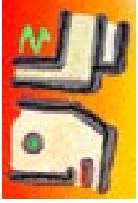
## Théorie de l'action de D. Norman Distances d'exécution et d'évaluation



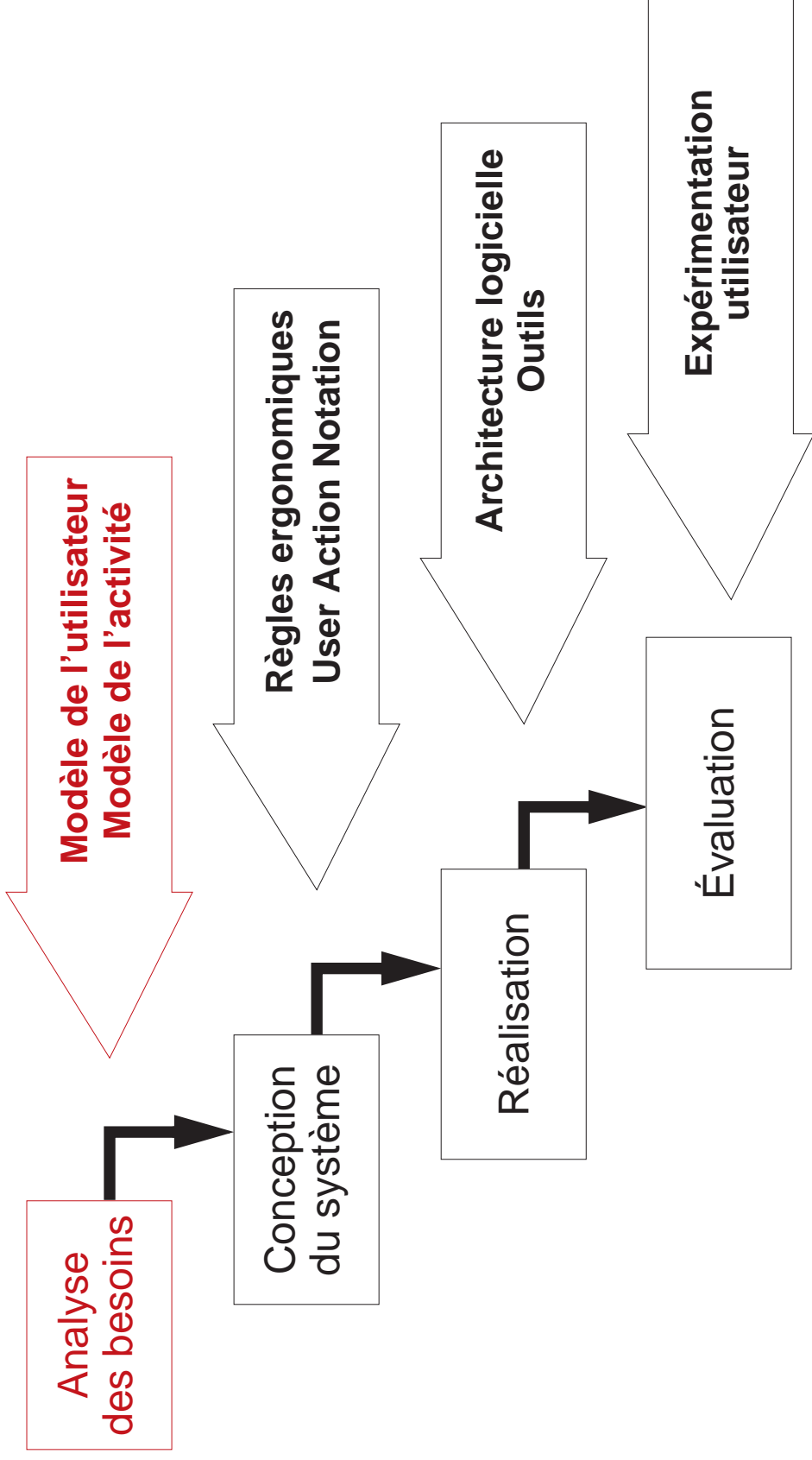


## Exemples

- Le voyant d'activité du four est rouge, l'utilisateur interprète "chaud" alors que le système signifiait "arrêt".
- La souris est ronde, la contact tactile ne permet pas de déterminer son orientation.
- Pour faire défiler le texte vers le haut, je clique sur la flèche vers le haut.
- Le message d'erreur s'affiche trop rapidement pour être lu par l'utilisateur.



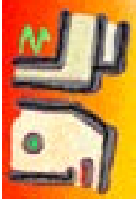
# Plan du cours





# Analyse des besoins

- Définition de la **raison d'être** du système.
- Modélisation du problème
  - ◇ modèle de l'utilisateur.
  - ◇ modèle de tâche



## VISER JUSTE

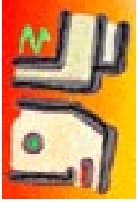
- Ne pas viser trop court (oublier une catégorie d'utilisateur / oublier des activités nécessaires).
  - ▷ dégradation du caractère utile du système.
- Ne pas viser trop long (public trop large, réalisation de nombreux services inutiles):
  - ◇ la complexité du problème de conception n'est sans doute pas linéaire en fonction des paramètres, elle est plutôt exponentielle,
  - ◇ gaspillage d'efforts qui auraient du servir à améliorer l'utilisabilité du système.
- ▷ dégradation du caractère utilisable du système.



## RAISON D'ÊTRE DU SYSTÈME

1. Résumer en une phrase la motivation principale à l'origine du travail qui va être entrepris.
2. Détailler davantage.
  - ◇ les principales faiblesses des solutions existantes
  - ◇ objectifs (quantifiables si possible: critères de succès)

Cette partie de l'analyse est **fondamentale**: c'est sur elle que reposera les choix entre plusieurs solutions possibles.



## Exemple

### Cas de la borne d'achat de billets de chemin de fer.

Offrir les mêmes services qu'un guichetier de gare afin de réduire la queue au guichet, permettre l'achat de billets à toute heure.

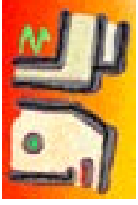
Le nouveau système à pour but principal de compenser les limites du système actuel:

- Délai trop long entre le moment où le client entre dans la gare et où il est en possession de son billet ("temps d'acquisition"). Deux causes pressenties: le temps de service d'un client est trop long, la queue au guichet.
- Impossibilité d'acquérir un billet en dehors des heures d'ouverture des guichets.

Critères de succès:

- Faire en sorte que le temps d'acquisition ne dépasse pas 15 minutes dans 90% des cas.
- Faire en sorte que la vente de billet soit accessible 24 heures sur 24, 7 jours sur 7.





# MODÈLE DE L'UTILISATEUR

## Classification générale des Utilisateurs

Type d'utilisateur	Connaissances
Novice	Faits simples Résous un problème en décomposant le système indépendamment du contexte
Débutant avancé	Peut anticiper, mémorise des situations, répète une liste d'action simples sans la comprendre
Compétent	Élabore des stratégies
Efficace	Sait quelle stratégie employer
Expert	Agit inconsciemment et globalement



## **Éléments de catégorisation d'un groupe d'utilisateurs**

- Ordre d'importance,
- Compétence (cf Rasmussen),
- Milieu socioculturel: nationalité, niveau de formation, géographie,
- Dans le domaine d'information du site,
- En informatique (connaissance du Web, utilisation de stations de travail),
- Biométrie: taille, âge, sexe, déficiences (ex: 8% des hommes sont daltoniens),
- Fréquence envisagée d'utilisation du site: fréquent, occasionnel, rare.



## Exemple

### Cas de la borne d'achat de billets de chemin de fer.

Niveau de connaissances générales

L'utilisateur sait lire et écrire dans une des langues suivantes: (...)

Niveau d'expertise dans le domaine: faible.

L'utilisateur connaît au minimum le vocabulaire et les concepts suivants: gare, train, horaires, correspondance, réservation (...)

Niveau d'expertise en manipulation de système informatiques: rudimentaire (Novice)

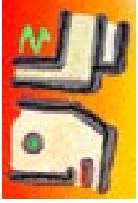
Tout utilisateur qui n'est pas effrayé par la manipulation d'un système informatique doit pouvoir utiliser le système. Faible priorité aux utilisateurs de niveaux d'expertise supérieurs.  
L'utilisateur

Capacités physiques

L'utilisateur à une hauteur du regard compris entre x et y.

Cas des personnes présentant un handicap physique: le système doit pouvoir être manipulé d'une seule main, (...).

(...)



## MODÈLE DE L'ACTIVITÉ

Objectif : assurer le caractère **utile** du système.

S'assurer que l'utilisateur pourra mener toutes les activités qu'il a besoin de mener un fois le système mis en production.

### **Choix d'un vocabulaire: différence entre "tâche" et "activité"**

Dans ce cours: l'**activité** regroupe les actions que font les personnes à qui est destiné le système avant qu'il ne soit question de ce système.

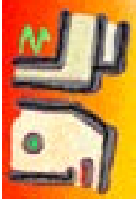
Les **tâches** regroupent les actions que feront les utilisateurs une fois le système mis en production.

Dans la phase d'analyse, c'est l'**activité** qui nous intéresse. Les **tâches** seront décrites dans la phase de conception.



## Comment s'y prendre ?

- Sondages exploratoires
- Entretiens - Focus group
- Observations in situ
- Construction de scénarios
- Exploration de l'existant



## Sondage exploratoire

Les réponses doivent provenir d'utilisateurs représentatifs

Diriger les questions sur des thèmes précis. Ne pas s'attendre à ce que les sondés soient capable de réponses globales (i.e. de faire votre travail).

### Exemple de sondage exploratoire (cas de la borne de gare)

1. Qu'avez vous demandé lors de votre dernier passage à un guichet ?
2. Qu'est-ce qui vous a le plus embêté lors de votre dernière acquisition d'un billet de train ?
3. Qu'est-ce que vous aimeriez pouvoir faire au guichet que vous ne pouvez pas faire actuellement ?
4. Quelles sont vos suggestions pour la création d'un guichet automatique ?  
(laisser la porte ouverte...)



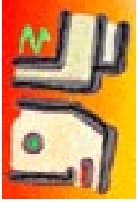
## Entretiens - Focus group

Nécessite:

- Un modérateur de métier qui établit un agenda
- Un petit groupe de concepteurs et d'utilisateurs (moins de 10 personnes)

Sessions enregistrées sur support vidéo

Grande quantité d'informations qualitatives



## **Observations in situ**

Observations de sujets membres de la population cible, dans leur “milieu naturel”, en train d’effectuer l’activité cible.

Nécessité d’enregistrer les observations (films / prise de notes).





## Construction de scénarios

La technique la plus efficace, mais nécessite des utilisateurs représentatifs.

Tend à placer les utilisateurs dans leur contexte

- plus créatifs
- expriment mieux leurs requis qu'avec les autres approches.

Les utilisateurs écrivent leur réponse: permet au concepteur d'identifier le bon vocabulaire.

Durée: 30 à 60 mn

Si difficulté d'identifier les activités fondamentales / secondaires: un questionnaire avec classement (échelle de 1 à 10)



## Exemple

Cas de la borne d'achat de billets de chemin de fer.

### Scénario 1:

Alexandre, étudiant à Grenoble, souhaite rendre visite à Jérôme qui habite Lyon. On lui a dit qu'il y a un train sur ce trajet à 17h15. Ça lui laisse le temps d'arriver à la gare avec 1/2 heure d'avance pour acheter son billet. Il se présente au guichet. Le guichetier lui demande s'il souhaite aller à Lyon Perrache ou Lyon Part Dieu. Alexandre ne sais pas laquelle des deux gares est plus proche de l'appartement de Jérôme. Il choisit Part Dieu car c'est le plus rapide.

### Scénario 2:

Marie est accompagnée en voiture à la gare de Grenoble. Elle a une réservation pour un TGV à destination de Paris. Malheureusement, elle arrive en retard à cause du trafic automobile. Elle souhaite savoir si elle peut changer son billet, ce qui lui en coûtera, et quel est le prochain TGV pour Paris.

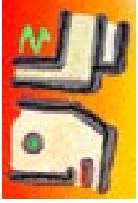


### Scénario 3:

Virginie cours pour attraper son train de 18h02. Elle arrive en retard à la gare... mais par chance son train a également du retard, il n'est pas encore parti. Elle se précipite sur une borne pour acheter un billet. Malheureusement, le système ne veut pas lui vendre un billet pour un train qui est (en théorie) déjà parti.

### Scénario 4:

Paul souhaite se rendre en train à Orange. Il demande quel est le premier train en partance pour Orange. Il y en a un qui part dans 30 minutes mais dont le trajet dure 2h50... c'est beaucoup. Paul aimerait savoir s'il n'y a pas un autre train qui mettrait moins de temps, quitte à flâner un peu en l'attendant. Il y en a un, deux heures plus tard, qui ne met que 2h pour atteindre Orange... Paul hésite. Il vérifie qu'il peut avoir une place "non fumeur" sur les deux trains. C'est le cas. Paul hésite encore. Il regarde s'il y a un train rapide qui peut l'amener à Avignon, où ses parents pourraient venir le chercher. C'est le cas. Dans combien de temps part ce train ? Aura-t-il le temps d'appeler ses parents avant de le prendre ?



## **Exploration de l'existant**

N'est pas toujours possible (y a-t-il un existant ?)

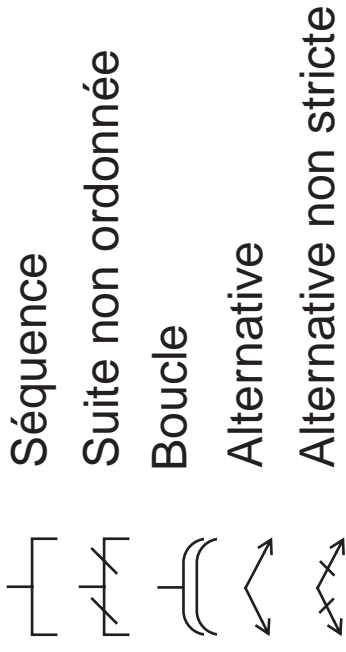
Faire attention à ne pas se laisser entièrement influencer par les solutions des autres.

▷ Avoir effectué et documenté sa propre analyse avant.

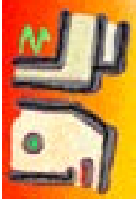


## Organisation : l'arbre de tâche / d'activité

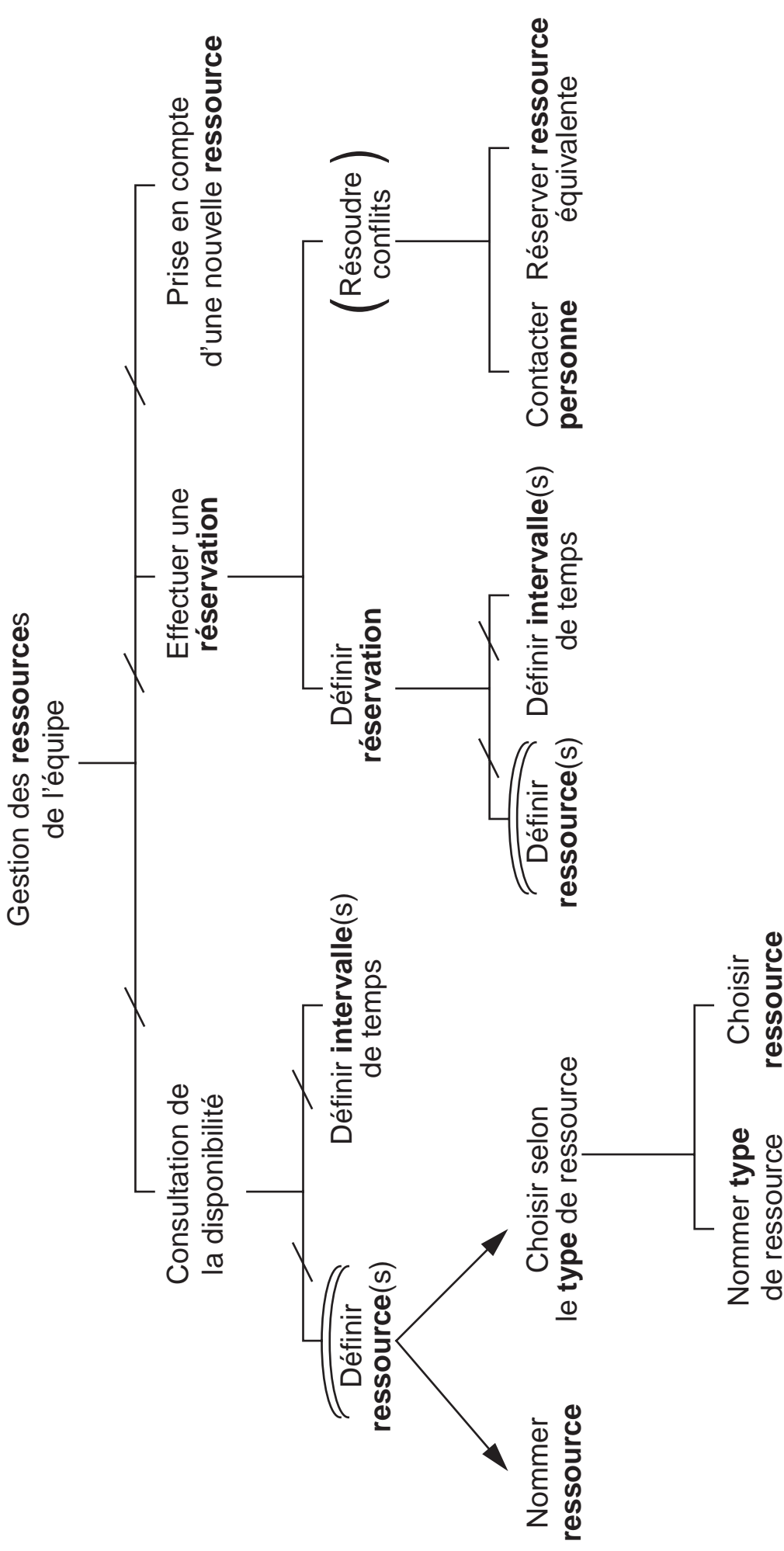
Décomposition hiérarchique en tâches / sous-tâches



