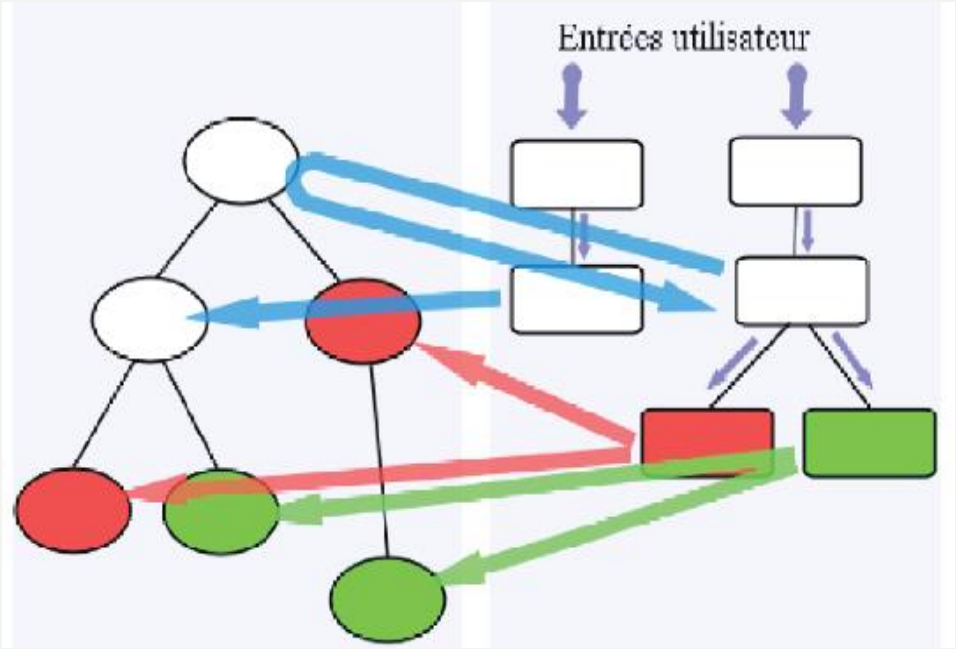
Centre Universitaire Abdelhafid Boussouf - Mila

3éme INF



INTERACTION HOMME - MACHINE

Chapitre3

Modèles & Architectures

2023 - 2024

PLAN

III.1- Le Contrôleur de dialogue (définition & rôle).

III.2- Présentation du modèle Seeheim

III.3- Présentation du modèle PAC

III.4- Présentation du modèle MVC

INTRODUCTION

Constat :

- conception d'IHM difficile, par conséquent itérative (itérativité implique modifiabilité du logiciel)
- complexité et taille croissante des IHM
- outils de développement des IHM imparfaits (boîtes à outils, squelettes d'application, générateurs d'interface)

Conséquence :

On a besoin d'un cadre de pensée.

L'architecture est abstraite, elle décrit des composants et leurs relations sans présumer de leur réalisation. Il existe des architectures éprouvées qui sont des modèles de référence.

LES MODELES D'ARCHITECTURE

- Définissent l'organisation logicielle d'un système interactif.
- **Principe de base**

la séparation entre le **noyau fonctionnel** qui implémente les concepts propres à un domaine d'application particulier et **l'interface** qui présente ces concepts à l'utilisateur et qui lui permet de les manipuler.

- **Rôle**
- Il doit permettre en théorie de modifier l'interface sans affecter le noyau fonctionnel et inversement, est en réalité difficile à concrétiser pour la totalité de l'interface.

En pratique et dans la majorité des cas, seule une partie de l'interface est réellement séparée du noyau fonctionnel.

- Il permet de structurer un système interactif et ainsi de disposer d'une meilleure modularité (primordiale vue la méthode itérative de construction des interfaces. Elle facilite la réutilisation logicielle des composants du système interactif, sa maintenance et son évolution).

UNE ARCHITECTURE, POURQUOI FAIRE ?

