



Interface Homme-Machine

*Processus et méthodologies
de conception*

K. Bouzoubaa





Processus de conception

Conception centrée sur l'*apprenant* et impliquant l'*utilisateur* (CACM, Avril 1996)

Avant 1980



- Conception centrée sur la *technologie*
- Conception centrée sur l'*utilisateur* (*rôle passif*)
- Conception centrée sur la *participation active de l'utilisateur*
- Conception centrée sur l'*apprenant*

Aujourd'hui





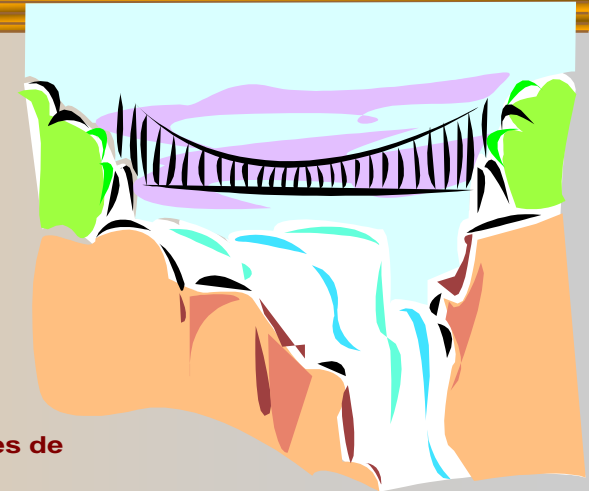
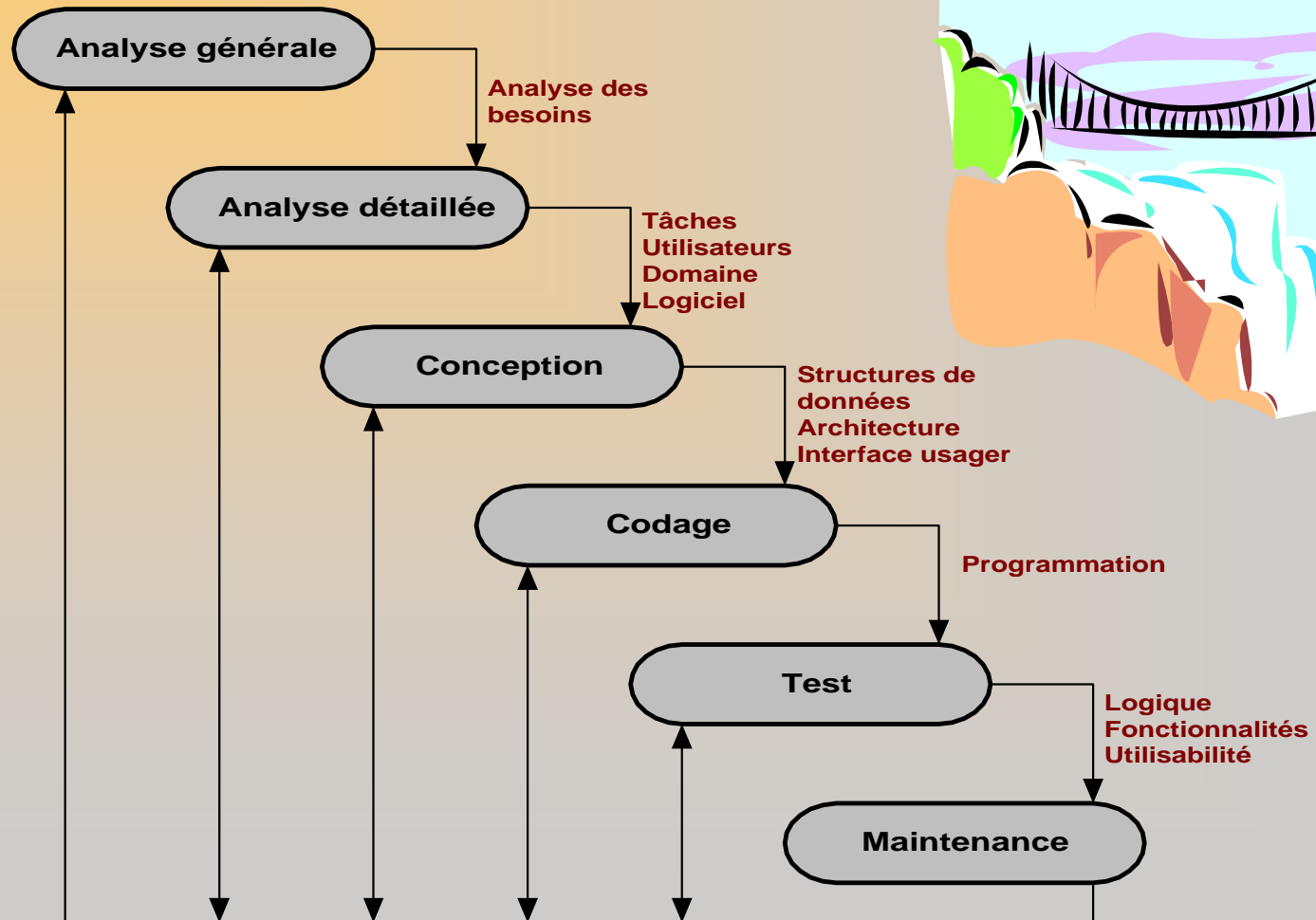
Méthodologies de conception