Met en avant les concepts du domaines (variables psychologiques)



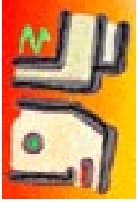
# Arbre d'activité de la gestion des ressources communes



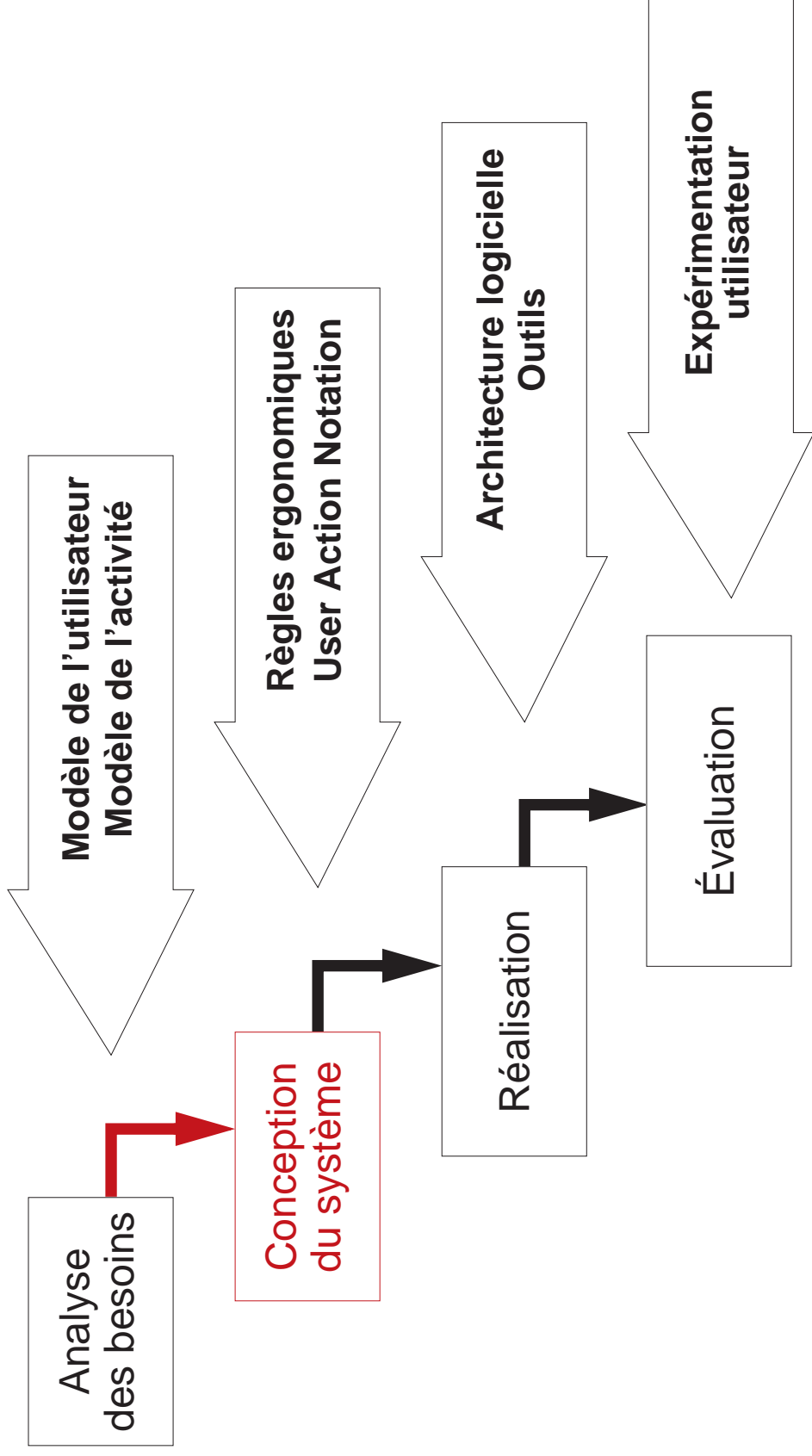


## Décorations des noeuds

- concepts du domaine = objets référencés (lettre, enveloppe, archives),
- préconditions / postconditions (but),
- fréquence,
- complexité,
- niveau de priorité (indispensable / nécessaire / optionnel),
- criticité (niveau de danger, caractère irrévocable),
- contraintes temporelles (durée maximale),
- autres (selon le domaine).



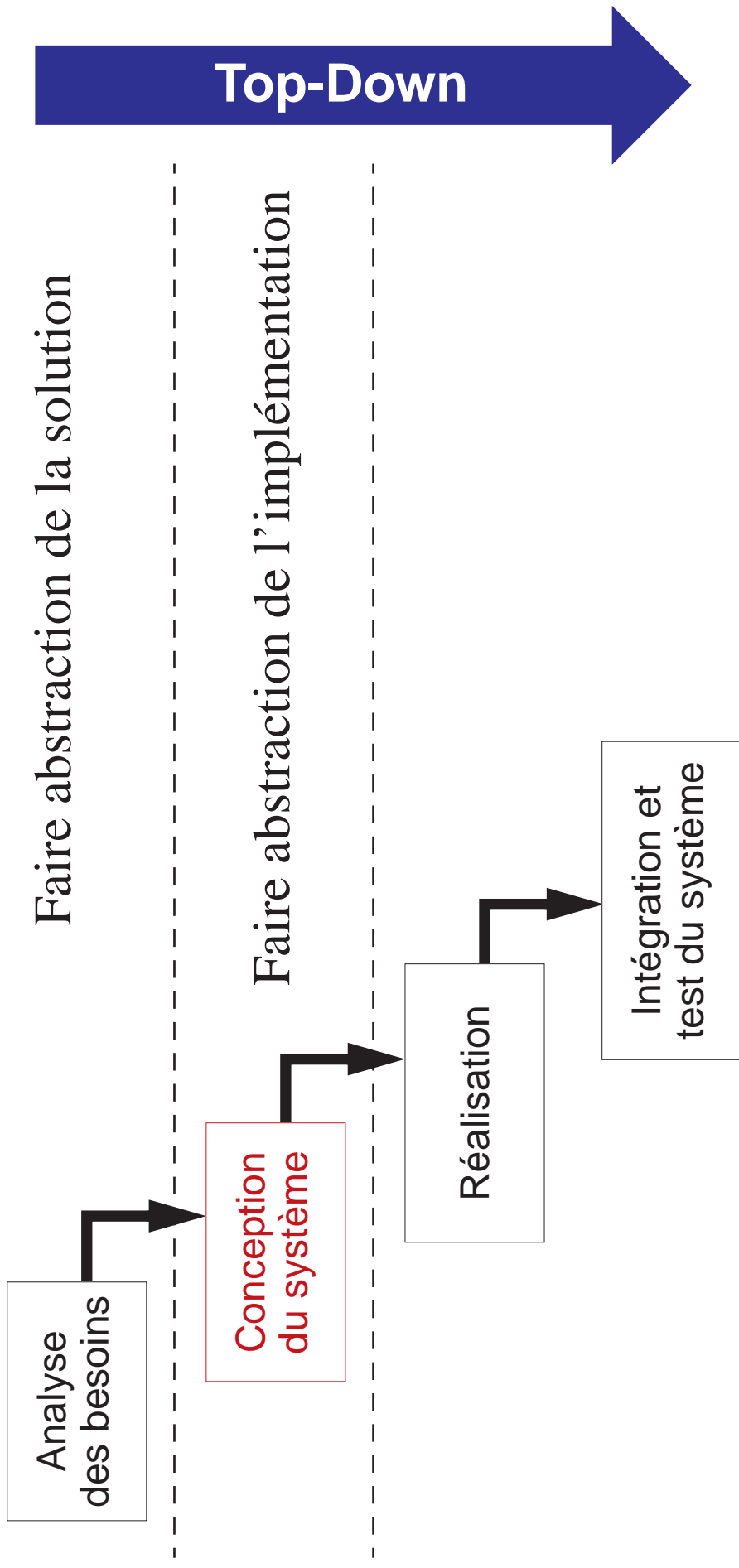
# Plan du cours







# Plan du cours





# Conception du système

Propose une “solution” aux problèmes identifiés dans la phase d’analyse.

Il n’existe pas de règle automatique: pour tel problème appliquer telle solution (émergence cependant des “patrons de conceptions”).

Il faut à la fois être créatif et s’appuyer sur l’expérience.

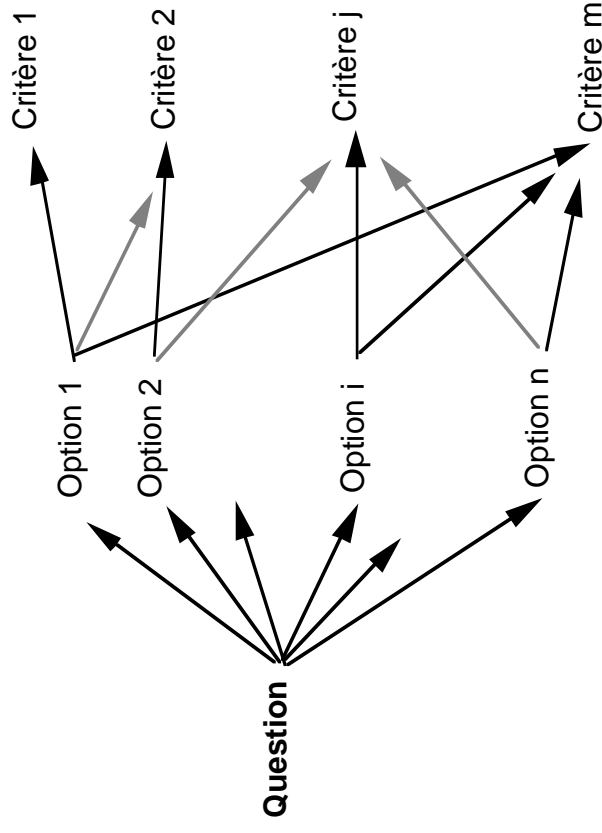
Approche “top-down”: les choix de conceptions sont **toujours** dirigés par l’intérêt de l’utilisateur (s’appuyer sur la phase d’analyse).

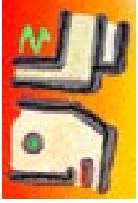


# DOCUMENTATION DES CHOIX

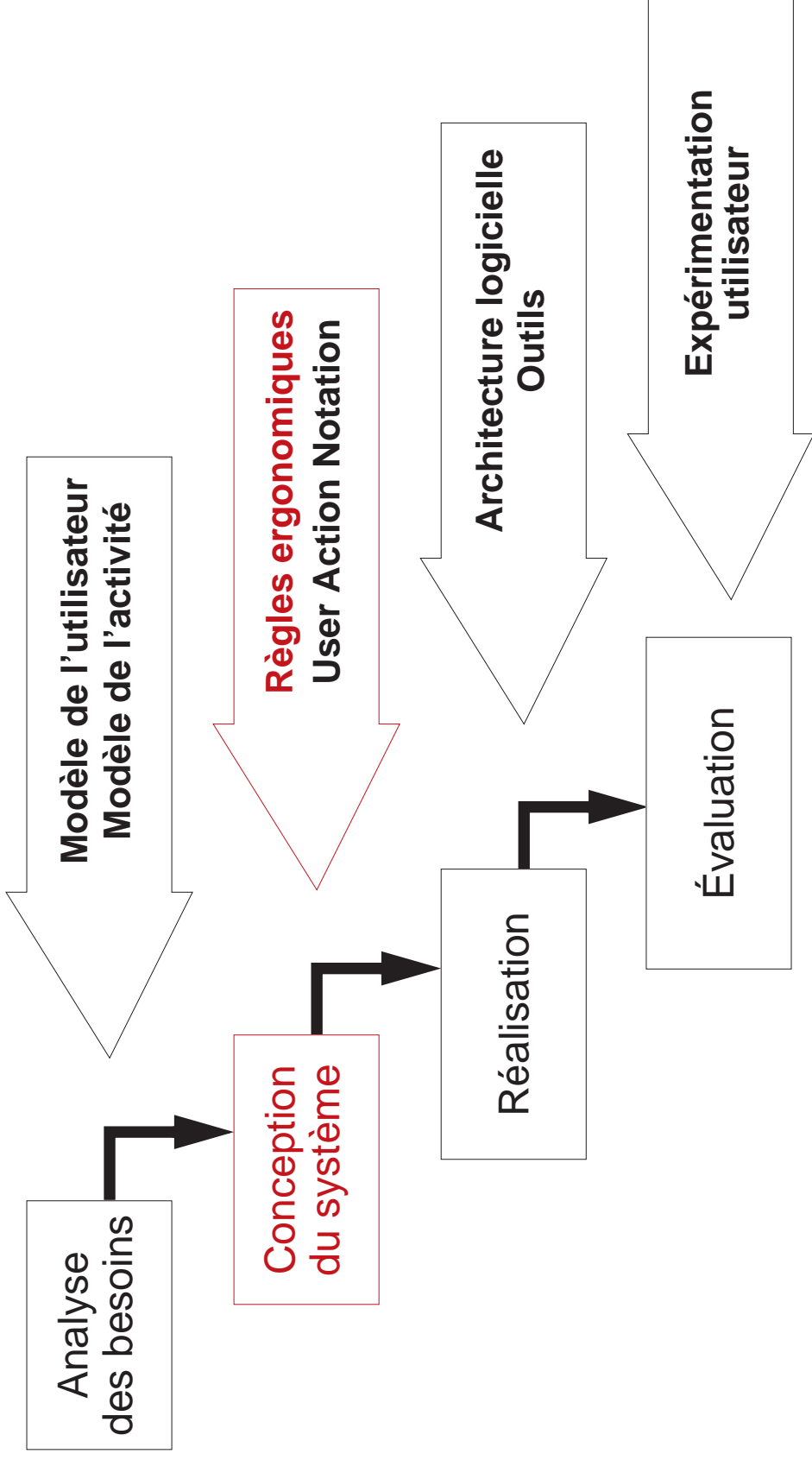
## Notation QOC (McLean)

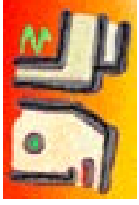
- traçabilité
- aide à la prise de décision (usage éventuel de poids)





# Plan du cours





# RÈGLES GÉNÉRALES D'ERGONOMIE DES SYSTÈMES INTERACTIFS

Sources:

- Ben Schneiderman
- Dominique Scapin
- Joëlle Coutaz



## Manipulation directe, métaphores

Donner l'impression à l'utilisateur qu'il "fait" les choses, plutôt que les "faire faire" à l'ordinateur.

- ▷ création d'un monde (le bureau) qui contient des objets et des lois.
- ▷ choix de représentations métaphoriques pour les objets, afin de faciliter la compréhension de ce monde.

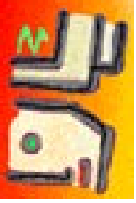
- Au lieu de créer un commande par action, on crée un ensemble minimal d'actions dont la combinaison permet de réaliser toutes les actions.
- L'apprentissage est facilité par les métaphores.



Langage de commandes



Manipulation directe

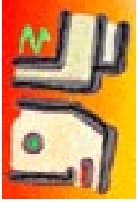


## Caractéristiques essentielles de l'interaction à manipulation directe

Schneiderman, B. (1983).

“*Direct Manipulation: a step beyond programming languages*”. *IEEE Computer*, 16(8), 57-69.

- représentation continue des objets d'intérêt,
- actions physiques au lieu d'une syntaxe complexe,
- opérations rapides, incrémentales, réversibles, dont l'impact sur les objets d'intérêt est visible immédiatement.



