



# Introduction au domaine de l'Interaction Homme-Machine

David Navarre  
navarre@irit.fr - <http://ics.irit.fr/navarre>

Interactive Critical Systems  
IRIT – Université Toulouse I Capitole

# Objectifs du cours

- Introduction au domaine
- Introduction aux concepts
- Introduction aux techniques
- Introduction à la modélisation

# Contenu du cours

- Éléments de «User Centered Design»
- Éléments d 'ergonomie
- Éléments de psychologie cognitive
- Éléments de génie des systèmes interactifs
- Éléments de technologie des systèmes interactifs

# Organisation du cours

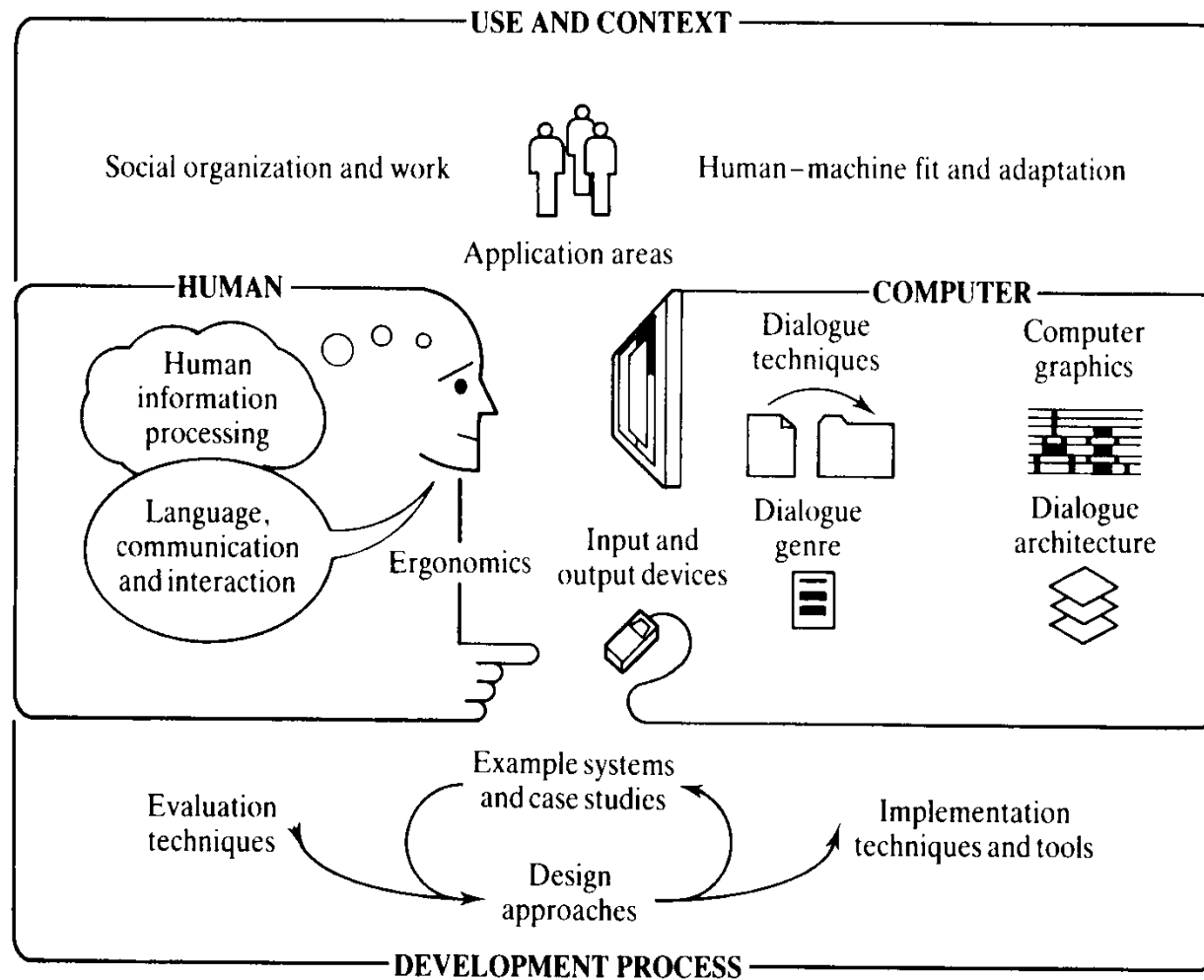
- Samedi 8h30 – 12h30
  - Introduction
  - Modèles utilisateurs
  - Introduction à la modélisation des tâches
- Samedi 14h30 – 18h30
  - Modélisation des tâches
  - Prototypage
- Dimanche 9h00 – 12h00
  - Prototypage (suite)
  - Introduction à l'ergonomie (évaluation)
  - Introduction à l'ergonomie (critères ergonomiques)
- Dimanche 14h00 – 17h00
  - Evaluation mise en œuvre
  - Processus de développement (IHM)
- Lundi 18h00 – 21h00
  - Fonctionnement par événement
  - Modélisation du dialogue
- Mardi 18h00 – 21h00
  - Modélisation du dialogue (suite)
  - Passage au code
  - Conclusion générale