- Organiser le code (rangement)
- Simplifier (diviser pour régner)
- Organiser le travail
  - Itératif
  - Parallèle (fusion)
- Modifier (une partie)
- Ré-utiliser

Notion : modularité, évolutivité, flexibilité

IHM ET ARCHITECTURE

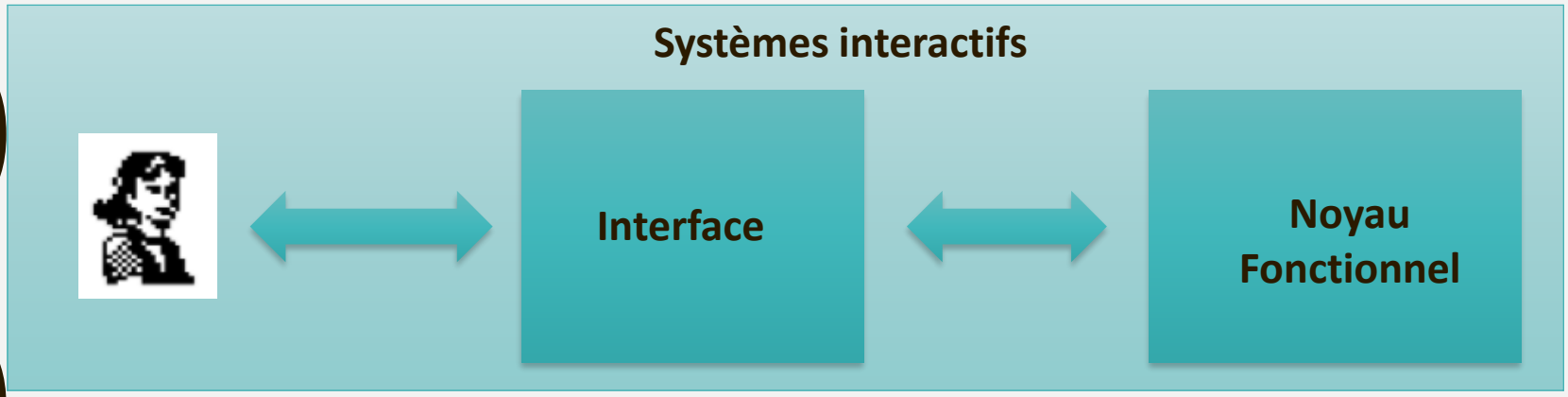
- Séparation possible
  - Code pour IHM
  - Code « Métier » (Fonctions, Objets, Services)

- **Objectif :**

éviter de tout modifier si on change la partie fonctionnel ou la partie IHM

SYSTÈMES INTERACTIFS

- Tous les modèles partent du principe : un système interactif comporte une partie **interface** et une partie **application pure**
- Cette dernière est souvent appelée **noyau fonctionnel**



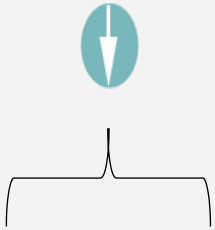
ARCHITECTURES LOGICIELLES

La plupart des modèles identifient en général trois types d'éléments :

- un ``coté utilisateur'‘ (présentations, vues),
- un ``côté noyau fonctionnel'‘ (interfaces du noyau fonctionnel, abstractions, modèles),
- et des éléments articulatoires (contrôleurs, adaptateurs).