## Bénéfices

(Schneiderman)

1. Les novices peuvent apprendre les fonctions de bases rapidement, souvent grâce à une démonstration d'un utilisateur plus expérimenté.
2. Les experts peuvent travailler de façon extrêmement rapide pour exécuter un nombre varié de tâches, et même définir de nouvelles fonctions.
3. Les utilisateur occasionnels retiennent les concepts opérationnels.
4. Les messages d'erreurs sont rarement nécessaires.
5. Un utilisateur voit immédiatement si ses actions l'approche de son but, et si ce n'est pas le cas, il peut simplement changer son approche.
6. L'anxiété des utilisateurs est moindre car le fonctionnement du système est compris et les actions sont facilement réversibles.





## Modes

Un mode correspond à un “état” particulier du système.

À chaque mode correspond un sous ensemble des actions de l'utilisateur autorisées.

*Exemple: Borne publique, boîte de dialogue modales, assistants, outil courant de l'éditeur de dessin.*

En règle générale, il faut éviter les modes : ce sont souvent des restrictions arbitraires du champ d'action de l'utilisateur.



## Cas particuliers d'usage acceptable des modes

- modes maintenus (appui continu d'une touche spéciale)
- métaphore du monde réel (outils de l'éditeur de dessin)
- débutants (borne, assistant)
- danger

Dans ce cas, faire en sorte que :

- le mode courant soit clairement perceptible,
- il soit facile de passer d'un mode à l'autre.



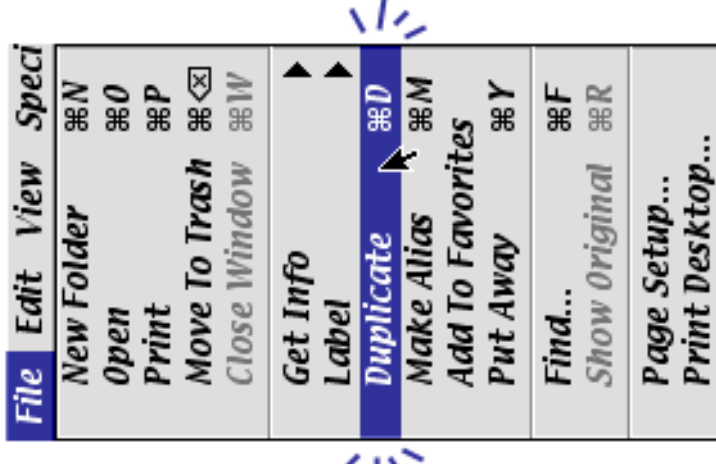
## Services généraux

- Aide
  - ◇ factuelle
  - ◇ procédurale
  - ◇ contextuelle
- Défaire-refaire (encourage l'exploration)  
Si pas possible: prévenir l'utilisateur.
- Valeurs par défaut (performances)
  - ◇ statiques mais ajustables
  - ◇ dynamiques (dernière valeur utilisée)
- Personnalisation (préférences)

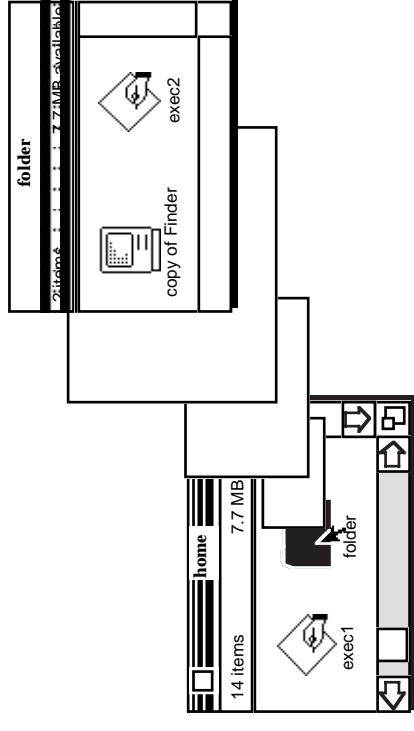


## Retour d'information (feedback)

- Immédiat



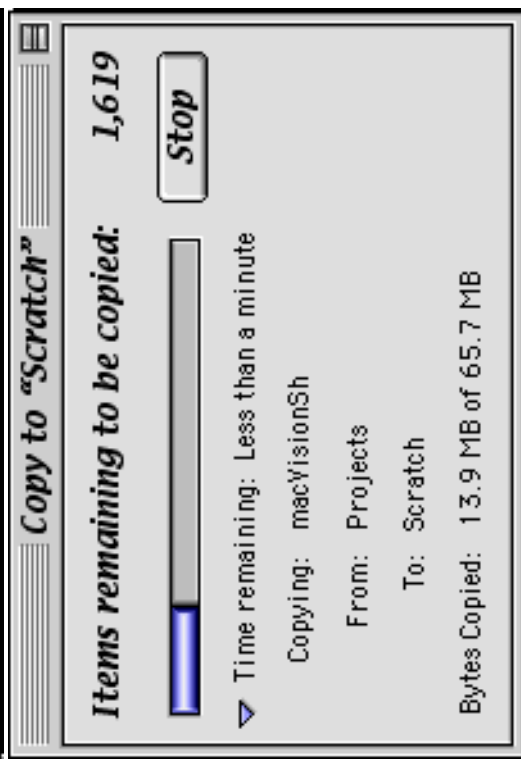
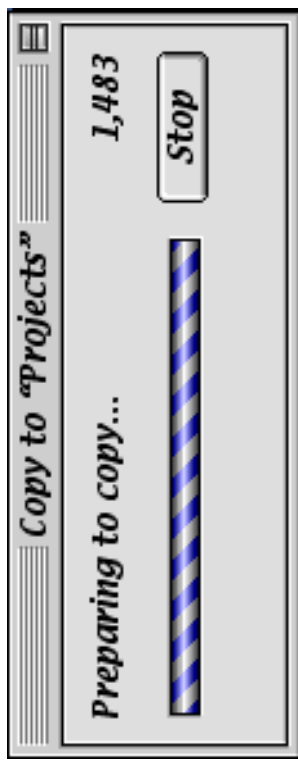
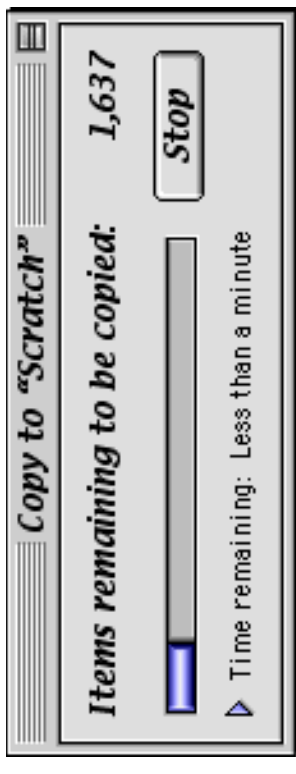
- Informatif

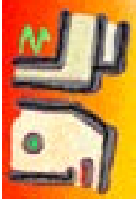




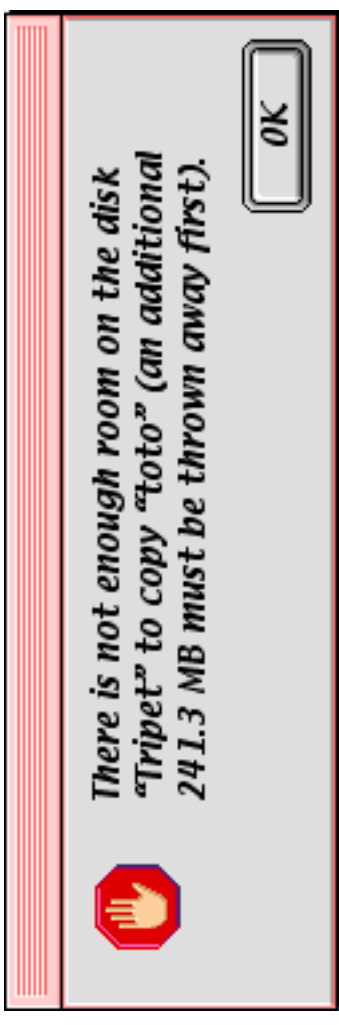
- Processus long

- Honnêteté



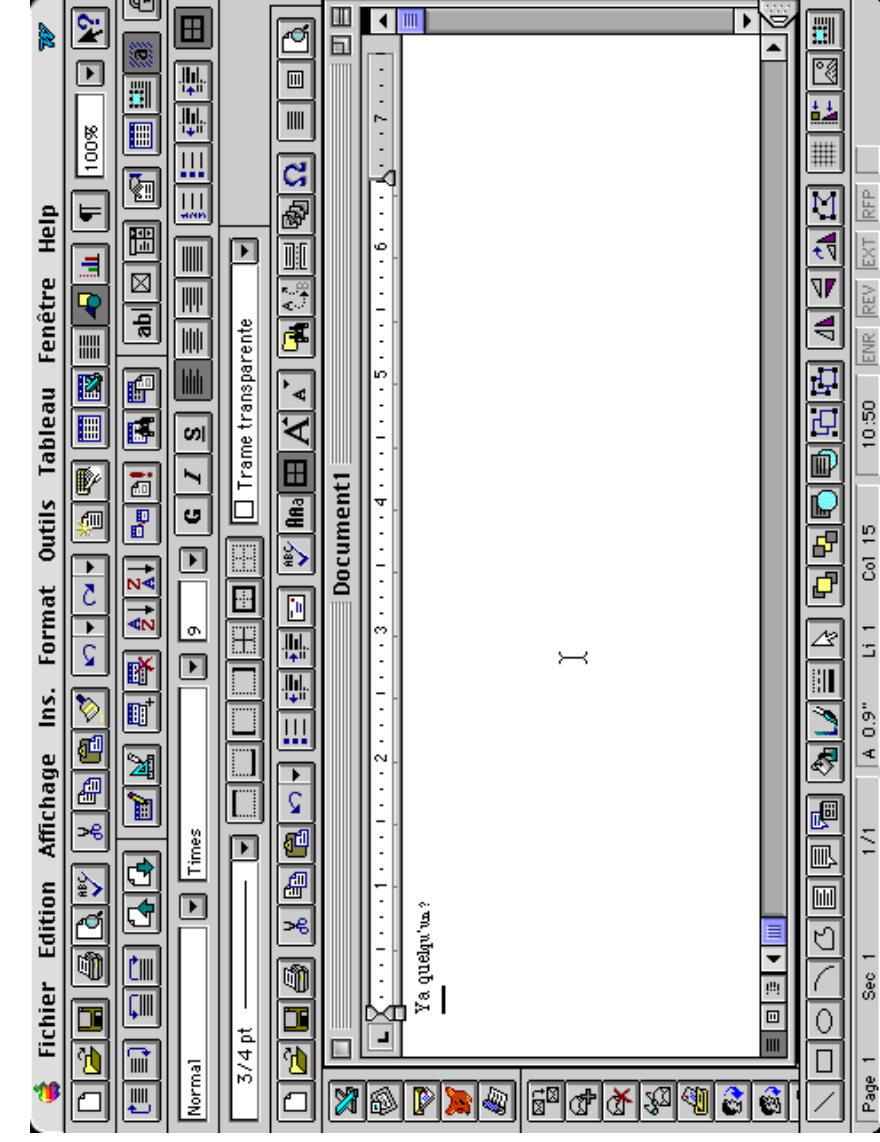


- Correction d'erreur





**Veiller à la surcharge :**  
**Ni + Ni -**



**EXCELLENCE**  
**GRAPHIQUE**  
**(TUFTÉ) :**

Exprimer le *maximum*  
d'information avec le  
*minimum* d'encre sur le  
minimum de surface.

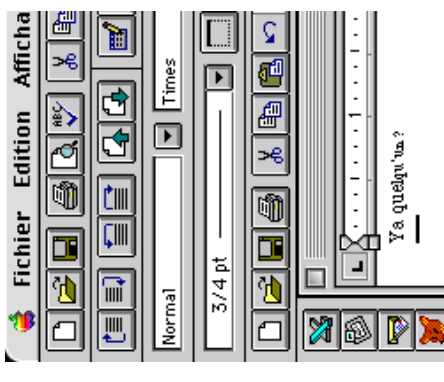


## Veiller à la surcharge

- Excellence graphique (Tuftte) :

Exprimer le maximum d'information avec le minimum d'encre sur le minimum de surface.

- Ni + Ni -







## Affordance, Stabilité, Concision

### Affordance

la représentation engendre la bonne intuition

### Stabilité de l'écran

exemple du retour d'information pro-actif

### Concision de l'interface

- trajectoire d'interaction
- raccourcis clavier
- valeurs par défaut



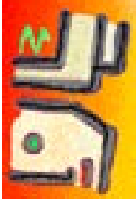
## Cohérence de l'interface

### Cohérence Interne

- Syntaxique
  - ◇ Majuscules
- Lexicale
  - ◇ Terminologie du métier
  - ◇ Congruence (Agrandir, Diminuer -> Agrandir, Réduire)
- Sémantique
  - ◇ Créer, création (verbes ou substantifs)

### Cohérence Externe

- Choix universellement appliqués (couleur des liens)



## Menus

### Composition

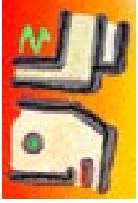
- Reflet de la décomposition en tâches et sous-tâches
- 7+-2 éléments
- Si nombreux éléments -> sous-rubrique

### Ordre des éléments

Fréquence, Logique, Alphabétique

### Libellés des commandes

Verbes uniquement ou substantifs uniquement (pas de mélange)



# RÈGLES GÉNÉRALES D'ERGONOMIE DES SITES WEB

## Sources:

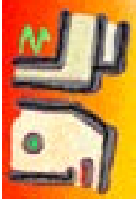
- Jakob Nielsen “Designing web usability”, New Riders Publishing
- <http://www.useit.com/> (Jakob Nielsen)
- <http://www.usableweb.com/>



## **Conception indépendante de la plate-forme d'exécution**

Différence fondamentale entre systèmes interactifs “classiques” et web: le concepteur du système n’a pas le contrôle de la plate-forme d’exécution.

Il existe de grandes disparités entre les plates-formes. La tendance est à l’accroissement des disparités.



## **Solution: séparer contenu et présentation**

C'était l'un des objectifs de HTML...

...mais HTML a été détourné par Netscape, Microsoft, etc...

Retour à la séparation entre contenu et présentation:

Cascading Style Sheets

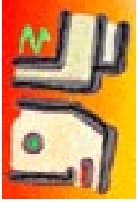
<http://www.w3.org/Style/CSS/>



## Surface de l'écran

C'est une ressource critique, en quantité variable.

	640x480	800x600	1024x768	>= 1280x1024
1997 (Horus & GVU)	22%	47%	25%	6%
1999 (Starmarket)	13%	55%	25%	2%
2002 (GeoStats) <a href="http://gheos.net/counter/stats.php?GheoStats&amp;report=screen_settings">http://gheos.net/counter/stats.php?GheoStats&amp;report=screen_settings</a>	1%	29%	57%	13%



## **Imposer un minimum de contraintes sur la plate-forme d'exécution**

- définition d'écran (160x120... 1600x1200),
- nombre de couleurs (noir et blanc, niveaux de gris, milliers, millions),
- dispositif interactif (clavier, stylo, doigt, voix)
- butineur supporté (Netscape, Internet Explorer, iCab...)
- technologie (Flash, CSS, Quicktime...)

**Vérifier le bon comportement des pages lors d'un redimensionnement de la fenêtre.**





## Temps de réponse

- 0,1 s. Maximum pour une “interaction fortement couplée”
- 1 s. Maximum pour ne pas interrompre le flux des idées.
- 10 s. Maximum pour garder l'utilisateur.

Octobre 1998: 70% de modems rapides (56 Kbit/s)

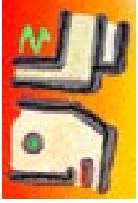


	Modem rapide	ADSL / Câble	Ether10
Débit théorique	56 Kbit/s 7 Ko/s	1 Mbit/s 125 Ko/s	10 Mbit/s 1.25 Mo/s
Débit observé	4 Ko/s	100 Ko/s	300 Ko/s
Temps de réponse pour 10 Ko	2.5 s	0.1 s	< 0.1 s
Taille pour 5 s	20 Ko	500 Ko	1.5 Mo



## **Afficher de l'information aussi vite que possible**

- Afficher l'information importante en haut de page
- Concevoir un code HTML de taille réduite
- Utiliser les attributs “width” et “height” des images  
Permet d'afficher les tables avant d'avoir chargé les images.
- Utiliser l'attribut “ALT” des images  
Permet de choisir un lien avant d'avoir chargé les images.



## Mise en page

### Frames

- Jakob Nielsen: “Just say no!”
- Les frames cassent l’unité de navigation: la page.
- On ne sais plus sur quel site on est (plusieurs sites dans une même page).
- À quelle frame correspond l’adresse dans le butineur?
- Comment revenir sur une “page”?
- Plusieurs barres de défilement: gaspillage de surface écran.
- Remplacer par des éléments communs à toutes les pages
  - ▷ un seul téléchargement nécessaire.