# Les fondements

Par l'exemple

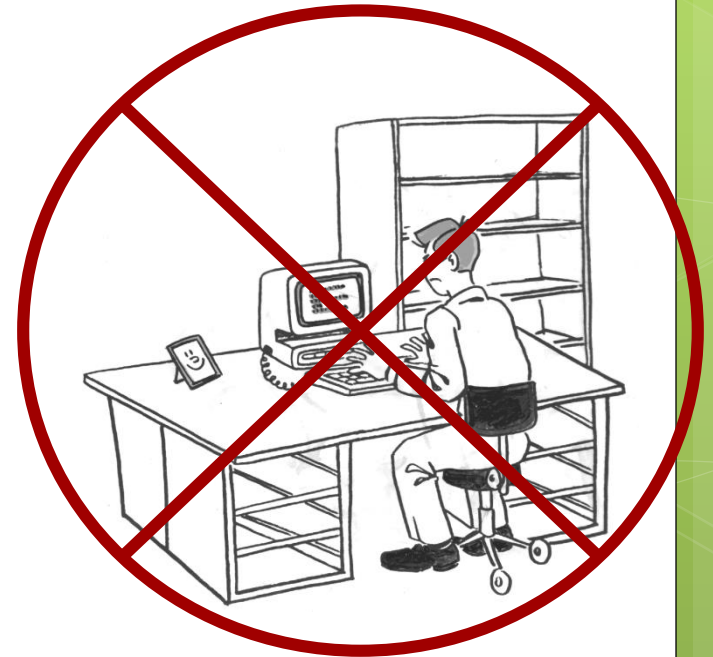
# La Carte de l'IHM



# Ça ne marche jamais !!!



# User-centered systems design in practice