DÉFINITION :ARCHITECTURE LOGICIELLE

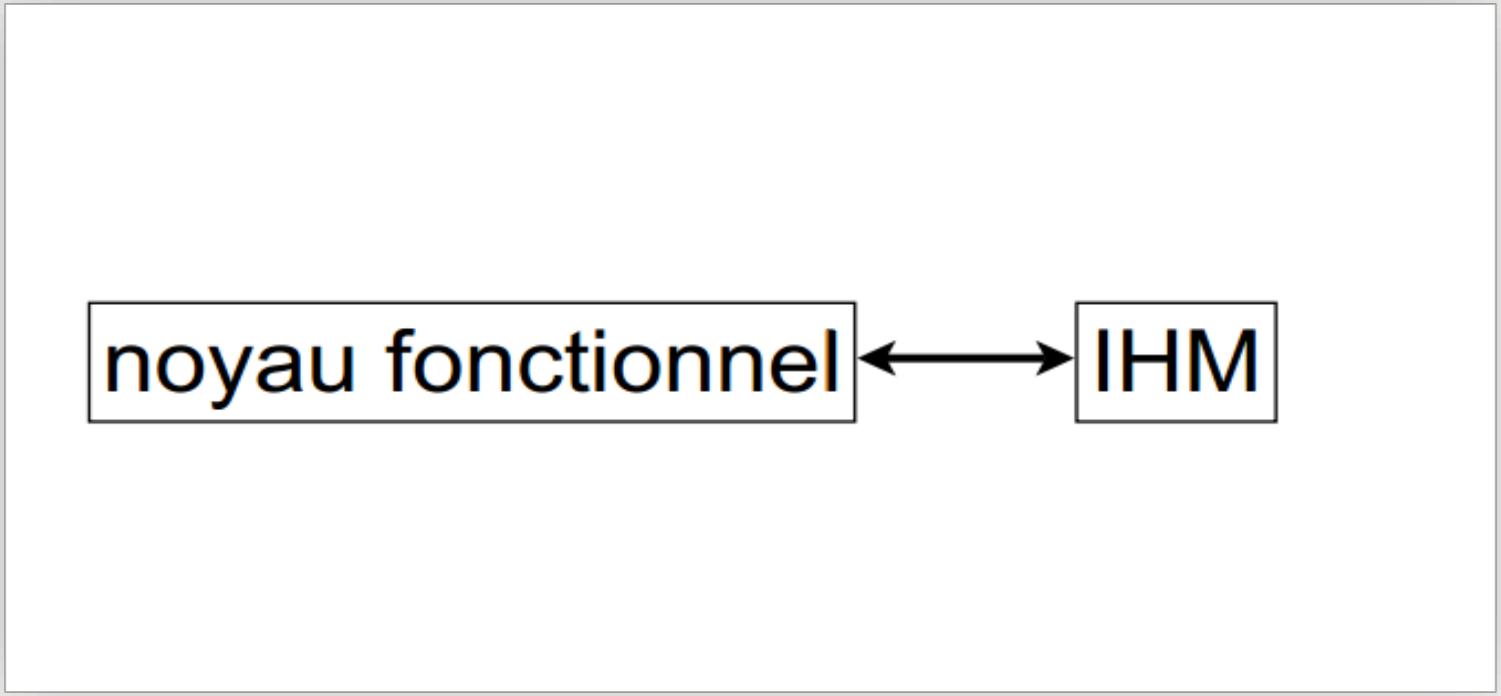
- **Décrit d'une manière symbolique et schématique les différents élément d'un ou plusieurs systèmes informatiques ,leurs interrelations et leurs interactions.**