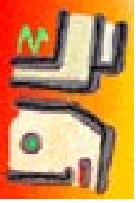
## Navigation

Répondre à 3 questions:

- D'où viens-je?
- Où suis-je?
- Où puis-je aller?

### D'où viens-je?

- Ne pas contrarier le bouton “retour arrière” (pas de re-direction)
- Choisir des bons titres de page pour l'historique
- Ne pas changer la couleur des liens  
facilite le repérage des pages déjà parcourue

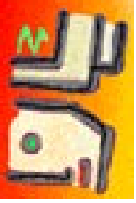


## Où suis-je ?

- Par rapport au web entier
  - ▷ Nom du site sur toutes les pages au même endroit.
- Par rapport au site
  - ▷ Affichage d'un plan du site permanent, **stable**.
  - ▷ Mise en évidence de l'emplacement courant.

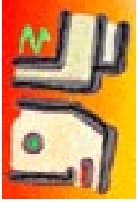
## Où puis-je aller ?

- Avoir pensé à la structure du site.
- Offrir en permanence un plan du site.
- Mise en évidence des liens.



## Dangers des éléments de navigation

- Gros consommateur de surface écran
  - ▷ choisir judicieusement.
- Préférer la barre de navigation horizontale en haut de page.
  - ▷ disponible immédiatement, défilement vertical plus aisé.
- Liste de liens entre crochets []  
Exemple:  
[[Page principale](#)][[Administration](#)][[Filières](#)]
- Tirer partie de la navigation implicite des boutons.  
(et surtout ne pas jouer contre elle)



## Contenu des pages

### Principes généraux

La lecture sur écran est plus difficile et fatigante que sur papier.

- Fonte simple “sans-serif” : helvetica, arial.
- Couleurs: privilégier le noir sur blanc.
- Réduire le volume de texte (50% de l'équivalent papier).
- Si on ne peut pas réduire, préférer un résumé et proposer un format pour l'impression.
- Utiliser un vocabulaire simple.





## **Principe de la pyramide inversée**

On commence par la conclusion, puis on ajoute du détail progressivement.

Au dernier niveau: un lien hypertext vers une autre page donnant tout le détail.

Principe: Le lecteur peut s'arrêter à tout moment, il aura lu le plus important.



## **Favoriser le survol**

- Soigner les titres: vocabulaire simple et informatifs.  
Pas “d’expression choc” qui n’informent pas sur le sens.  
Changer les (mauvais) titres des liens pointant vers les sites externes.
- Structurer les paragraphes en listes (numérotées ou à points).
- Utiliser des tables.
- Mettre en valeur les mots-clefs (gras, lien hypertext).

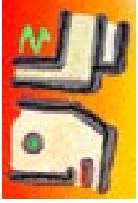


## Titre (HTML) des pages

Ils jouent le rôle de “clefs” pour les moteurs de recherche.

Ils deviennent l'intitulé des signets (bookmarks).

- Résumer le contenu de la page en trois mots.
  - Les titres de toutes les pages doivent être différents (problème de l'historique du parcours).
  - Supprimer le “liant” .
- “Bienvenue sur le site de l'UFRIMA” -> “UFRIMA - Accueil”



## Liens

### **Le lien est l'élément clef du succès du web.**

- Élément de navigation universel.
- Implémente l'idée que toutes les connaissances sont connectées.

### **Respect des principes fondamentaux:**

Ne pas modifier le “look and feel”.

Utiliser souvent, à bon escient.

### **Trois types de liens**

- Liens de navigation (structure du site)
- Liens pour avoir plus de détail
- Liens vers un sujet proche (voir aussi...)



## Intitulé

### Erreurs fréquentes

Pour voir les emplois du temps, [cliquez ici](#).

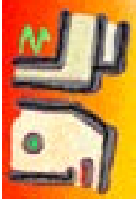
Préférer:

Les [emplois du temps](#).

[Le site contient les emplois du temps de toutes les filières](#).

Préférer:

Le site contient les [emplois du temps](#) de toutes les filières.



## Recommandations

- Toujours utiliser la même adresse dans les URL  
Mauvais exemple: <http://iihm.imag.fr> et <http://iihm.imag.fr/>  
▷ Le navigateur colore en mauve les liens déjà parcourus.
- Ne pas hésiter à faire les liens vers d'autres sites (liens sortants)  
Des liens bien choisis sont de grande valeur !  
Les utilisateurs peuvent **de toute façon** quitter votre site.
- Liens entrants.  
Bien choisir la structure du site dès le départ.  
Conserver les liens.
- Prévenir de la taille pour un téléchargement (durée ?).

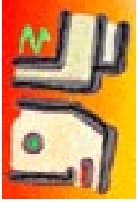


## Page d'accueil

Souvent la page la plus importante parce que la plus fréquemment visitée.

### **Doit contenir en général:**

- Une description succincte de l'objet du site.  
Pour les nouveau utilisateurs
- Les Nouveautés (fraîches !)  
Pour les utilisateurs fréquents
- Aide à la navigation
- Moteur de recherche



## **La page d'accueil définit le “look and feel” du site**

Mais: Logo plus gros, éléments de navigation plus importants

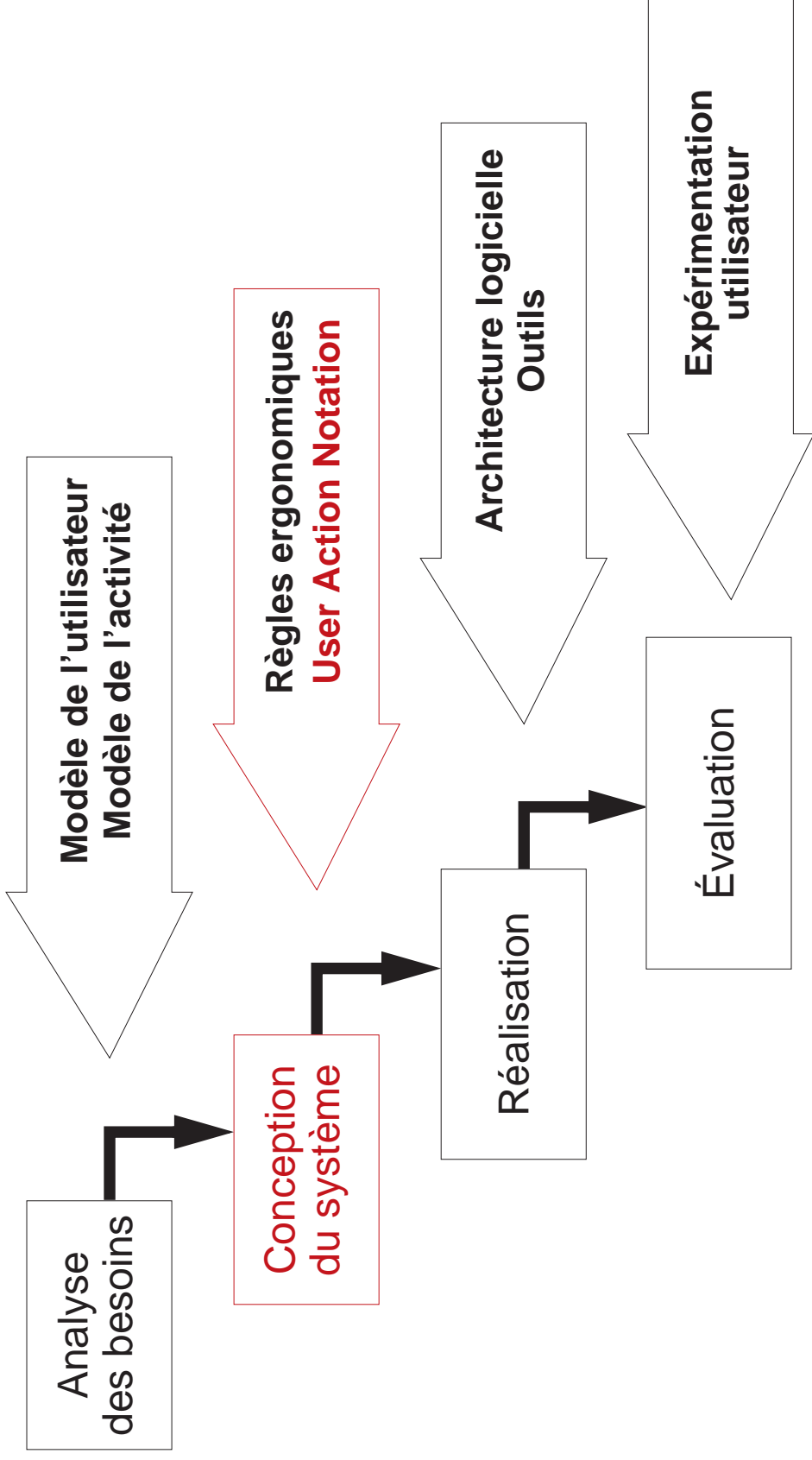
### **À éviter**

Un bouton pour le retour à la page d'accueil (entraîne la confusion).





# Plan du cours





## USER ACTION NOTATION

*Deborah Hix, H. Rex Hartson “Developing User Interfaces” Wiley 1993*

- Une description **comportementale** du système  
(opposé à une représentation **constructionnelle**)
- Outil de communication :
  - ◇ entre **concepteurs** de l’interface homme-machine
  - ◇ entre concepteurs et **développeurs**.
- Outil de conception :  
nous force à nous poser les bonnes questions

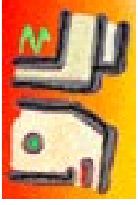


## Exemple : sélection d'un fichier à la souris

*“Déplacer le curseur sur l'icône du fichier  
Appuyer sur le bouton de la souris et relâcher  
immédiatement”*

En notation UAN :

~[icône du fichier] Mv M^



## Retour d'information de l'interface :

Tâche : Sélection d'un fichier à la souris	
Actions de l'utilisateur (User Actions)	Retour d'information (Interface feed-back)
~[icône du fichier] Mv	icône du fichier' !
M^	



## Pré-condition, changement d'état de l'interface :

Tâche : Sélection d'un fichier à la souris		
Actions de l'utilisateur (User Actions)	Retour d'information (Interface feed-back)	Changement d'état (Interface state)
icône du fichier -! : ~[icône du fichier] Mv	icône du fichier' ! ∀ icône de fichier ≠ icône du fichier' : icône de fichier -!	sélectionnés = fichier'
M^		

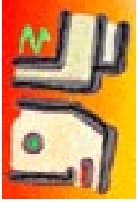


## Abstraction de Tâches

Tâche : Sélection (objet')	Actions de l'utilisateur (User Actions)	Retour d'information (Interface feed-back)	Changement d'état (Interface state)
	objet' -! : ~[objet'] Mv	objet' ! $\forall$ objet $\neq$ objet' : objet -!	sélectionnés = objet'
	M^		



Tâche : Afficher les messages reçus		
Actions de l'utilisateur (User Actions)	Retour d'information (Interface feed-back)	Changement d'état (Interface state)
-(vue = messages reçus): Sélection (bouton message reçus)	afficher (fenêtre des messages reçus)	vue = messages reçus



## Relation temporelles

- Séquence : A B
- Itération :  $A^+ A^n A^*$
- Option : { A }
- Choix : A | B
- Ordre indéterminé : A & B  
 $\langle \text{Ctrl} \rangle v Av (\langle \text{Ctrl} \rangle^{\wedge} \& A^{\wedge})$
- Interruption :  $A \Rightarrow B$
- Entrelacement :  $A \Leftrightarrow B$
- Parallélisme :  $A \parallel B$





## Exemple: gérer le courrier

Tâche: gérer le courrier	
Actions de l'utilisateur (User Actions)	Retour d'information (Interface feed-back)
(Traiter le courrier arrivé <=>	
Écrire des lettres <=>	
Gérer les archives)*	

Tâche: traiter le courrier arrivé
Actions de l'utilisateur
(Trier les enveloppes)*
(Lire le courrier)+

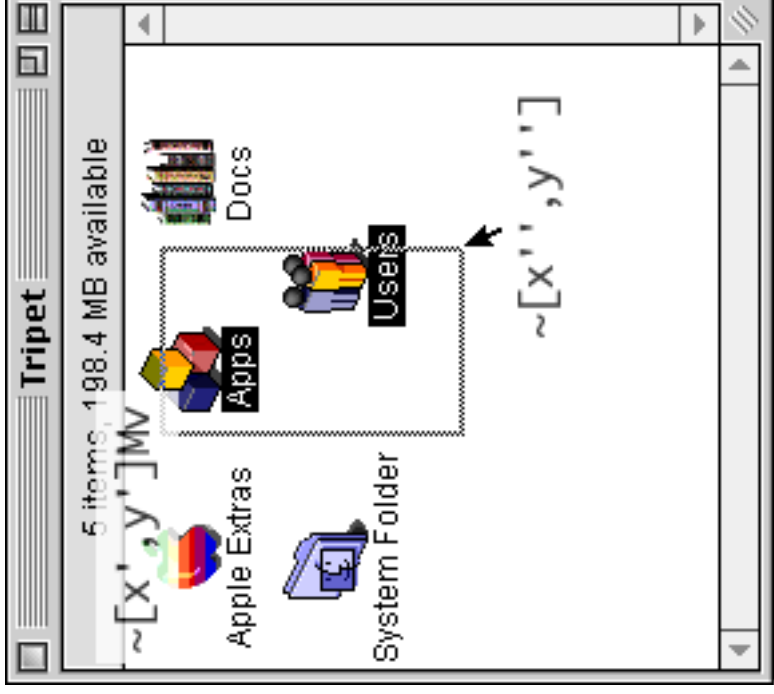


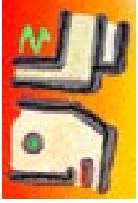
## Extensibilité de la notation

- Définition de nouveaux symboles, fonctions :  
contour (icône du fichier’)
- Commentaires hors du tableau :  
“sélectionné” représente l’ensemble des icônes mis en surbrillance à un instant donné.
- Connexion avec le noyau fonctionnel  
Exemples :  
consulter les derniers messages reçus  
détruire les fichiers sélectionnés

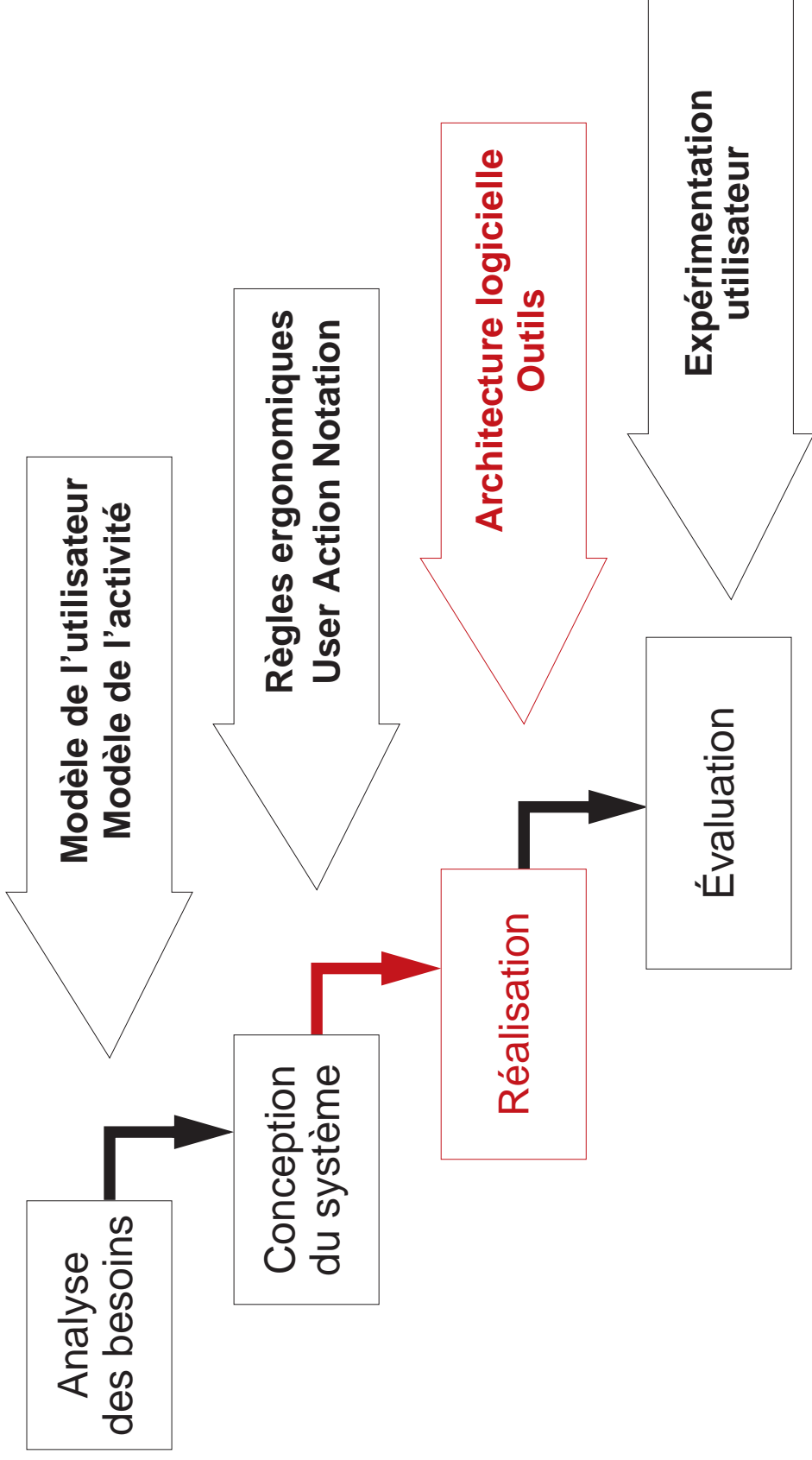


## Schémas, copie d'écran :





# Plan du cours





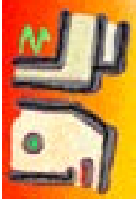
# Réalisation (codage)

En principe, dans la phase de réalisation il n'y a plus de problèmes centrés sur l'utilisateur: tous les problèmes ont du être étudiés dans les phases précédentes (Analyse et Conception).