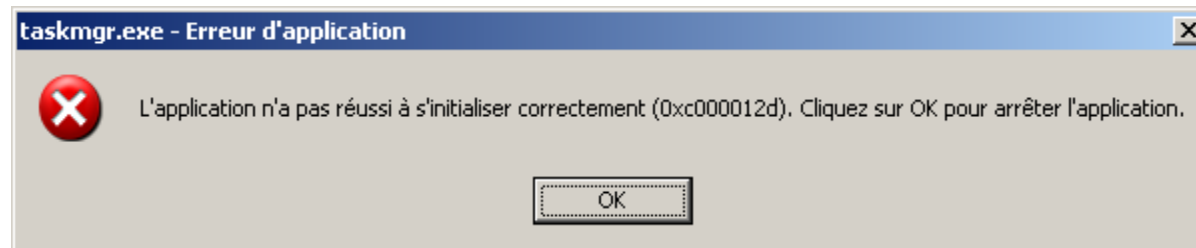




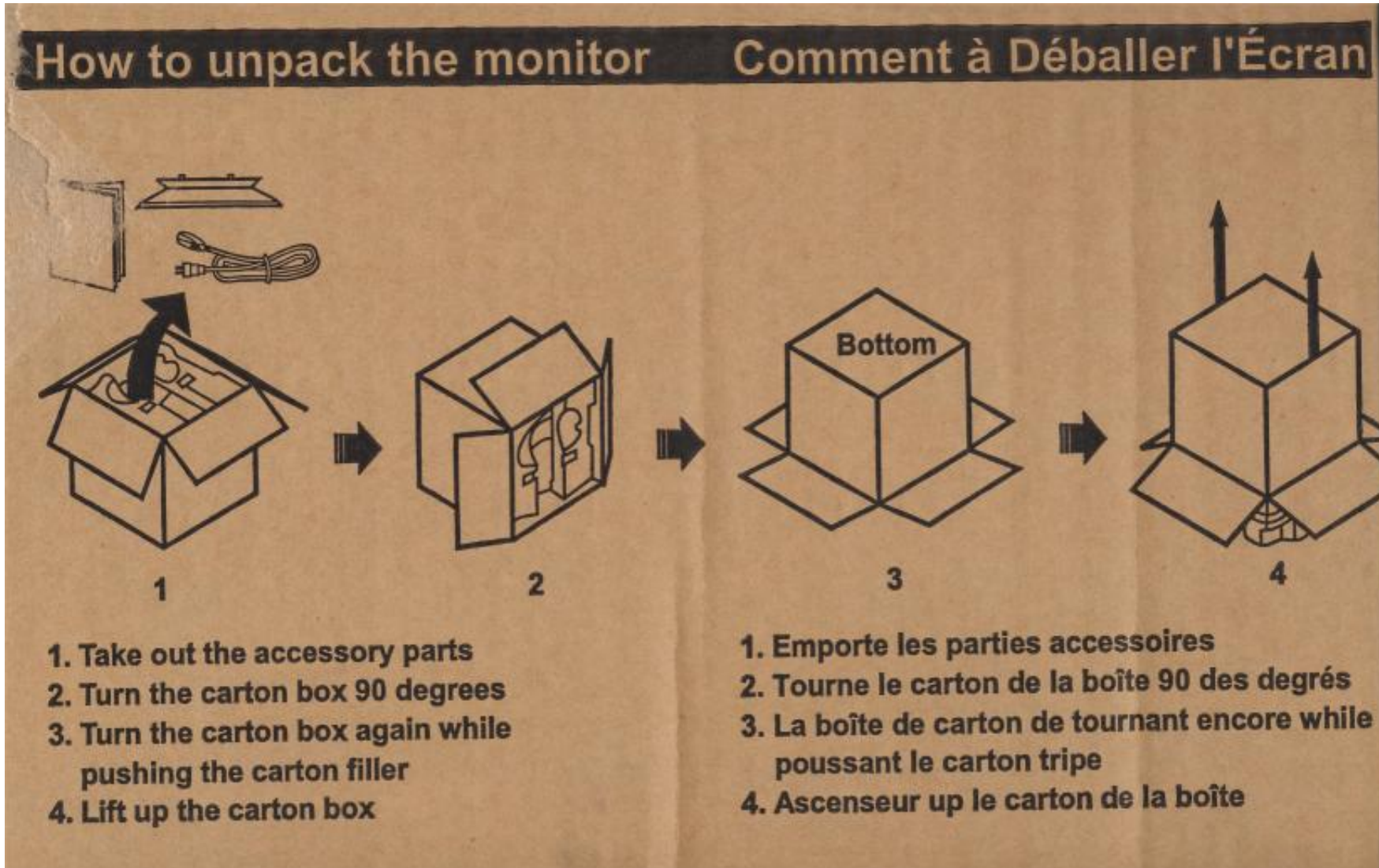
# Gestion des erreurs et récupération

- Exemples d'erreurs Windows

- Plus...