Modele a couches

Modele a agents

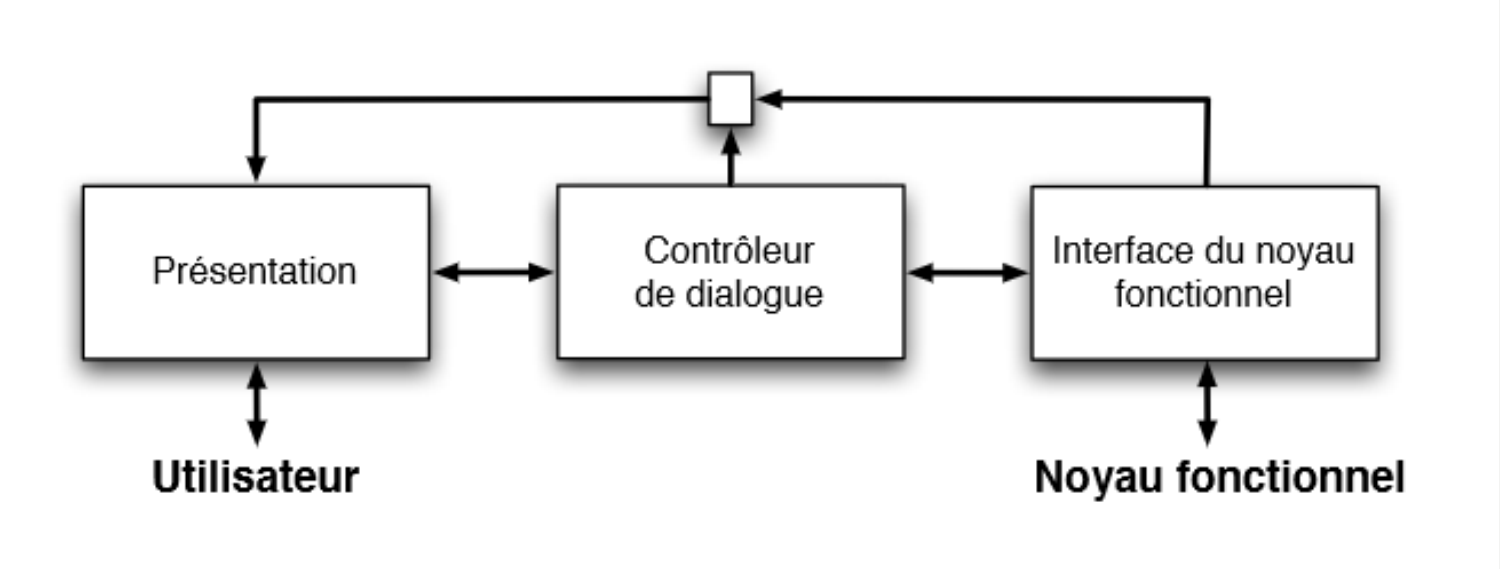
ARCHITECTURE FONDAMENTALE



MODÈLE À COUCHES

MODÈLE DE SEEHEIM

PRÉSENTATION DU MODÈLE SEEHEIM



Présentation	**Définit le comportement du système tel qu'il est perçu par l'utilisateur.**
Contrôle de dialogue	**Médiateur entre l'interface du noyau fonctionnel et la présantation**
Interface d'application	**Décrit la sémantique de l'application du point de vue de l'interface utilisateur.**

PRÉSENTATION DU MODÈLE SEEHEIM

Le composant de présentation

- la production effective des sorties (affichage, production de sons...) à partir des informations reçues du contrôleur de dialogue,

- la réception et du traitement (d'un premier niveau) des événements d'entrée induits par les actions de l'utilisateur pour ensuite transmettre les informations résultantes au contrôleur de dialogue.

Le contrôleur de dialogue

- **où est** déterminée la structure du dialogue entre l'utilisateur et l'application.

- Le contrôleur de dialogue est responsable du **séquencement** des opérations au niveau des actions de l'utilisateur et des appels aux fonctions du domaine

- réalise un pont entre les objets concrets de l'interface (composant de présentation) et les concepts du domaine (noyau fonctionnel).

PRÉSENTATION DU MODÈLE SEEHEIM

- **L'interface du noyau fonctionnel**

  Ce composant sert de passerelle entre le noyau fonctionnel et le contrôleur de dialogue. Il ajuste les différences entre les formalismes utilisés par ces deux composants.

Il contient en général une description des entités du domaine qui sont directement liées à l'interface ainsi qu'une description des procédures du noyau fonctionnel qui peuvent être invoquées par le contrôleur de dialogue.

MODÈLE DE SEEHEIM [PFAFF, 1985]

- **Le noyau fonctionnel**

manipule les objets du domaine d'application.

- **L'interface du noyau fonctionnel**

décrit la sémantique de l'application du point de vue de l'interface utilisateur.