En pratique certaines contraintes peuvent apparaître (temps, faisabilité). Ces contraintes peuvent obliger à amender les spécifications externes (résultat de la phase de Conception).

Dans ce cas, effectuer des changements de contextes explicites entre les phases.

À partir de maintenant, nous nous intéressons aux problèmes de réalisation (codage).



# ARCHITECTURE

## Principes essentiels:

- **Séparation stricte entre le code du noyau fonctionnel et le code de l’I.H.M.**
- **Le code du noyau fonctionnel est indépendant du code de l’I.H.M. (il n’y fait jamais référence)**



## Qu'est-ce que le Noyau Fonctionnel (N.F.) ?

Il n'y a pas toujours facilement identifiable.

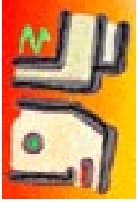
Exemple : Gestion du courrier électronique

Le NF est constitué des services réseau (émission / réception), des bases de données des messages et des adresses.

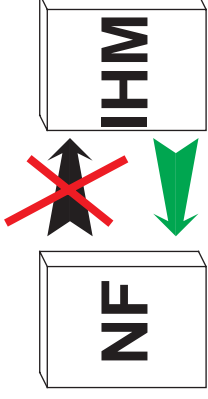
Définition :

“Les services offerts par le système, indépendamment de la façon dont l'utilisateur s'en sert”, ou bien:

“Tout ce qui n'a pas d'intérêt en I.H.M.”



## Indépendance du N.F. vis-à-vis de l'I.H.M.M.



### Pourquoi ?

- Séparer les problèmes : mise au point du N.F. sans se soucier de l'I.H.M.M.
- Les évolutions fréquentes de l'I.H.M.M. (itération des cycles de développement) n'ont pas de conséquences sur le N.F.





## Comment ?

Instrumentation du code

- état du système modifiable
- état du système accessible



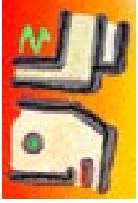
## État du système modifiable

	Code du N.F.	Code de l'IHM
<b>X</b>	<pre>set NF_nom [.entry get]</pre>	
<b>!</b>	<pre>proc NF_SetNom { val } {   global NF_nom   set NF_nom \$val }</pre>	NF_SetNom [.entry get]





## État du système observable

	Code du N.F.	Code de l'IHM
X	<pre>.labelNbMessage configure \ -text \$NF_nbMessage</pre>	
!	<pre>proc NF_nbMessage { } { global NF_nbMessage return \$NF_nbMessage }</pre>	<pre>.labelNbMessage configure \ -text [NF_nbMessage]</pre>



## Cas des événements

	Code du N.F.	Code de l'IHM
	<pre>.boutNouvMsg configure \ -background red</pre>	
	<pre>proc NF_EnregTraitNouvMsg { t } { global NF_traitantsNouvMsg  lappend NF_traitantsNouvMsg \$t }  # Un message vient d'arriver foreach traitant \     \$NF_traitantsNouvMsg {     \$traitant }</pre>	<pre>NF_EnregTraitNouvMsg \     IHM_TraitantNouvMsg  proc IHM_TraitantNouvMsg {} {     .boutNouvMsg configure \         -background red }</pre>



## Cas des événements (en C)

	Code du N.F.	Code de l'IHM
<b>X</b>	<pre>IHM_GereNouveauMsg ( );</pre>	
<b>!</b>	<pre>typedef void (*Traitant) ( ); Traitant gNF_TraitantsNouvMsg[10];  void NF_NouveauMsg ( ) { int i;      for (i = 0; i &lt; nbTraitant; i++)         (*(NF_TraitantsNouvMsg[i]))             (); }</pre>	<pre>NF_EnregistreTraitantNouvMsg (&amp;IHM_GereNouveauMsg);  void IHM_GereNouveauMsg ( ) { ... }</pre>



# HTML

“HyperText Markup Language”.  
Langage de description d’information structurée portable

Référence: WWW consortium.  
<http://www.w3c.org/>

HTML 3.2: <http://www.w3.org/TR/REC-html32>

HTML 4.01: <http://www.w3.org/TR/html4/>

Utiliser votre moteur de recherche préféré (<http://www.google.com/>) et chercher “introduction à HTML”.



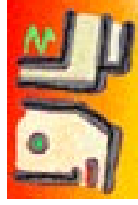
HTML à pour rôle la description de la **structure** du contenu, pas sa présentation.

(Malheureusement ?) Rien n’a été prévu pour la **présentation**.

▷ Netscape, Microsoft ont introduit des modifications “sauvages” à HTML qui ont été ensuite incorporée au standard.

Problème de la présentation: va à l’encore du “multi plate-forme”.

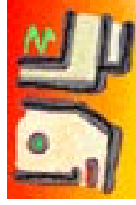
▷ Solution: “cascading style sheets” (CSS)



## La base

- La description d'une page en HTML est contenue dans un fichier.
- Par convention, on utilise le suffixe.html (“hello.html”).
- Les navigateurs peuvent charger un page HTML à partir du réseau mais aussi directement depuis un fichier (option <Fichier> <Ouvrir>).
- On peut toujours voir le code HTML d'une page (<Vue> <Source>).
- Si on modifie un fichier dont la page est affiché à ce moment là, on peut voir les nouvelles modifications avec le bouton “rafraîchir” .





## Les balises

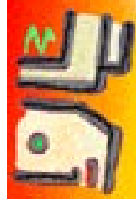
- L'information est structurée à l'aide de balises (“tags”) compris entre les caractères “inférieur” < et “supérieur” >.  
exemple: <html >
- Pas de différences entre majuscules et minuscules.  
<html > <HTML > <hTmL >
- Fonctionnement par deux: balise ouvrante et balise fermante (préfixé par le signe division /).  
<html > . . . </html >
- Les balises ont souvent des options  
<table width= " 60% " >



## Document HTML minimal

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN" >
<html>
<head>
  <title>
    Document HTML minimal
  </title>
</head>
<body>
  Du texte.<p>
</body>
</html>
http://ihtm.imag.fr/fberard/ens/webdev/ex\_html/minimal.html
```

- Le code HTML est compris entre les balises `<html>` et `</html>`.
- Il contient obligatoirement 2 sous-sections: “head” et “body”.
- La section “head” doit contenir un “title”.



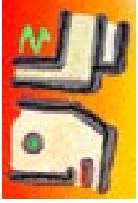
*Que se passe-t-il si on ne respecte pas les règles ?*

## **Que se passe-t-il si on ne respecte pas les règles ?**

Ça peut bien se passer... ou mal.

Si ça se passe bien sur un navigateur, ce n'est sans doute pas le cas sur les autres.

Si ça se passe bien maintenant, ce ne sera peut être pas le cas plus tard (nouvelles versions).



## Balise <body>

**bgcolor = “black | green | gray | white | yellow | ... | #RRGGBB”**

#RRGGBB: valeur des composantes rouge (Red), vert (Green) et bleu (Blue) de la couleur, exprimée en hexadécimal. Exemple #FF8800.

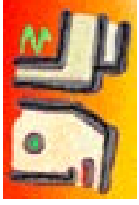
**background = “image.gif”**

Utilise une image comme fond. Attention! Souvent illisible.  
Atténuer le contraste de l’image, augmenter la luminosité, pas de texte.

**text, link, vlink, alink**

Modifie la couleur par défaut du texte, des liens. En règle générale un mauvaise idée.

```
<body bgcolor="white" text="#000044" > ... </body>
```

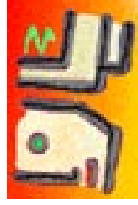


## Division

```
<div> ... </div>
```

Option: align="left | center | right | justify"

Utiliser `<div>` plutôt que `<center>`.



## Retour à la ligne

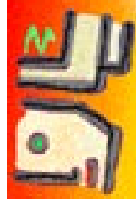
- `<P>` termine un paragraphe.
- `<BR>` provoque un retour chariot.

Noter que les retour-chariot dans le code HTML ne sont pas utilisés par les navigateurs.

`<p>`

Pour provoquer un retour chariot : `<br>`  
utiliser les balises paragraphe ou break.

`<p>`



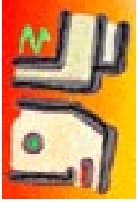
## Caractères spéciaux

é	&eacute;
è	&egrave;
ê	&ecirc;
ë	&euml;
ç	&ccedil;
œ	&oelig;

>	&gt;
<	&lt;
&	&amp;
"	&quot;

Référence:

<http://www.w3.org/MarkUp/html-spec/html-spec.toc.html#SEC9.7>



## Éléments de texte

- Insister: `<em> .. </em>` (emphasis)
- Insister fortement: `<strong> ... </strong>`
- Citation: `<cite> ... </cite>`

[http://iim.imag.fr/fberard/ens/webdev/ex\\_html/texte.html](http://iim.imag.fr/fberard/ens/webdev/ex_html/texte.html)





## Sous-titres

Balises `<h1>` sous-titre de niveau 1 `</h1>` `<h2>`... `</h2>`...

```
<h1>Grand sous-titre</h1>
```

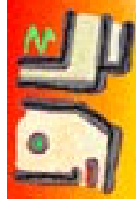
```
<h2>Petit sous-titre (1)</h2>
```

```
Un paragraphe<P>
```

```
<h2>Petit sous-titre (1)</h2>
```

```
Un autre paragraphe<P>
```

[http://iim.imag.fr/fberard/ens/webdev/ex\\_html/headings.html](http://iim.imag.fr/fberard/ens/webdev/ex_html/headings.html)



## Listes

### Non ordonnées

```
<ul>  
  <li> premier élément </li>  
  <li> deuxième élément </li>...  
</ul>
```

### Ordonnées

```
<ol> .. </ol>
```

### De définitions

```
<dl>  
  <dt> liste ordonnée </dt>  
  <dd> une liste à numéros </dd>  
  <dt> liste non ordonnée </dt>  
  <dd> liste sans numéros </dd>  
</dl>
```



## Autres balises

### Texte préformaté

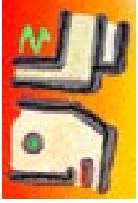
```
<pre>
for (i = 0; i < 10; i++) {
    a += i;
}
</pre>
```

### Trait horizontal

```
<hr>
```

### Citation

```
<blockquote> ... </blockquote>
```



## Tables

Souvent utilisées pour organiser l'information dans l'espace.

```
<table>  
<tr>  
  <td>case 0,0</td>  
  <td>case 1,0</td>  
</tr>  
<tr>  
  <td>case 0,1</td>  
  <td>case 1,1</td>  
</tr>  
</table>
```

case 0,0	case 1,0
case 0,1	case 1,1

[http://ihtm.imag.fr/fberard/ens/webdev/ex\\_html/tables.html](http://ihtm.imag.fr/fberard/ens/webdev/ex_html/tables.html)  
[http://ihtm.imag.fr/fberard/ens/webdev/ex\\_html/tableComplexe.html](http://ihtm.imag.fr/fberard/ens/webdev/ex_html/tableComplexe.html)



## Balise `<table>`

- Largeur de la table par rapport à la page: `width="80%"`
- Largeur des traits de la table: `border="1"` (en pixels)
- Écart entre les cellules: `cellspacing="5"` (en pixels)
- Écart entre le texte et sa cellule: `cellpadding="10"` (en pixels)

## Balise dans une table

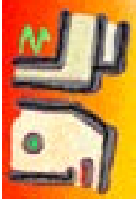
- Titre: `<caption> ... </caption>`
- Ligne: `<tr> ... </tr>`
- Cellule: `<td> ... </td>`

Couleur: `bgcolor="red"`

Cellule sur plusieurs lignes: `rowspan="x"`

Cellule sur plusieurs colonnes: `colspan="x"`

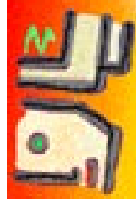
Alignement horizontal et vertical du texte dans les cellules:



## RÉALISATION (CODAGE)

## HTML *Tables*

```
align="left | center | right, valign = top | middle | bottom"
```

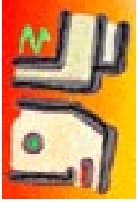


## Liens

Utilisent la syntaxe des URL (Uniform Resource Locator).

`<protocole><ressource>`

Web	<code>http://iihm.imag.fr/index.html</code>
ftp anonyme	<code>ftp://iihm.imag.fr/</code>
ftp connecté	<code>ftp://fberard@iihm.imag.fr/tmp/</code>
Fichier	<code>file:///Escampet/Installation</code>
Web + ancre	<code>http://iihm.imag.fr/fberard/#web</code>



## Ancres

### **Ancre avec lien absolu**

La page d'accueil de

l' `<a href="http://iihm.imag.fr/">équipe IAHM</a>`.

La page d'accueil de l'[équipe IAHM](#).

### **Ancre avec lien relatif**

Pour plus d'info, voyez la

`<a href="infos/coordonnees.html">page des coordonnées</a>`.

### **Définition de la base des liens relatifs**

Dans la partie `<head> ... </head>`:

`<base href="http://iihm.imag.fr/">`





## Ancre de destination dans une page

```
<a name="contacts" >
```

Liens vers cette ancre:

- de la même page: `<a href="#contact" > ... </a>`
- d'une autre page: `<a href="repertoire/la_page.html#contact" > ... </a>`



## Images

```

```

Alignement par rapport à la ligne de texte:

```
align="top | middle | bottom | left | right"
```

Largeur du bord pour des images utilisées comme ancre de départ:  
`border="0"`

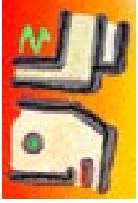


## Image cliquable

```

<map name="map1" >
  <area href="guide.html" alt="Access Guide"
    shape="rect" coords="0,0,118,28" >
  <area href="search.html" alt="Search"
    shape="circle" coords="184,276,28" >
</map>
shape="rect" coords="gauche, haut, droite, bas"
shape="circle" coords="centre-x, centre-y, rayon"
shape="poly" coords="x1,y1, x2,y2, x3,y3, ..."
```

<http://www-clips.imag.fr/>



# CSS

“Cascading Style Sheet” “feuilles de styles en cascade”

Référence: <http://www.w3c.org/Style/CSS/>

Document(s) décrivant **la présentation** de l’information (“feuilles de styles”).

Implémentation d’héritage de propriétés (aspect “cascade”).

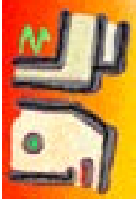


## Exemple

Tiré de <http://www.w3c.org/Style/LieBos2e/enter/>

- Description de la présentation des balises

```
H1 {  
    color: green;  
    text-align: center;  
}  
H1, H2, H3 { font-style: bold }
```



- Association des styles au document

## ◇ Dans le document HTML

```
<HTML>
<HEAD>
  <TITLE>Bach's home page</TITLE>
  <STYLE>
    H1, H2 { color: green }
  </STYLE>
  ...
<BODY>
  <P STYLE="color: green">affiche en vert</P>
```

## ◇ Par référence à une feuille de style externe (dans la section “HEAD”)

```
<LINK REL=STYLESHEET TYPE="text/css"
      HREF="http://style.com/cool" TITLE="Cool" >
```



# HTTP

**Hyper Text Transfer Protocol.**

Objectif: permettre à de nombreuses plates-formes hétéroclites de communiquer entre elles.

3 clefs du succès

- TCP/IP,
- Messages sous formes de chaînes de caractères (lisibles par l'homme),
- Protocole de très simple.

HTTP est orienté vers la portabilité, non pas vers la performance.



Protocole client / serveur, point à point.

Client



Requête



Réponse

Serveur



Clients et serveurs communiquent par **transaction** (séquence requête, réponse).

Les requêtes et réponses ont un format particulier: celui des **messages HTTP**





## Messages

Format:

1. Première ligne
2. Lignes d'en-tête (optionnel)
3. Ligne blanche
4. Corps du message



## Message de requête

### 1. Première ligne