# Internationalisation



# Distance articulaire



# Facteurs Humains

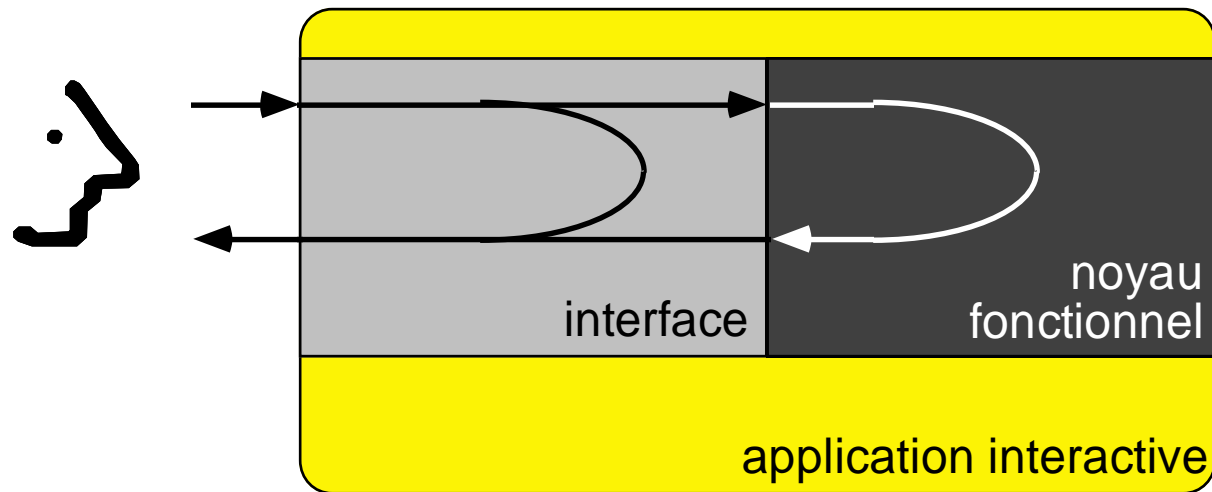
- Selon une étude de l'Université de Cambridge, l'ordre des lettres dans un mot n'a pas d'importance, la seule chose importante est que la première et la dernière soit à la bonne place. Le reste peut être dans un désordre total et vous pouvez toujours lire sans problème. C'est parce que le cerveau humain ne lit pas chaque lettre elle-même, mais le mot comme un tout.

# Quelques définitions

- ◉ Systèmes Interactifs
- ◉ Interface Homme-Machine
- ◉ Communication Homme-Machine
- ◉ Modèle d'interaction
- ◉ Style d'interaction
- ◉ Métaphore d'interaction
- ◉ Dialogue, noyau fonctionnel



# Systèmes interactifs



Interface => Communication - Canal, Langage, Sens

Homme => Tâche - Prise en compte de facteurs humains

Ordinateur => Processus - Aspects logiciels

# Communication Homme-Machine

- Canaux de communication
  - Entrée/sortie: 2 Plans différents
    - E: Textuel: Langage de commande
    - Gestuel: Souris, Interface Graphique
    - Vocal, Gestuel évolué, Multi modal
  - S: Ecran, Ecran Graphique, Synthétiseur...
- Langage = Mode d'interaction
  - Plusieurs Méthodes d'expression pour un même média
- Objectif de la communication
  - Exécuter une tâche = Entrer des commandes et interpréter des sorties

# Modèle d'interaction

- Modèle d'interaction : Aspect morphologique de l'interaction
- Style d'interaction : Aspect morpho-lexical et morpho-syntaxique

=> choix d'utilisation des périphériques

- Compromis entre plus naturel et plus simple à implanter
- Exemple: interfaces iconiques
  - choix objet-verbe ou verbe-objet...
  - cohérence inter-applications

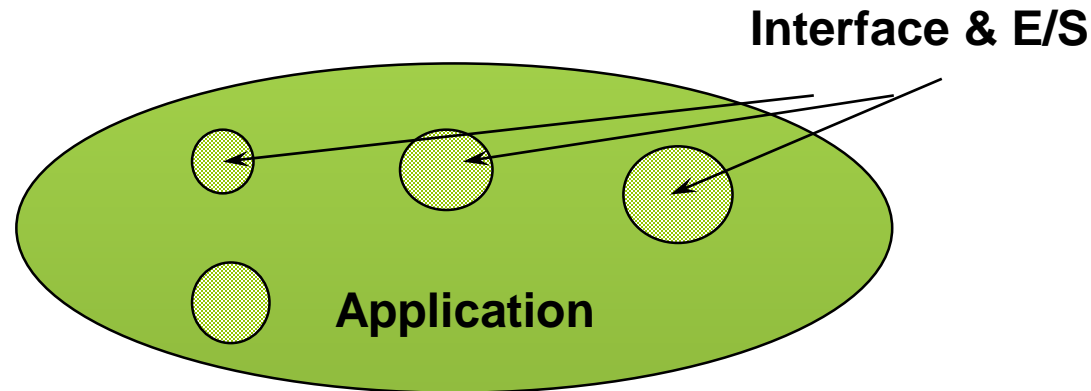


# Modèle d'interaction

- ◉ Modèle d'interaction : Look (qu'est-ce que l'on montre)
- ◉ Style d'interaction : Feel (comment on manipule)
- ◉ Métaphore d'interaction
- ◉ Exemples :
  - ◉ Représentation des objets sous forme iconique
  - ◉ Construction d'une commande (action-paramètre ou paramètre action)
  - ◉ Mode de saisie de la commande :
    - ◉ Drag and drop
    - ◉ Sélection Menu
    - ◉ Langage de commande

