



# **Modèles d'architecture et liens avec les outils de production d'interface H/M**

Présenté par SALL Ndeye Bineta – M2PLS

# SOMMAIRE

- Introduction
- Modèles d'architectures
  - Modèles à couches
  - Modèles multi-agents
  - Modèles hybrides
- Outils de génération d'interfaces
- Conclusion

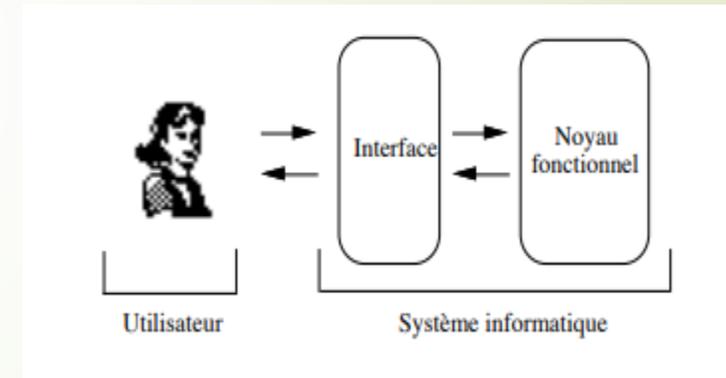
# INTRODUCTION

Vue tournée sur l'**organisation interne** et le **découpage d'un logiciel en modules**. Il décrit:

- ▶ la nature des différents modules d'un logiciel
- ▶ la nature des relations entre les module

Elle donne une première série de réponses sur comment sera le futur logiciel, avant le début du travail de programmation.

Tous les modèles partent du principe qu'un système interactif comporte une partie « **interface** » et une partie « **application pure** »



# Modèles d'architectures

On distingue communément deux grands groupes de modèles de référence :

- ▶ **Modèles linguistiques ou modèles à couches**
  - ▶ décrivent la structure globale d'**une application interactive sous forme de couches logiques**.
- ▶ **Modèles à agents encore modèles orientés objet**
  - ▶ s'inspirent de programmation par objets, et proposent une **décomposition modulaire de l'interface en un ensemble d'agents communicants**

Par ailleurs, des **modèles mixtes** tel que PAC-Amodeus tentent de tirer partie des avantages respectifs de ces deux approches.

# Modèles à couches

Ils se basent sur une **approche linguistique** de l'interaction, inspirée des architectures de compilateurs.

Il identifie trois aspects dans l'interaction :

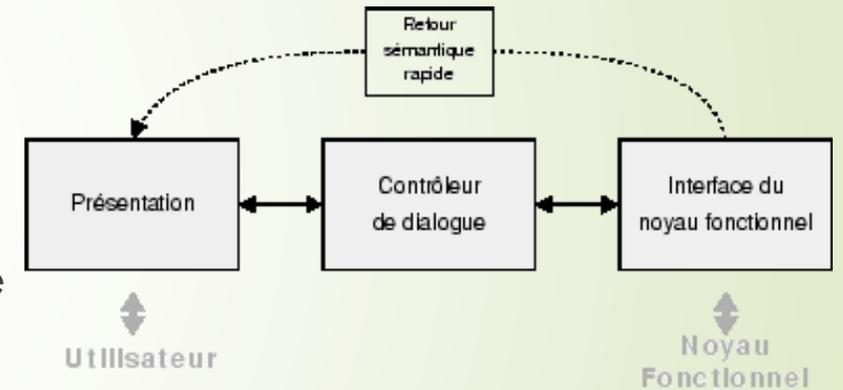
- ▶ **Aspects lexicaux** : tout ce qui peut être assimilé à un **vocabulaire d'entrée ou de sortie**
- ▶ **Aspects syntaxiques** : grammaires d'entrée représentant les **séquences d'actions valides**, ou les aspects spatiaux et temporels de l'affichage (construction des phrases)
- ▶ **Aspects sémantiques** : correspondent à la partie **fonctionnelle de l'application**, qui détermine en dernier lieu le sens d'une action et génère les erreurs (signification des phrases)

# Modèles à couches (Seeheim)

Issu d'un groupe de travail sur les systèmes interactifs ayant eu lieu à Seeheim (Allemagne) en 1985.

Il stipule qu'une interface homme-machine est composée des parties:

- **Présentation** : couche qui gère les entrées et les sorties
- **Contrôleur de Dialogue** : gère le séquençement des entrées et des sorties
- **Interface du noyau fonctionnel** : Modèle d'Interfaçage de l'Application