```
GET /chemin/vers/fichier.html HTTP/1.0
```

“GET” est une des **méthodes** d’une message de requête.

Demande au serveur de transmettre le contenu d’une **ressource**.

“POST” et “HEAD” sont deux autres méthodes possibles.



## 2. Lignes d'en-tête

### Sous la forme:

Nom-en-tete: valeur

ou

Nom-en-tete: tres-longue-valeur,  
suite-de-la-tres-longue-valeur

### En-têtes requis (HTTP/1.1)

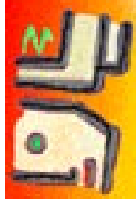
Host: www-ufrima.imag.fr:80

Nécessaire pour gérer les sites “multi-accueil” (“multi-homing”)

### En-têtes recommandés

User-Agent: Mozilla/3.0Gold

If-Modified-Since: Fri, 31 Dec 1999 23:59:59 GMT



## 1. Ligne blanche

Les caractères “Carriage Return” ET “New Line” (standard PC).

En C: `printf ( "\r\n" );`

## 2. Corps du message

Requête: le corps est vide (exception: méthode POST)

Réponse: le contenu de la ressource demandée (exception: réponse à une méthode HEAD).



## Message de réponse

### 1. Première ligne

HTTP/1.0 200 OK

HTTP/1.0 404 Not Found

### Classes de codes de retour:

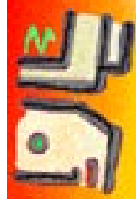
1xx Informations

2xx Succès

3xx Re-direction

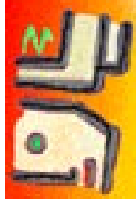
4xx Erreur client

5xx Erreur serveur



Codes de retour les plus fréquents

- 200 “Ok” La requête a été traitée, la réponse est dans le corps du message.
- 301 “Moved Permanently” La ressource a été assignée à une nouvelle adresse. L'URL est donnée par l'en-tête “Location:” .
- 304 “Not Modified” La ressource n'a pas été modifiée depuis la date précisée par le champs “If-Modified-Since:” de la requête.
- 400 “Bad Request” Erreur de syntaxe.
- 401 “Unauthorized” Tentative d'accès à une ressource protégée. Nécessite une identification préalable.
- 404 “Not found” La ressource n'existe pas.
- 500 “Internal Server Error”
- 503 “Service Unavailable” Le serveur n'est pas en mesure de traiter la requête pour des raisons de surcharge ou de maintenance.

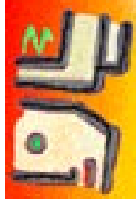


## 2. Lignes d'en-tête

```
Server: Apache/1.2b3-dev  
Last-modified: Fri, 31 Dec 1999 23:59:59 GMT  
Location: /new_site/page.html (cas d'un code retour 301)
```

### Si la réponse a un corps

```
Content-Type: text/html (type des données contenues dans la ressource)  
Content-Length: 345 (nombre d'octets de la ressource)
```



## 1. Ligne blanche

Les caractères “Carriage Return” ET “New line” (standard PC).

En C: `printf ( "\r\n" );`

## 2. Corps du message

Le contenu de la ressource demandée.

Exceptions:

réponse à une requête HEAD

réponse à une requête GET avec un en-tête “If-Modified-Since:”





## Exemple

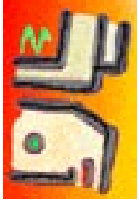
### Requête:

```
GET /fichier1.html HTTP/1.0  
Accept: text/html
```

### Réponse

```
HTTP/1.1 200 OK  
Date: Fri, 09 Jan 1998 09:49:11 GMT  
Server: Apache/1.3b2  
Last-Modified: Tue, 19 Aug 1997 11:57:17 GMT  
Content-Length: 118  
Content-Type: text/html
```

```
<HTML>  
<HEAD>  
(...)
```



## Un client HTTP minimal en Tcl

```
proc http_get { host rsrc { port 80 } } {  
    set sid [socket $host $port]  
  
    puts $sid "GET $rsrc HTTP/1.0"  
    puts $sid "Host: ${host}:${port}"  
    puts $sid ""  
    flush $sid  
  
    while { ![eof $sid] } {  
        gets $sid line  
        puts "rcv: $line"  
    }  
}
```

<http://ihtm.imag.fr/fberard/ens/webdev/http/client.tcl>



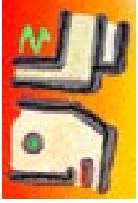
## Un serveur HTTP minimal en Tcl

```
set port8080
set serSock[socket -server Accept $port]

proc Accept { channel address port } {
    fconfigure $channel -translation { crlf crlf } -blocking 0

    puts $channel "HTTP/1.0 200 OK"
    puts $channel "Content-Type: text/html"
    puts $channel {}

    puts $channel \
"!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN\">"
    puts $channel
    puts $channel
    puts $channel <HEAD><TITLE>MyWebServer response</TITLE></HEAD>"
    puts $channel "<BODY>The request was<BR><PRE>"
```



## RÉALISATION (CODAGE)

## HTTP

*Un serveur HTTP minimal en Tcl*

```
while { [gets $channel line] != -1 } {  
    puts $channel $line  
}  
  
puts $channel "</PRE></BODY>"  
puts $channel "</HTML>"  
  
close $channel  
}
```

<http://iihm.imag.fr/fberard/ens/webdev/http/server.tcl>



## Limitation de HTTP

Les requêtes sont limitées à la demande de ressources “statiques” (fichiers).

Le client ne peut demander la “génération” d’une information qui lui est propre: les ressources existent sur le serveur indépendamment de tel ou tel client.

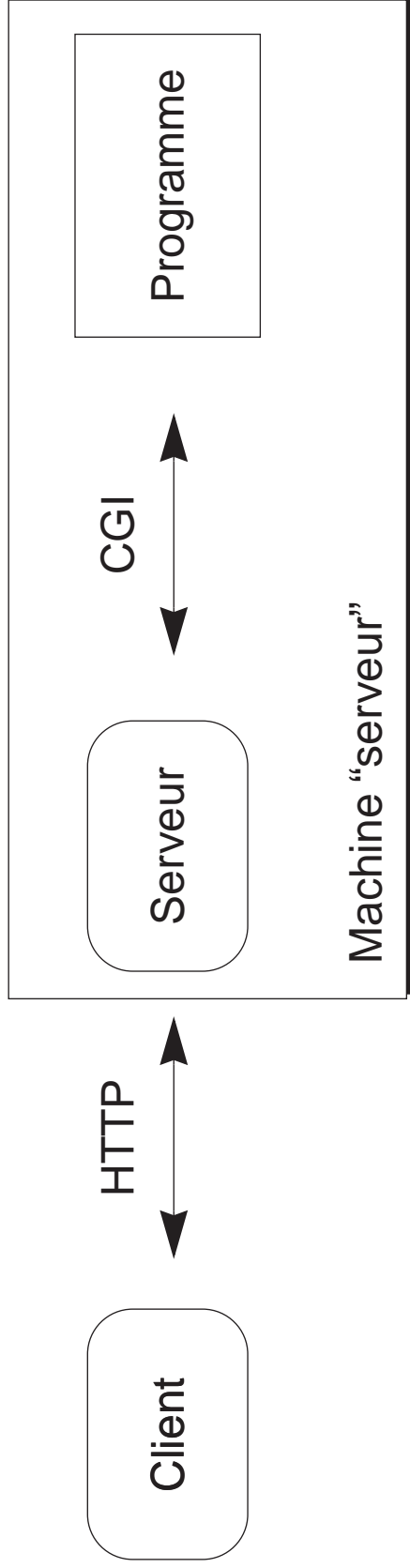
Nécessité de pouvoir demander l’exécution d’un programme sur le serveur.

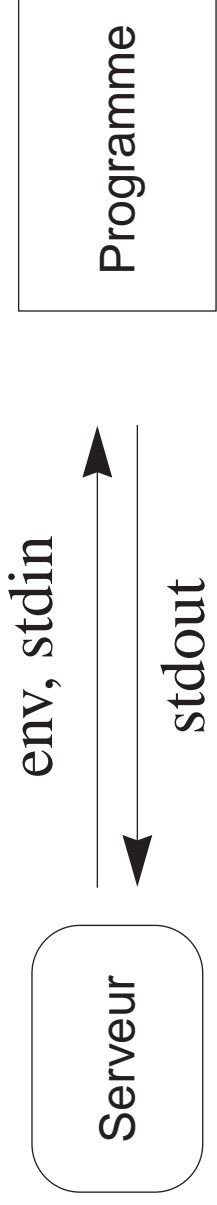


# CGI

“Common Gateway Interface”

Protocole de communication entre un serveur HTTP et les programmes destinés à générer des ressources dynamiques.

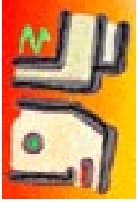




Le serveur effectue une “requête” auprès du programme par l’intermédiaire des variables d’environnement et du flux “standard input” .

Le programme donne sa “réponse” par l’intermédiaire du flux “standard output” .

Les cgi sont placés dans un répertoire particulier afin de les séparer les ressources non exécutable.



## Exemple minimal

```
#include <stdio.h>

int main (int argc, char* argv[])
{
    printf ( "Content-type: text/html\r\n\r\n" );

    printf ( "<!DOCTYPE HTML PUBLIC " );
    printf ( "\" - //W3C//DTD HTML 3.2 Final//EN\">\r\n" );

    printf ( "<html>\r\n<head>\r\n" );
    printf ( "<title>Hello, CGI !</title>\r\n" );
    printf ( "</head><body>A very small CGI" );
    printf ( "</body></html>\r\n" );

    return 0;
}
```

<http://iihm.imag.fr/fberard/ens/webdev/cgi/basic.c>





## Passage de paramètres au CGI

Balise `<isindex>`

Nécessite la définition d'une adresse de base dans la partie `<head>`

```
<HEAD>  
<TITLE>Exemple IsIndex</TITLE>  
<BASE href="http://eprouvet.imag.fr/cgi-bin/isindex.cgi">  
</HEAD>
```



## RÉALISATION (CODAGE)

## CGI

### Passage de paramètres au CGI

```
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char* argv[])
{
    printf ( "Content-type: text/html\r\n\r\n" );

    (...)

    printf ( "</head><body>\r\n" );
    printf ( "env(QUERY_STRING)=%s<BR>\r\n" ,
            getenv ( "QUERY_STRING" ) );
    printf ( "</body></html>\r\n" );

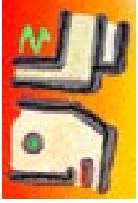
    return 0;
}
```

<http://iitm.imag.fr/fberard/ens/webdev/cgi/isindex.html>



## Formulaires

```
<form method="GET" action="/cgi-bin/traitform.tcl">
Nom:
<input type="text" name="nom" size="8" maxlength="20"><br>
Status <select name="status" >
    <option selected>Etudiant</option>
    <option>Militaire</option>
    <option>Sans emploi</option>
    <option>Retraite</option>
</select>
<P>
<INPUT TYPE="SUBMIT" NAME="Valide" VALUE="valide">
<INPUT TYPE="RESET" VALUE="Remise &agrave; z&eacute;ro">
</form>
```



```
#!/usr/bin/tclsh
puts "Content-type: text/html"
puts ""
puts -nonewline "<!DOCTYPE HTML PUBLIC "
puts "\"-//W3C//DTD HTML 3.2 Final//EN\">"
(...)
set query $env(QUERY_STRING)
while { [regexp {(^[=]*)(&|)(.*)} \
        $query dum opt val dum2 query] } {
    set res($opt) $val
}
foreach opt [array names res] {
    puts "$opt = $res($opt)<br>"
}
close stdout
exit 0
(...)
```

**penser au “chmod a+x”**

<http://iuhm.imag.fr/fberard/ens/webdev/cgi/traitform.tcl>



## 2 méthodes de passage de paramètres

- GET

Les paramètres sont fournis dans l'URL de la requête

```
env(HTTP_QUERY) =  
http://ihtm.imag.fr/cgi-bin/traitform.cgi?
```

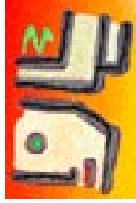
- POST

Les paramètres sont fournis au travers du canal “standard input”

```
int c  
while ((c = getchar()) != EOF)  
    printf ("%c", c);
```

**Une bibliothèque de services pour écrire des CGI en langage C**

<http://www.boutell.com/cgiic/>



## Autre limitation de HTTP

Requêtes “sans-état” : il est difficile de construire une interaction dont la durée de vie est supérieure à un cycle “requête - réponse” .

## COOKIES

Définis par Netscape.

[http://home.netscape.com/newsref/std/cookie\\_spec.html](http://home.netscape.com/newsref/std/cookie_spec.html)

## Principe

1. Le serveur demande au navigateur de sauvegarder un “cookie” sur le disque dur du client.
2. Lors d’une requête ultérieure à la même ressource, le client envoie le cookie dans les lignes d’en-tête de la requête.



## Exemple

### Requête:

```
GET /chemin/vers/fichier.html HTTP/1.0
```

### Réponse

```
HTTP/1.0 200 OK
```

```
Set-Cookie: id=0xb1234;
```

```
expires=Friday, 08-March-01 00:00:00 GMT
```

(...)

### Requête:

```
GET /chemin/vers/fichier.html HTTP/1.0
```

```
Cookie: id=0xb1234
```



## Attributs de “Set-Cookie”

- nom=valeur;

Minimum un, il peut y en avoir plusieurs. C’est l’information qui sera renvoyé par le client à la prochaine requête.

- expires=date;

Date d’expiration, heure GMT. Au delà de cette date le cookie est effacé sur la machine client. Par défaut, le cookie expire à la fin de la session client.

- path=/un/chemin

**Racine** des ressources pour lesquelles le cookie sera envoyé.

- domain=iilm.imag.fr

Domaine des ressources pour lesquelles le cookie sera envoyé. Le domaine doit avoir au moins 2 points (“.imag.fr” ok mais pas “.com”)





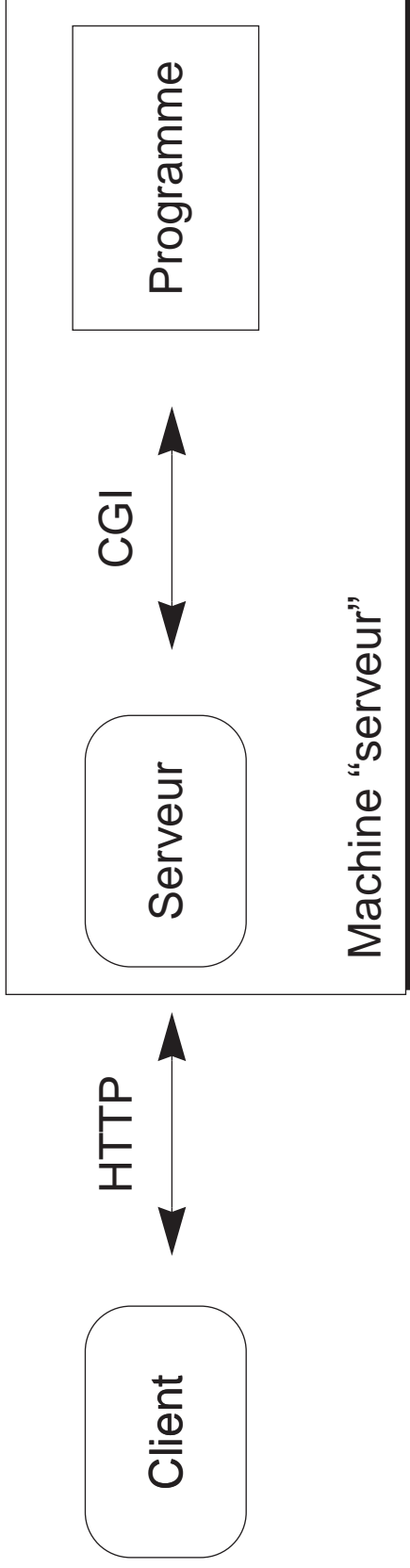
## Exemple de CGI

```
(...)  
  
printf ("Set-Cookie: TEST=HELLO_WORLD\r\n");  
printf ("Content-type: text/html\r\n\r\n");  
  
(...)  
  