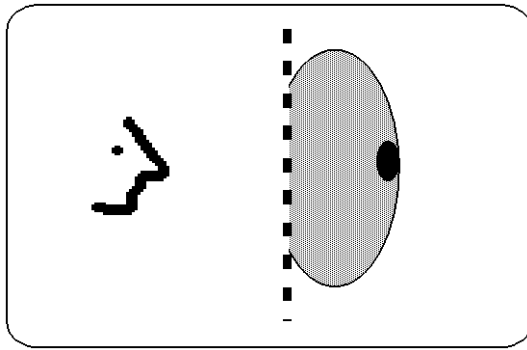
# Séparation Interface/Application

## Application non modulaire

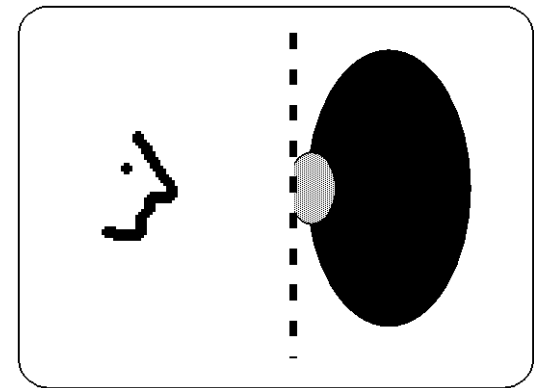


80 % du code d'une application peut être considéré comme faisant partie de la gestion du dialogue (Brad Myers 90)

# Proportion interface/noyau



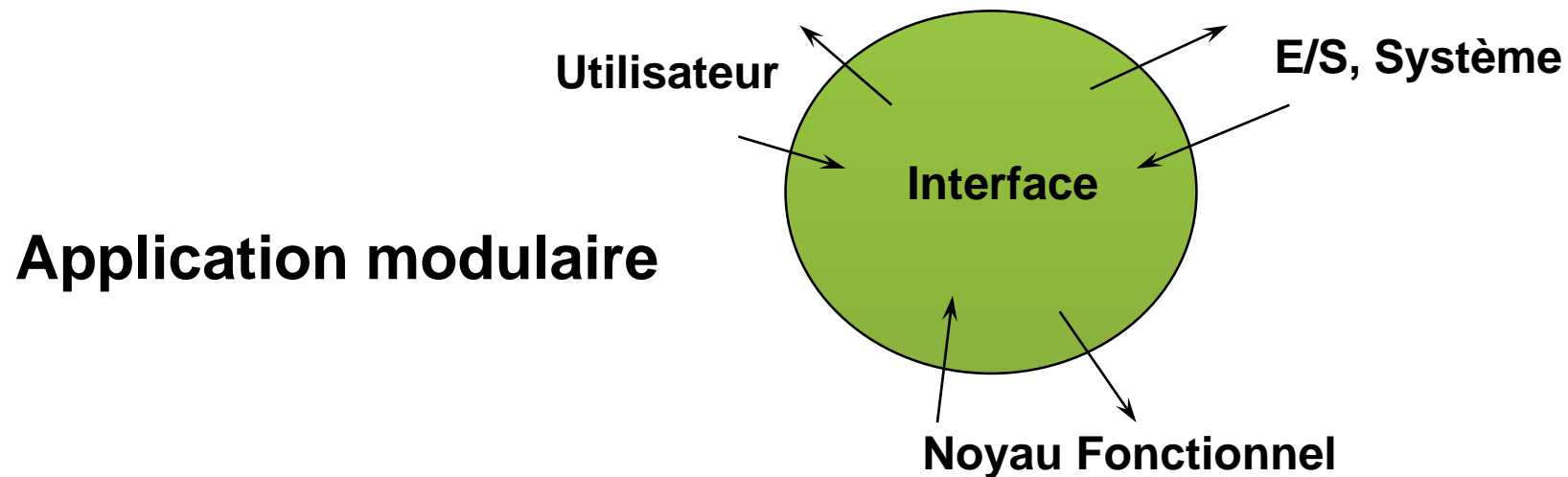
**Editeurs (de textes, de dessins, ...)**