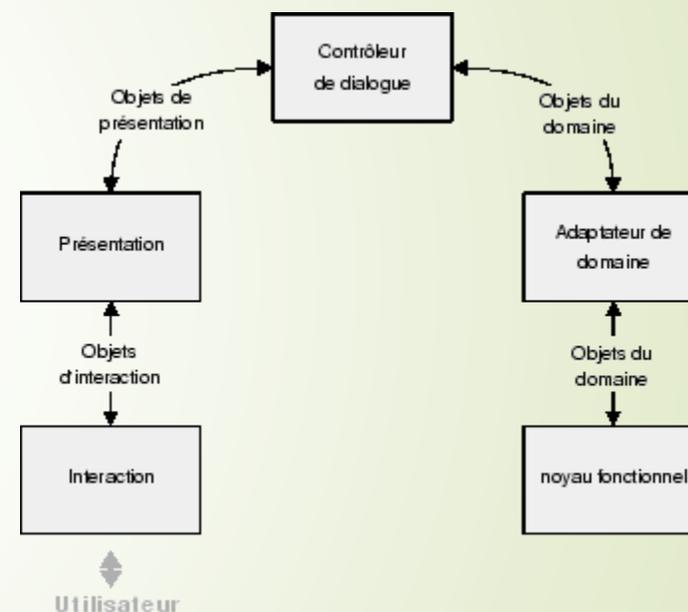
# Modèles à couches (Arch)

7

Introduit en 1992 pour structurer les logiciels interactifs.

Affine le modèle de Seeheim en s'inspirant davantage des boîtes à outils graphiques actuelles

- **Le noyau fonctionnel** réalise les concepts du domaine
- **Le composant d'interaction**, en contact direct avec l'utilisateur, est mis en œuvre au moyen des objets d'interaction d'une boîte à outils.
- **L'adaptateur de domaine** sert, pour l'essentiel, à ajuster les différences de modélisation des objets conceptuels entre le noyau fonctionnel et le contrôleur de dialogue.



# Modèles Multi-agents

- Système composé d'un **ensemble d'agents interagissant entre eux**
- Agent : entité partiellement autonome

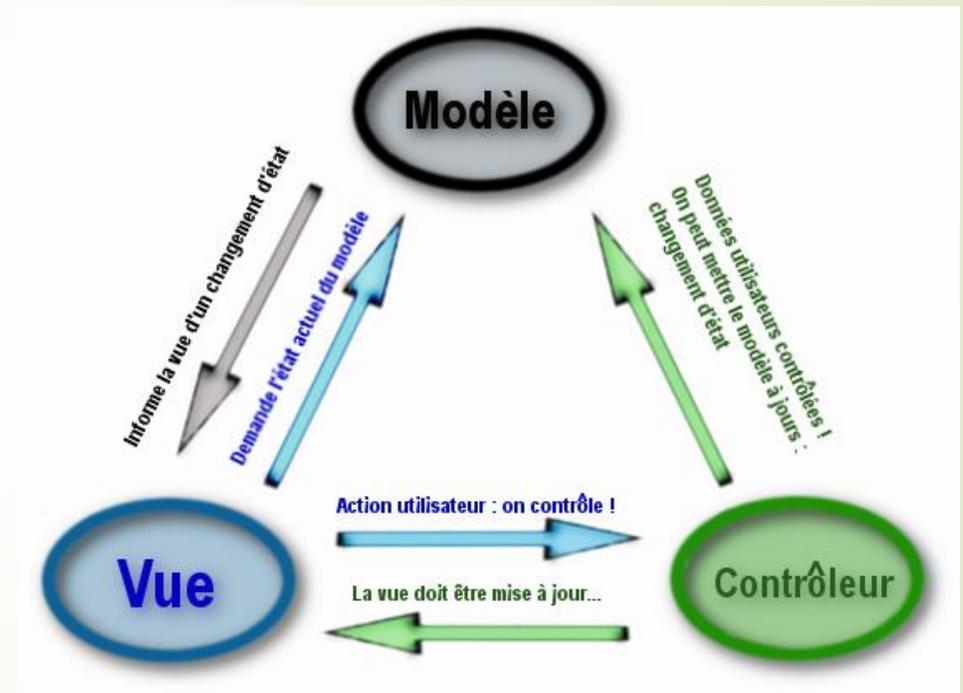
Les modèles multi-agents procèdent donc à une **décomposition de l'application en plusieurs entités autonomes et coopératives**.

Cette nouvelle décomposition permet **d'accélérer les retours de l'application à l'utilisateur**, indispensables pour la manipulation directe.

# Modèles Multi-agents (MVC)

Issu des travaux de Trygve Reenskaug en 1978-79.