if (getenv ("HTTP_COOKIE") != (char*)NULL) {  
    printf ("env(HTTP_COOKIE)=%s<BR>\r\n",  
           getenv ("HTTP_COOKIE"));  
} else {  
    printf ("pas de cookie<BR>\r\n", getenv ("HTTP_COOKIE"));  
}
```

<http://iitm.imag.fr/cgi-bin/fberard/webdev/cookies.cgi>



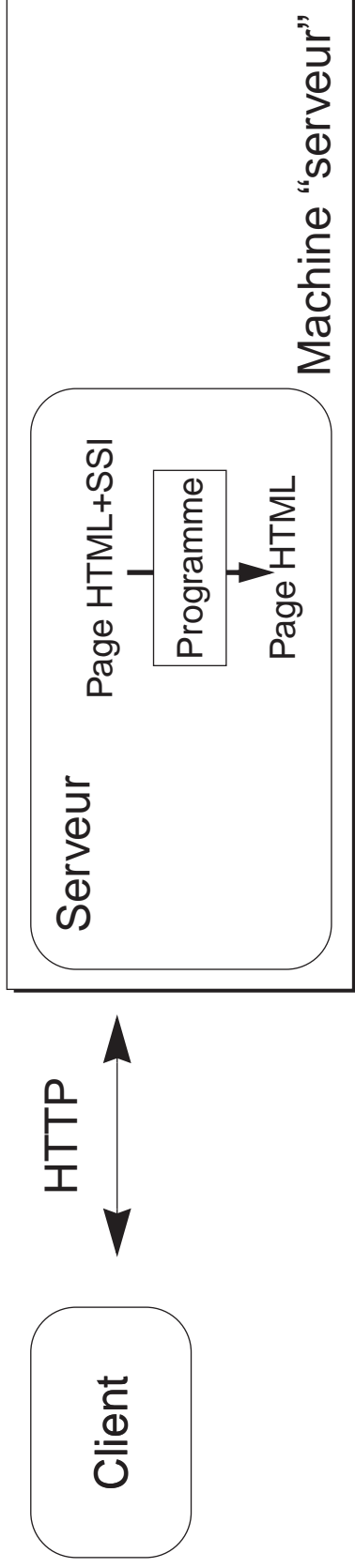
## PROBLÈMES LIÉS AUX CGI



1. À chaque requête, le serveur doit créer un nouveau processus pour exécuter le programme CGI.
2. Il est fastidieux de générer toute la page HTML dans le programme CGI alors que seulement une petite partie est dynamique.
3. L'aspect "dynamique" est limité par le temps de réponse: requête et réponse sur le réseau.

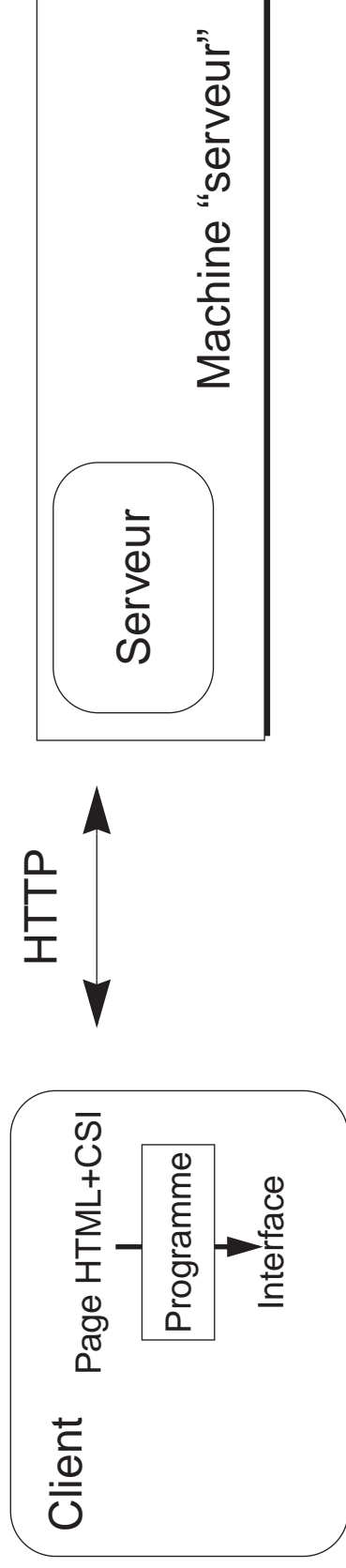


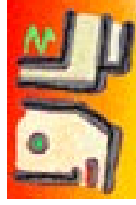
Problèmes 1 & 2 (création de processus, génération du code HTML):  
Server-Side Includes (SSI): PHP, tcl



Problèmes 3 (temps de réponse):

Client-Side Includes (CSI): JavaScript





## SERVER SIDE INCLUDE: PHP

“Php: Hypertext Preprocessor”

Langage interprété, créé pour faire des SSI.

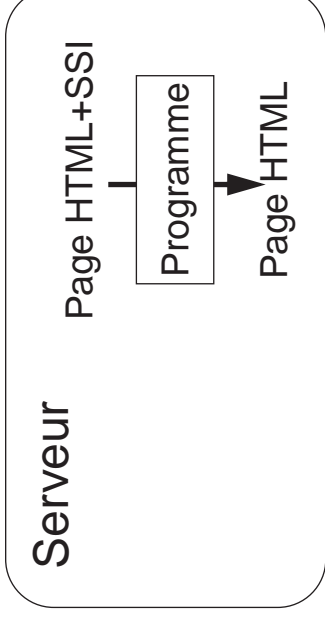
Maintenant, il est également utilisé pour faire des CGI, et même des applications autonomes.

La page officielle: <http://www.php.net/>

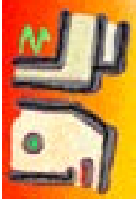
Un site “portail PHP”: <http://www.zend.com/>



## Mise en oeuvre



- Modification du serveur http par adjonction d'un interpréteur PHP
- Configuration du serveur pour qu'il traite certains fichiers.  
La convention la plus utilisée est d'utiliser un suffixe particulier (.php ou .php3) pour les fichiers qui doivent être pré-traités par PHP.
- Insertion de code PHP dans les pages HTML.



## Insertion dans HTML

Le code PHP est inclus entre les symboles: `<?php ... ?>`

### Exemple

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN" >
<html>
<head>
  <title>Exemple PHP</title>
</head>
<body>
  La suite de ce texte est en <?php
    print("PHP\n");
  ?>, n'est-ce pas ?
</body>
</html>
```

<http://iitm.imag.fr/fberard/ens/webdev/php/test.php3>



## Formulaires

Les valeurs des éléments d'interaction du formulaire (zone de saisie, case à cocher) sont disponibles:

- Dans les tableaux associatifs \$HTTP\_POST\_VARS et \$HTTP\_GET\_VARS.

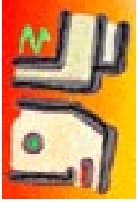
Le nom qui a été saisi est

```
<?php  
    print ( $HTTP_POST_VARS[ "nom" ] );  
?>
```

[http://iilm.imag.fr/fberard/ens/webdev/php/form\\_php.html](http://iilm.imag.fr/fberard/ens/webdev/php/form_php.html)

source du script qui traite le formulaire:

<http://iilm.imag.fr/fberard/ens/webdev/php/traitform.php3>



- Dans les variables qui portent le même nom que le nom donné aux éléments d'interaction.

Le nom qui a été saisi est

```
<?php  
    print ($nom);  
?>
```





## Cookies

Le serveur peut demander la sauvegarde d'un cookie en appelant la fonction "SetCookie".

L'appel doit être fait avant toute génération de code HTML de la part du script.

```
int setcookie (string name[, string value[, int expire[, string  
path[, string domain[, int secure]]]])  
  
setcookie ( "Option_Test" , "Valeur_Test" , time() + 3600 );
```

La lecture se fait par le tableau associatif `HTTP_COOKIE_VARS`:  
`$HTTP_COOKIE_VARS["Option_Test"]`



# Accès à un Système de Gestion de Bases de Données (SGBD)

Cas du SGBD MySQL.

## Connexion

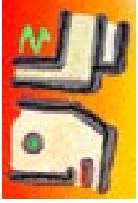
```
$link = mysql_connect("localhost", "user_ihm", "1totoche" )  
    or die ("Could not connect");  
print ("Connect&ecute;<BR>");  
mysql_select_db("cours_ihm", $link)  
    or die ("Could not select cours_ihm");  
print ("Table cours_ihm active<BR>");  
  
...  
mysql_close ($link);
```



## Ajout de données

```
mysql_query(  
    "INSERT INTO noms VALUES (\"$nom\", \"$prenom\");", $link)  
    or die ("Could not insert values in table");
```

[http://iitm.imag.fr/fberard/ens/webdev/mysql/test\\_mysql.html](http://iitm.imag.fr/fberard/ens/webdev/mysql/test_mysql.html)



## Affichage d'une table

```
$result = mysql_query( "SELECT * FROM noms; ", $link );
print ( "<TABLE width=\"100%\"><TR bgcolor=\"#BBBBBB\">" );
print ( " <TD># <TD><TD>Nom <TD><TD>Prenom <TD>" );
print ( " <TD>Sup ? </TD></TR>" );

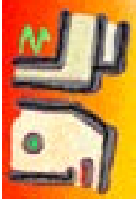
print ( " <FORM ACTION=\"delete_records.php3\" METHOD=\"POST\">" );
$count = 0;
while($row = mysql_fetch_array($result)) {
    print ( " <TR><TD>$count </TD><TD>$row[nom] </TD>" );
    print ( " <TD>$row[prenom] </TD><TD><INPUT TYPE=\"CHECKBOX\"" );
    print ( " NAME=\"delete[\" $row[nom]\"></TD></TR>" );
    $count = $count+1;
}
print ( " </TABLE><P>" );
print ( " <INPUT TYPE=\"SUBMIT\" VALUE=\"Delete records\"><BR>" );
print ( " </FORM>" );
```

[http://iilm.imag.fr/fberard/ens/webdev/mysql/show\\_table.php3](http://iilm.imag.fr/fberard/ens/webdev/mysql/show_table.php3)

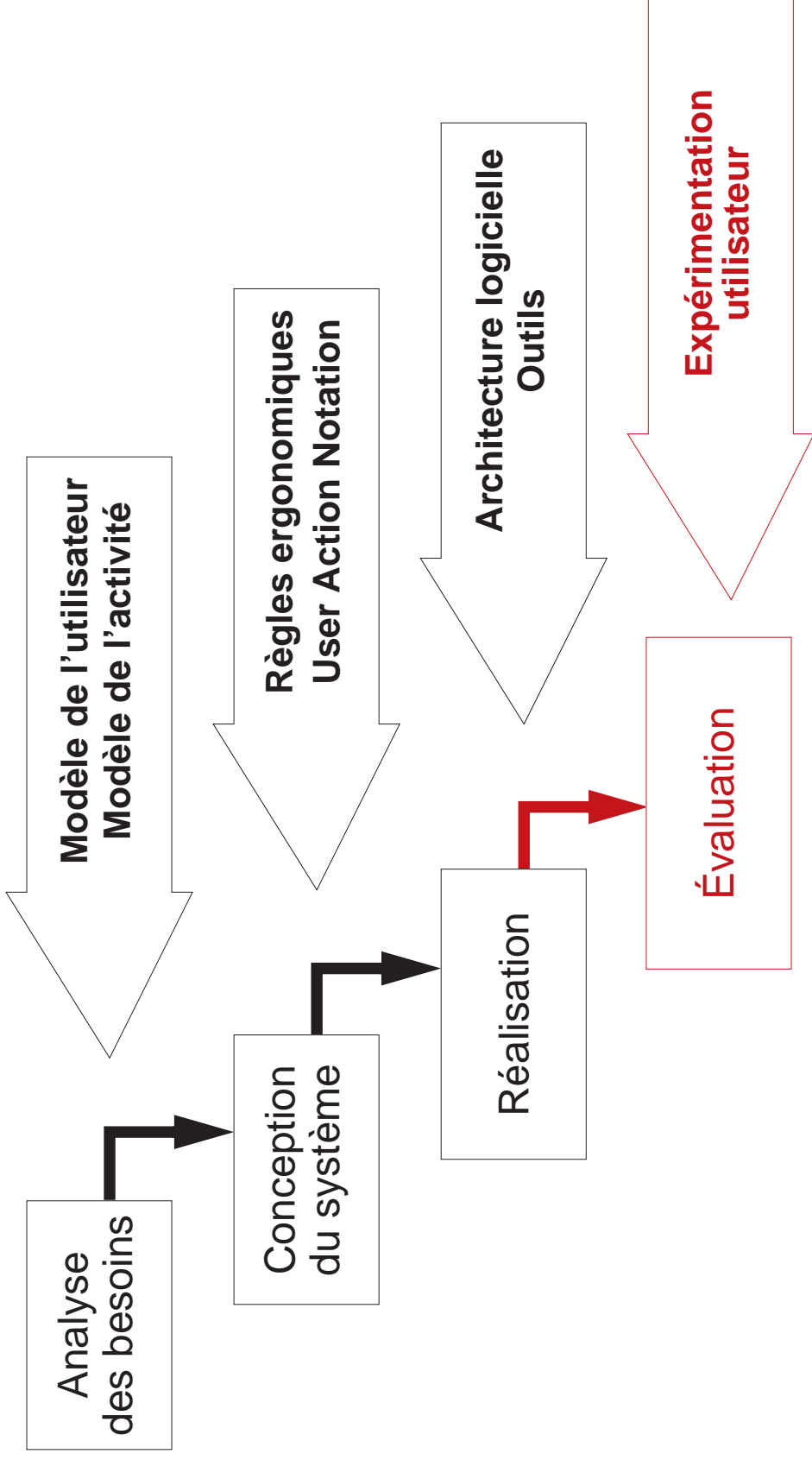


## Suppression de données

```
while (list($key,$val) = each($delete)) {  
    print ("nom=$val<BR>");  
    $result = mysql_query(  
        "DELETE FROM noms WHERE nom='$val';", $link)  
        or die ("Could not delete record $val");  
    }  
}
```



# Plan du cours





# Expérimentation utilisateur

Le conseil le plus important concernant l'expérimentation utilisateur est:  
“faites-la !”

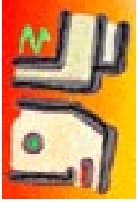
## DÉFINITION

Test du système fonctionnel.

Séances contrôlées d'utilisation du système par des “sujets”.

Les sujets sont externes au groupe de conception / développement.

Les séances sont enregistrées à des fins d'analyse.



## OBJECTIF

- Vérification de la couverture fonctionnelle du système
  - Est-ce que le système permet au sujet d'accomplir toutes les tâches qu'ils ont besoin de faire ?
  - Détection des erreurs de conception
  - Difficile de tout prévoir à priori.
- L'expérimentation met en évidence les parties de l'IHM qui posent problème aux sujets.
- Choix entre plusieurs solutions d'interface
  - Plusieurs solutions ont été retenues en phase de conception. Laquelle est la meilleure ?

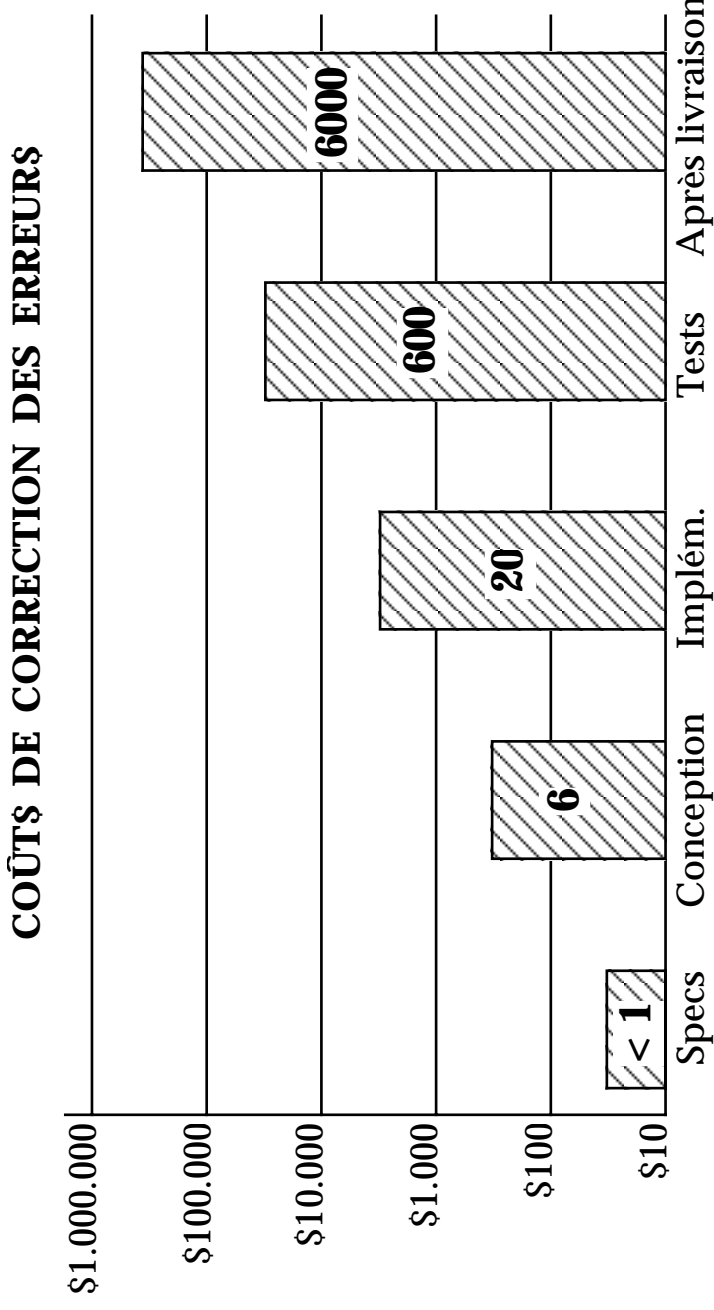




## À QUEL MOMENT ?

Dès que possible, souvent.