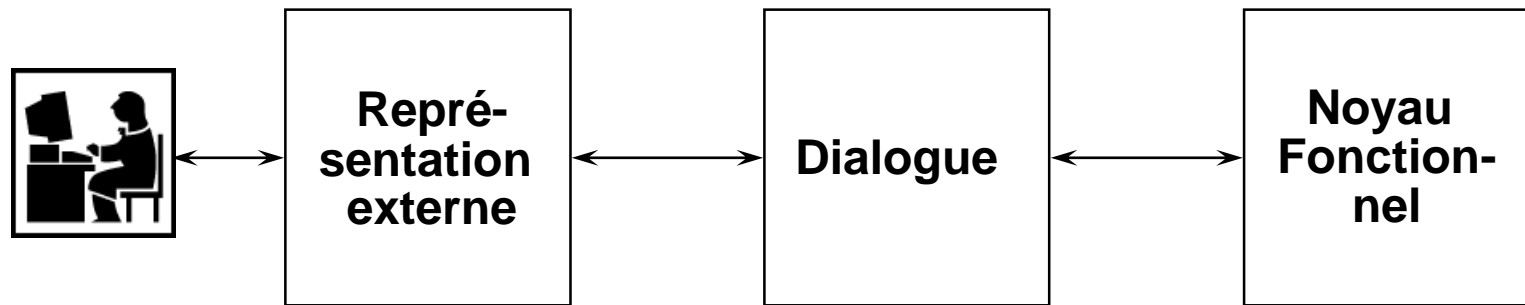
**Code de calcul**

# Séparation interface/noyau

- Au niveau de l'ordinateur, architecture logicielle séparer :
  - Ce qui établit la communication,
  - Ce qui gère le dialogue
  - Ce qui fait les opérations effectives.



# Séparation Interface/Application



Architecture Seeheim  
Un exemple d'architecture favorisant  
la séparation interface / application

# Objectifs d'une interface utilisateur

## **Améliorer la communication**

- Communication "naturelle"
  - Cohérence intra et inter applications
  - Métaphores
- Communication efficace
  - Choix des styles d'interaction
  - Structures de tâches et sous-tâches
  - Étude de la fréquence des commandes
  - Méta-communication

# Caractéristiques d'une bonne IHM

- Adaptée aux besoins
- Facile à apprendre
- Intuitive : prédictible
- Rassurante : apprentissage par essais - erreurs
- Agréable - esthétique - dynamique - ludique
- Performante : tâches rapides à exécuter - réactions immédiates

# Compétences Requises

- Une Approche multi-disciplinaire
  - Informatique
  - Domaine de l'Application
  - Connaissance de l'utilisateur
- Nécessite d'être centré sur l'utilisateur
  - Penser à lui
  - Connaître ses caractéristiques cognitives
  - Connaître son environnement
  - Se mettre à sa place
  - Communiquer avec lui



# Domaines abordés

- Tous et même plus
  - Application de bureau
  - Systèmes temps réel critiques (avion, ATC, ...)
- Systèmes E/S
  - Systèmes standard (souris, clavier, écrans)
  - Systèmes exotiques (tout !!)

# Dangers d'une "mauvaise" IHM

- Rejet pur et simple par les utilisateurs
- Coût d'apprentissage (formation)
- Perte de productivité des utilisateurs
- Utilisation incomplète: manque à gagner
- Coûts de maintenance
- Perte de crédibilité

# Exemples de mauvaise conception

- Libellés non clairs ou ambigus (syntax error...)
- Fonctionnalités inutiles
- Ecrans trop denses, mal structurés
- Ecrans trop nombreux
- "Chemins" trop longs
- Contexte inconnu ou oublié
- Essai en vol (Aéro)

# Conclusion

- Domaine en pleine mutation
  - on sait faire beaucoup
  - il reste beaucoup à inventer
- Enjeux commerciaux très grands
  - besoin de standards
  - les standards sont parfois dangereux