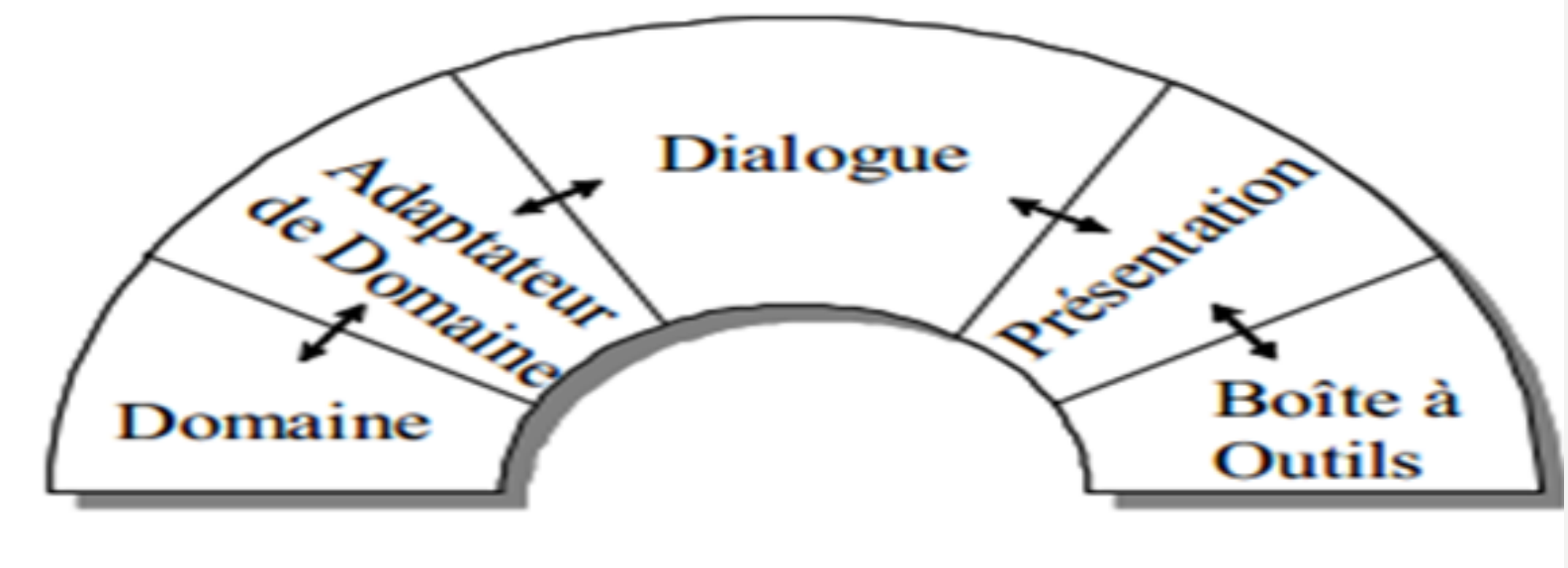
- **La présentation**

Définit le comportement du système tel qu'il est perçu par l'utilisateur et tel qu'il manipule.

- **Le contrôleur de dialogue**

agit comme médiateur entre l'interface du noyau fonctionnel et la présentation.

LE MODÈLE ARCH



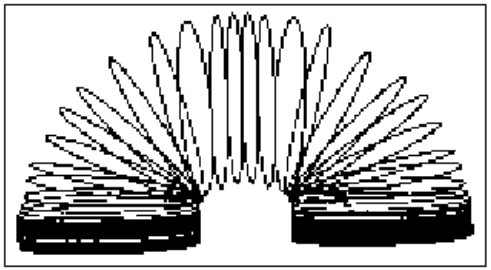
MODÈLE DE L'ARCH [1992]

Le noyau fonctionnel et le composant d'interaction (généralement une boîte à outils) forment les pieds de l'arche, car dans la plupart des cas ces deux composants existent avant le développement même de l'interface et forment ainsi la base de départ.

MODÈLE DE L'ARCH [1992]

- Arch est une version raffinée de Seeheim qui tient compte de l'apparition des boîtes à outils.
- Les pieds de l'arche représentent les éléments préexistants : noyau fonctionnel d'une part et boîte à outils d'interface d'autre part.
- La présentation et l'adaptateur du noyau fonctionnel permettent l'adaptation entre les 3 autres composants.
- Ces adaptations ne sont pas toujours nécessaires : le méta-modèle **Slinky** autorise alors leur disparition ou leur fusion dans les autres éléments de l'arche.

LE MÉTAMODEL S



- Le terme SLINKYTM provient d'un jouet flexible qui une fois mis en mouvement voit son centre de gravité se déplacer suite à un changement dans la répartition de sa masse.
- Cette métaphore est utilisée dans le modèle ARCH pour représenter la modification de la répartition des fonctionnalités à travers les différents composants du modèle, d'une instance à l'autre