But principal : proposer une solution générale aux problèmes d'utilisateurs manipulant des données volumineuses et complexes



# Multi-agents (Modele PAC)

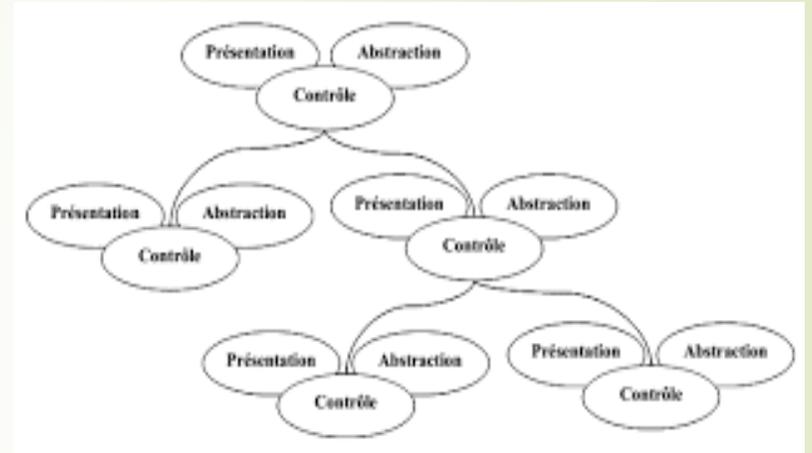
10

Introduit par la chercheuse en informatique grenobloise Joëlle Coutaz dans les années 1980

Décompose l'application interactive en une **hiérarchie d'agents interactifs** structurés en trois composants:

- **Présentation** est chargé de la transmission (visuelle, sonore, etc.) des informations à l'utilisateur et de l'acquisition des entrées de l'utilisateur ;
- **Abstraction** contient les données et les traitements sémantiques
- **Contrôle** est le moteur des dialogues qui s'établissent entre l'agent et l'utilisateur, et entre l'agent et les autres agents.

PAC permet un **dialogue multi-fils** (plusieurs dialogues en parallèle) grâce à l'autonomie relative des agents dans l'interprétation des événements et dans la gestion de leur état.



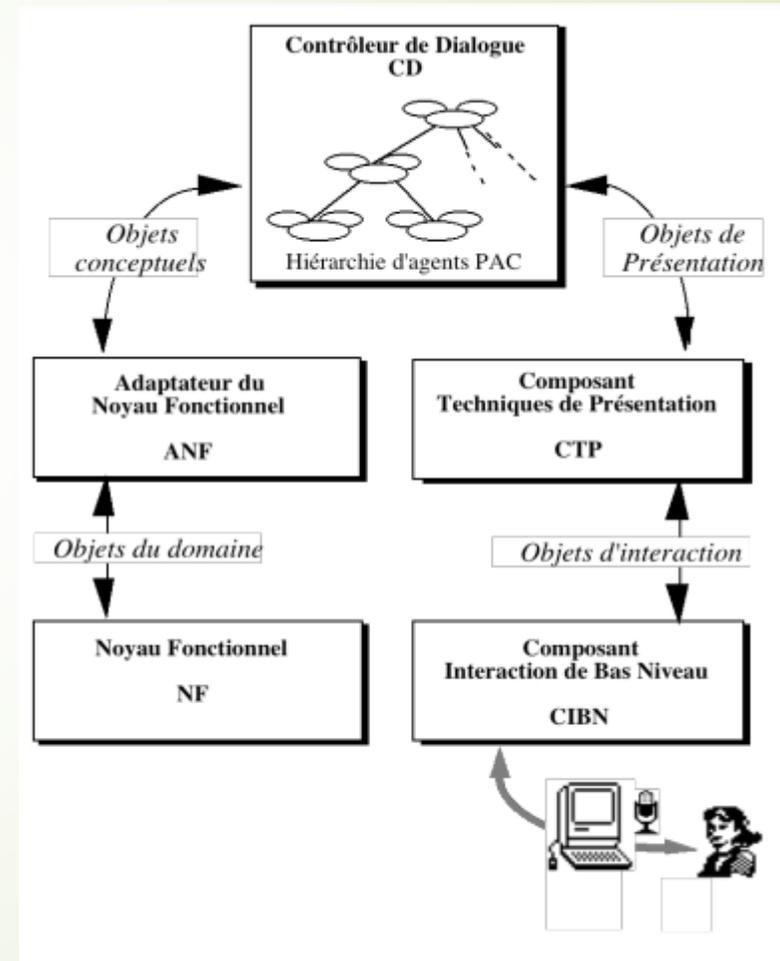
# Modèles Hybrides

Dans certains cas, il apparaît souhaitable de combiner les modèles à couches avec les modèles multi-agents pour essayer de tirer partie de leurs avantages respectifs.

# Modèles Hybrides PAC-Amodeus (Arch + PAC)

Reprend le découpage en couches de Arch tout en exprimant le composant contrôleur de dialogue sous la forme d'agents PAC.

Il peut ainsi facilement exprimer le parallélisme de contrôle des différentes modalités.



# Outils de génération d'interfaces

- ▶ **Le code de l'IHM est produit par l'outil** et l'architecture globale est implicitement véhiculée par le générateur d'IHM
- ▶ Interface Builder
- ▶ XFaceMaker

# Conclusion

La construction d'une IHM dédiée, dotée d'interacteurs innovants comme l'interaction multimodale, exige de définir une architecture globale précise.

D'où le lien entre modèles d'architectures et outils de générations d'interfaces.

# SOURCES

- ▶ <http://iihm.imag.fr/tarpin/publis/tsi99.pdf>
- ▶ <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.5295&rep=rep1&type=pdf>