- ★ Développement *en cascades*
- ★ Le *modèle itératif des 4 phases*
- ★ Cycle de vie basé sur l'*utilisabilité*
(« *Usability lifecycle* ») (J. Nielsen, 1992)
- ★ Méthode de conception *centrée sur l'utilisateur* ISO 13 407, étapes d'analyse
- ★ *Prototypage et design participatif*
 - de basse fidélité (*lo-fi*)
 - de haute fidélité (*hi-fi*)



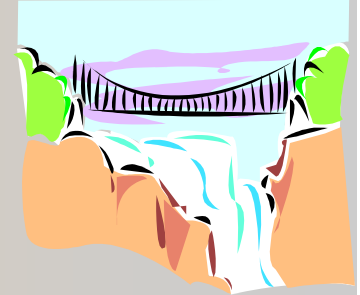


Développement en cascades...





Développement en cascades...



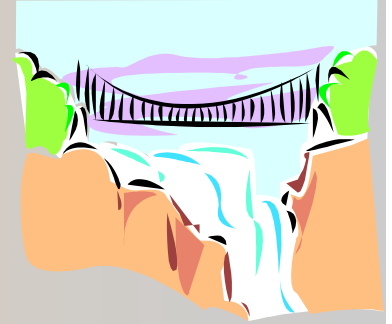
★ Inconvénients (-)

- Les projets réels sont de nature ***non séquentielle*** comme le suppose ce modèle
- ***Fige les besoins*** dès le début (en particulier pour l'***interface usager***)
- Obtention ***trop tardive*** d'une ***version exécutable***
- ***Évaluation trop tardive*** (après l'effort de codage) (**voir acétate suivante**)





Développement en cascades



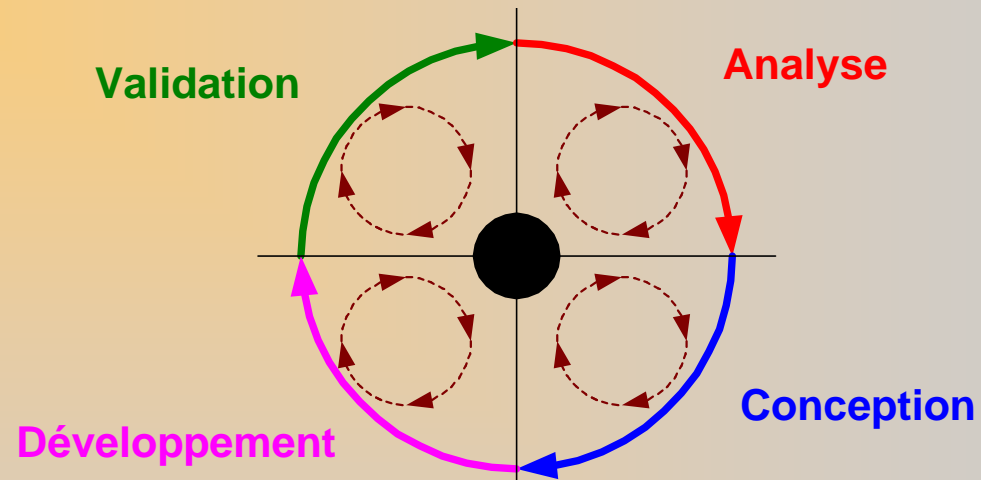
★ **Avantages (+)**

- Étapes *génériques* dans la plupart des paradigmes du *Génie logiciel*
- Modèle *le plus couramment utilisé* dans les projets de grande envergure
- Préférable à tout autre *modèle aléatoire*
- Permet la *gestion de la production* du logiciel





Le modèle itératif des 4 phases (B. Boehm, 1988)