➤ sur maquette dans les phases initiales du développement



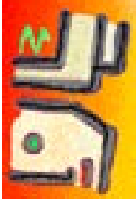


## EXEMPLE 1: ÉVALUATION D'UN SITE WEB

Rédaction d'un plan de test faisant intervenir les tâches principales (référence à l'arbre des tâches).

Choisir des tâches centrées utilisateur (ex: commander un article), non pas des tâches systèmes (ex: créer un login sur le serveur).

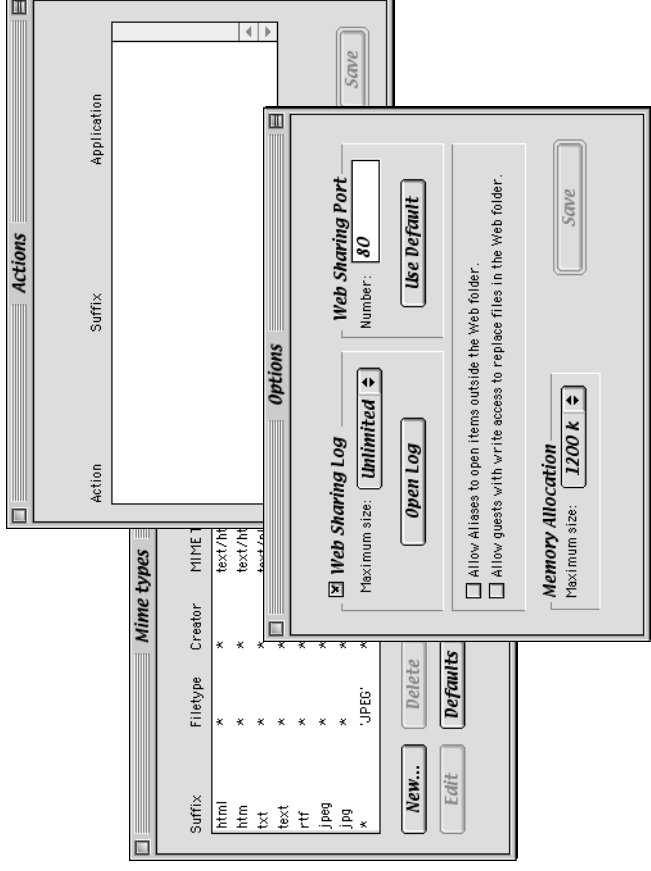
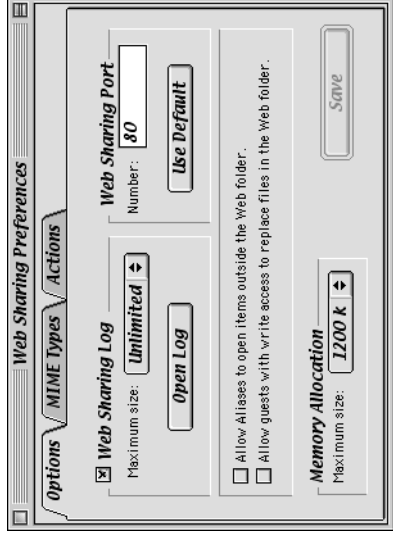
- Trouver le prix de l'article X.
- Quel est le meilleur rapport poids/prix pour un article de type Y
- Faites la commandes des article Z, W et K.
- Trouvez sur le site la réponse à la question Q

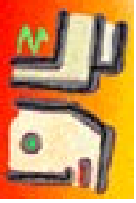


## EXEMPLE 2: CHOIX ENTRE DEUX SOLUTIONS D'INTERFACE

Évaluer statistiquement les effets de variations dans les conditions d'utilisation

Exemple: “*Une seule fenêtre à onglets est-elle préférable à plusieurs fenêtres ?*”





- Choix d'une tâche mettant en oeuvre l'interface à tester.  
Liste de réglages à effectuer
- Identification des variables indépendantes (facteurs)  
La présentation des fenêtres
- Identification des conditions  
Fenêtre à onglets (A) / plusieurs fenêtres (B)
- Choix d'une variable dépendante  
Le temps d'accomplissement de la tâche

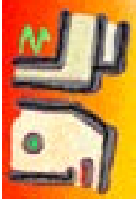


## ORGANISATION D'UN TEST UTILISATEUR

1. Se demander ce que l'on veut tester
2. Établir un plan de test (tâches à réaliser)
3. Faire passer le test
4. Analyser les résultats

### Deux critères importants

- Fiabilité du test  
Peux-t-on reproduire les résultats ?
- Validité du test  
As-t-on prévu les bonnes tâches à exécuter ?  
Est-ce que les utilisateurs sont représentatifs ?



## CHOIX DE LA POPULATION TEST

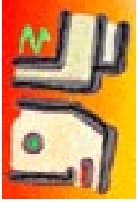
Idéalement: un sous-ensemble des utilisateurs finaux (ceux pour qui le système a été développé).

- Maximiser les variations entre utilisateurs (dans la catégorie cible)  
Test d'une population "représentative"
- Minimiser les variations entre les conditions  
Fiche de consigne



## DÉROULEMENT DU TEST (CHECKLIST)

1. Préparer le système pour le mettre dans l'état initial prévu avant l'arrivée de l'utilisateur
2. Souhaiter la bienvenue à l'utilisateur
3. Informer l'utilisateur de l'objet du test, et en particulier lui dire que **“c'est le système que l'on teste, non pas l'utilisateur”**.  
L'assurer de la confidentialité des résultats  
Ne pas lui informer que VOUS êtes concepteur du système, l'encourager à critiquer
4. Lui donner les connaissances minimales pour exécuter la tâche
5. Lui demander de “penser tout haut”
6. Durant le test, enregistrer le comportement, les paroles de l'utilisateur
7. Réaliser un entretien post-test pour recueillir les commentaires généraux de l'utilisateur et lui poser vos questions systématiques

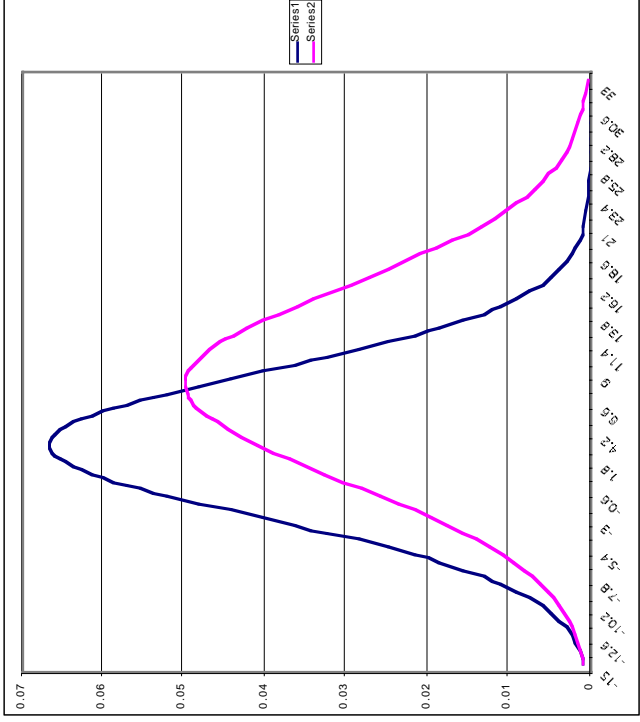


## ANALYSE DES DONNÉES

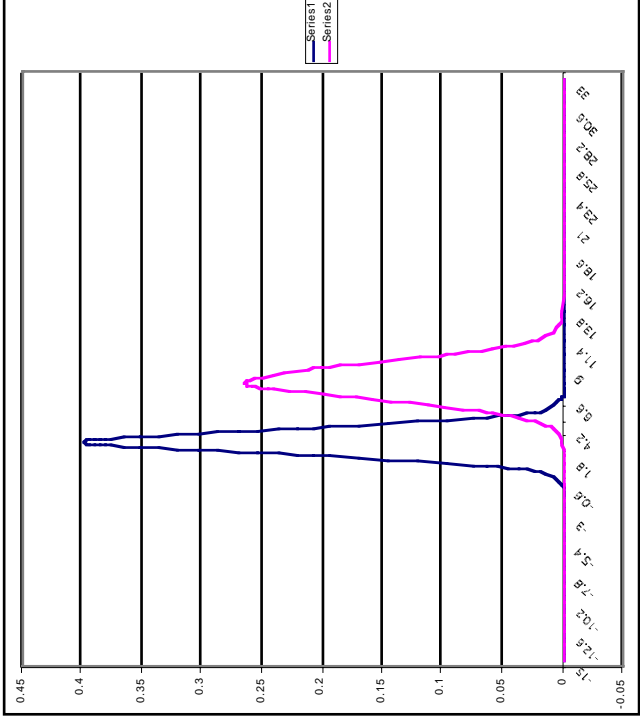
Condition	Moyenne des temps d'accomplissement de la tâche
(A) fenêtre à onglets	308 s.
(B) plusieurs fenêtres	402 s.

Condition (A) plus performante ?



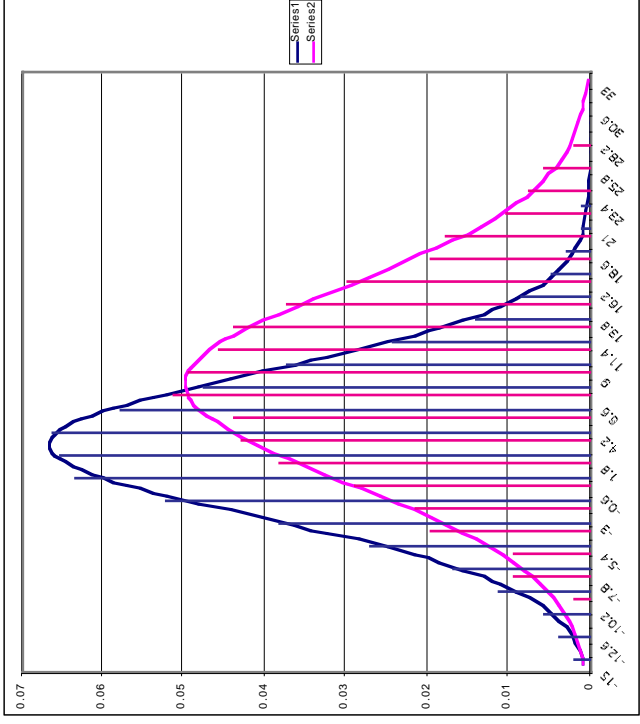


Cas 1 : forte variance

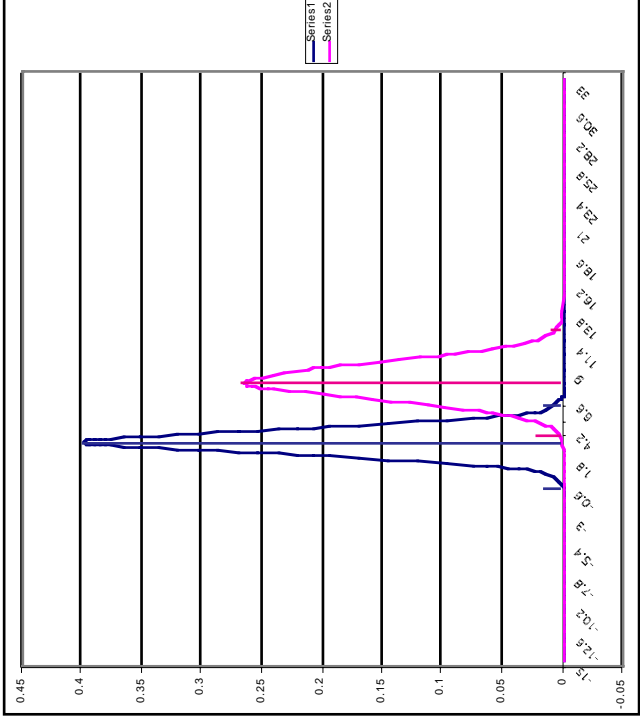


Cas 2: faible variance

Condition (A) plus performante?



Cas 1: grand degré de liberté



Cas 2: faible degré de liberté



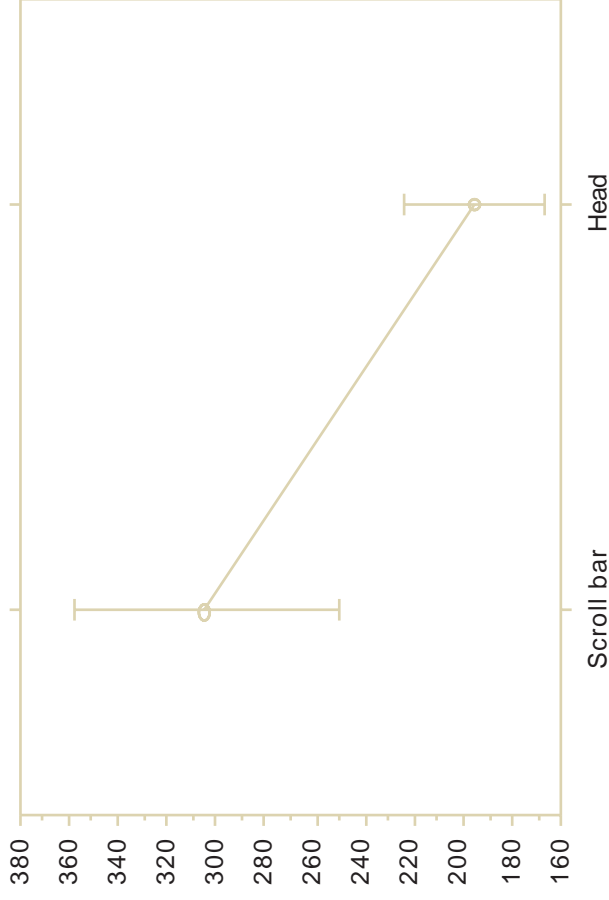
## Utilisation d'outils statistiques

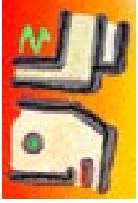
t-test, analyse de variance (excel, statview)

<http://davidmlane.com/hyperstat/>

Répondent à la question:

*Quelle est la probabilité que l'écart mesuré soit la cause du hasard ?*  
On valide l'expérience (“résultats significatifs”) si  $p < 0.05$  ou  $p < 0.01$





# Références en ligne

## Histoire de l'IHM

- Démonstration de Doug Engelbart (souris) en 1968  
<http://sloan.stanford.edu/mousesite/1968Demo.html>

## Règles ergonomiques

- “Human Interface Design” (Apple, dans “Introduction to the Aqua Human Interface Guidelines”)  
<http://developer.apple.com/techpubs/macosx/Essentials/AquaHIGuidelines/AHIGIGs/index.html>

## Conception de sites Web

- L’ergonome (Français) <http://www.lergonome.org/>
- UseIt (Jakob Nielsen) <http://www.useit.com/>
- Usable Web <http://www.usableweb.com/>

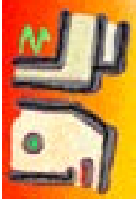


## HTML

- HTML 3.2 <http://www.w3.org/TR/REC-html32>
- HTML 4.0 <http://www.w3.org/TR/REC-html40/>
- WebTv <http://developer.webtv.net/authoring/>

## Feuilles de Style Surchargeables (Cascading Style Sheets, CSS)

<http://www.w3.org/TR/REC-CSS1>



## HTTP

- HTTP made really easy

<http://www.jmarshall.com/easy/http/>

- HTTP référence au W3C

<http://www.w3.org/Protocols/>

## CGI

- The Common Gateway Interface

<http://hoohoo.ncsa.uiuc.edu/cgi/>

- Tcl lib, contient les modules “ncgi”, “html”

<http://sourceforge.net/projects/tcllib/>

## Cookies

[http://home.netscape.com/newsref/std/cookie\\_spec.html](http://home.netscape.com/newsref/std/cookie_spec.html)



## RÉFÉRENCES EN LIGNE

### PHP

<http://www.php.net/>

<http://www.zend.com/>

### HyperStat Online (statistiques expérimentales)

<http://davidmlane.com/hyperstat/>