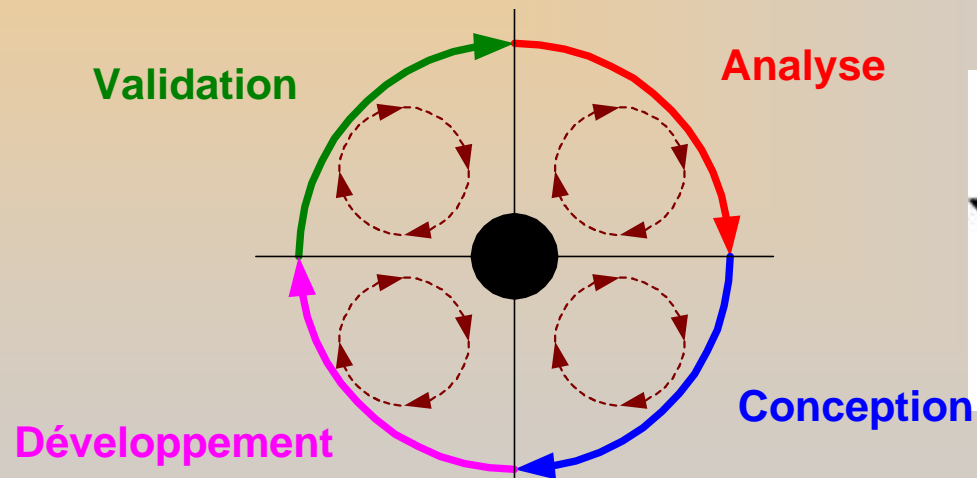
- ★ **Phase 1 : Collecter et analyser l'information usager**
- ★ **Phase 2 : Concevoir l'interface usager**
- ★ **Phase 3 : Développer l'interface usager**
- ★ **Phase 4 : Valider l'interface usager**



Conception itérative/participative

★ Conception itérative

- Les utilisateurs prennent un rôle actif dans la conception





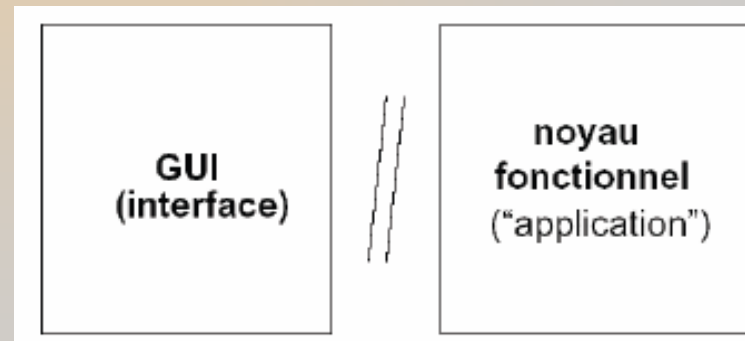
Conception Logicielle

★ **Modèle conceptuel**

- Modèle de fonctionnement du système
- **Idéalement: correspond au modèle mental de l'utilisateur**

★ **Principes de base de conception des IHMs**

- Pas toujours réalisable en toute rigueur
 - **Exemple: traitement de texte**



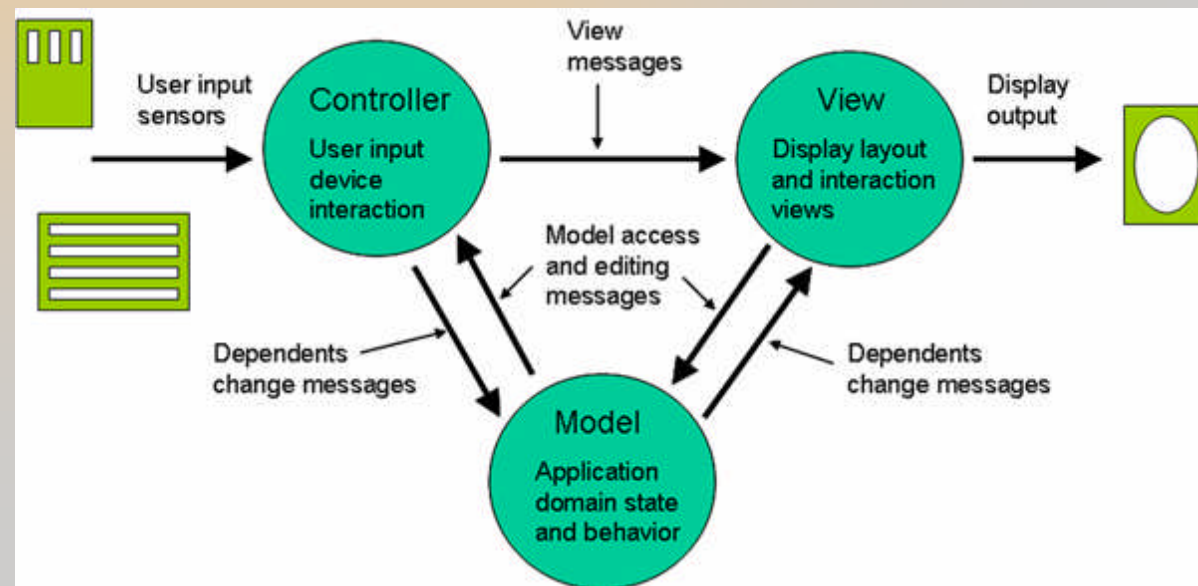
★ **Partage de données entre GUI et NF**

- **Modèle MVC**



Model-View-Controller Pattern

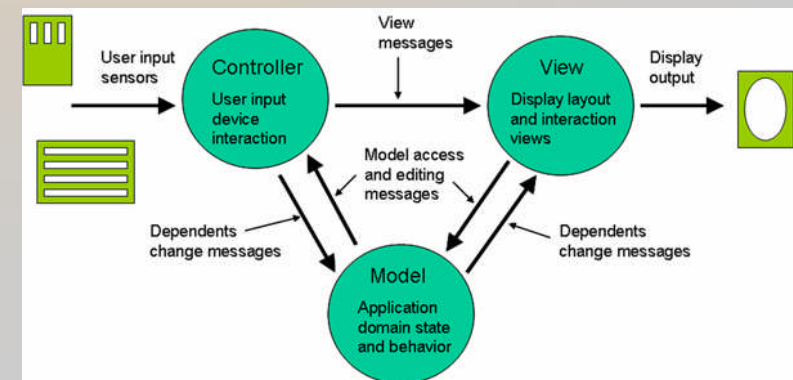
- ★ Goal: better structure of applications
- ★ Three components
 - **Model:** Data
 - **View:** Visual Representation
 - **Controller:** Date Input management





MVC Model

- ★ Separates **frontend** concerns from **backend**
- ★ Separates **input** from **output**
- ★ Permits **multiple views** on the same application data
- ★ Permits views/controllers to be **reused** for other models
- ★ Example: text box
 - Model: mutable string
 - View: rectangle with text drawn in it
 - Controller: keystroke handler

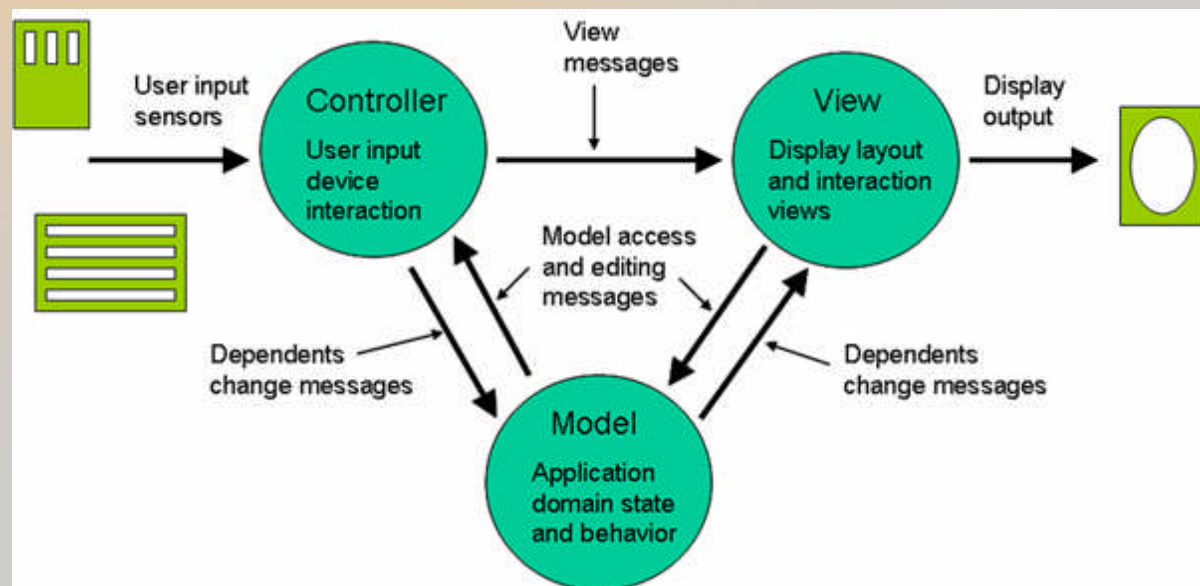




Model

★ Responsible for data

- Maintains application state (data fields)
- Implements state-changing behavior
- Notifies dependent views/controllers when changes occur





View

- ★ Responsible for output
 - Occupies screen extent (position, size)
 - Draws on the screen
 - Listens for changes to the model
 - Queries the model to draw it
- ★ A view has only one model
 - But a model can have many views





Controller

★ Responsible for input

- Listens for keyboard & mouse events
- Instructs the model or the view to change accordingly
 - e.g., character is inserted into the text string

★ A controller has only one model and one view