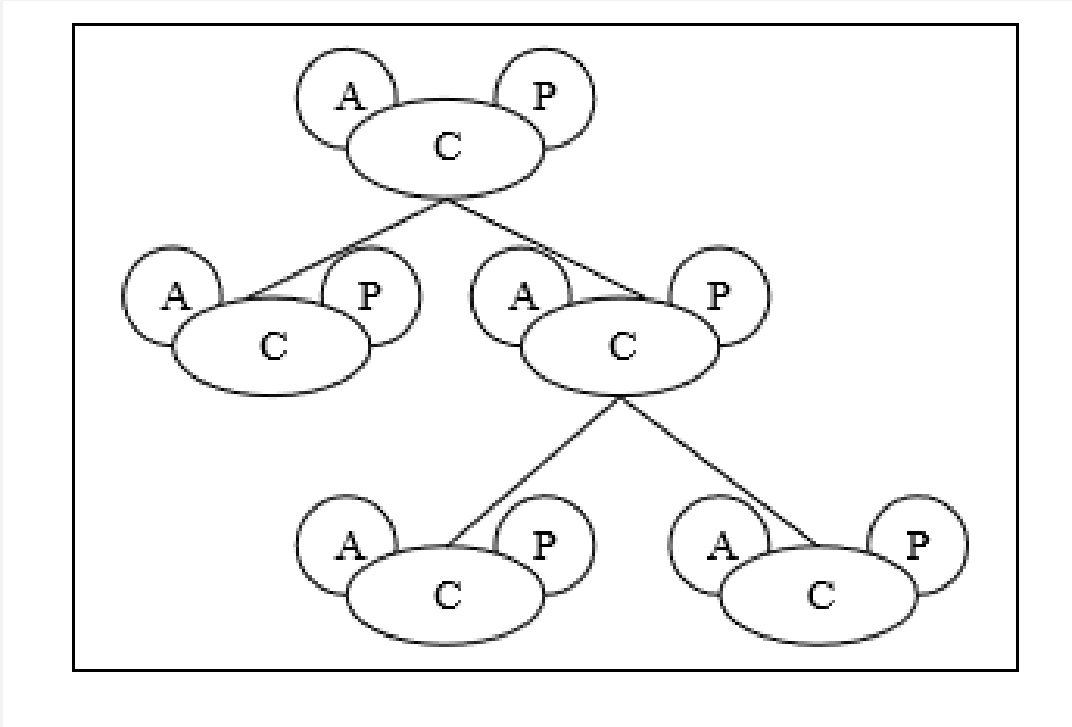
MODÈLE À AGENTS

MODÈLES DE RÉFÉRENCE À AGENT

- **Principe** : Un système interactif est constitué d'une collection d'unités de calcul spécialisées (agents).
- **Un agent** :
  - a un état ;
  - a une expertise ; et
  - est capable d'émettre et de réagir à des événements.
- **Un interacteur** est un agent en contact direct avec l'utilisateur.

MODÈLE PAC [COUTAZ, 1987]

- Il structure l'application interactive de manière récursive sous forme d'une hiérarchie d'agents comprenant chacun trois facettes



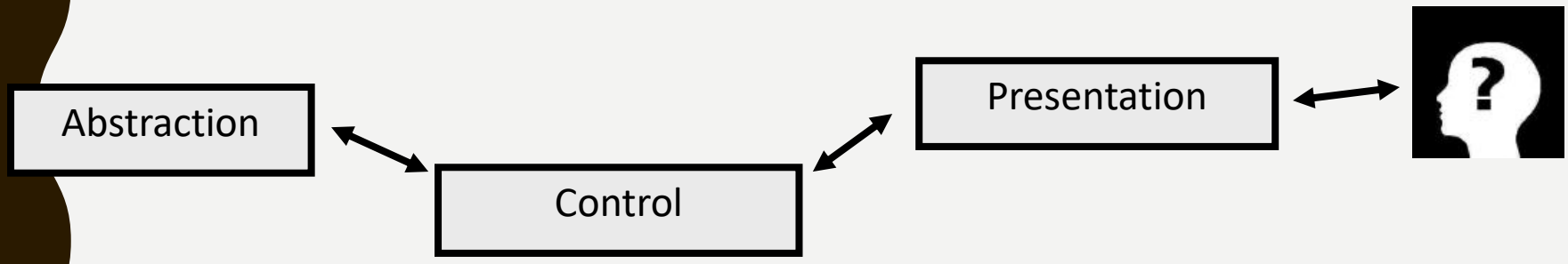
PRÉSENTATION DU MODÈLE PAC

Un agent PAC est composé de trois facettes réalisées par des objets :

.• l'abstraction (le modèle de MVC) ;

• la présentation (la vue et le contrôleur de MVC) ; et

• le contrôle qui exprime les dépendance entre abstraction et présentation et qui gère les échanges avec les autres agents.



MODÈLE PAC [COUTAZ, 1987]

- Le **contrôle** notifie le modèle lorsque les manipulations de la **présentation** par l'utilisateur le nécessitent ; et notifie la **présentation** lorsque les modifications du modèle le réclame.
- Le **contrôle** notifie les facettes contrôle des autres agent PAC de la hiérarchie si besoin.

PRÉSENTATION DU MODÈLE MVC

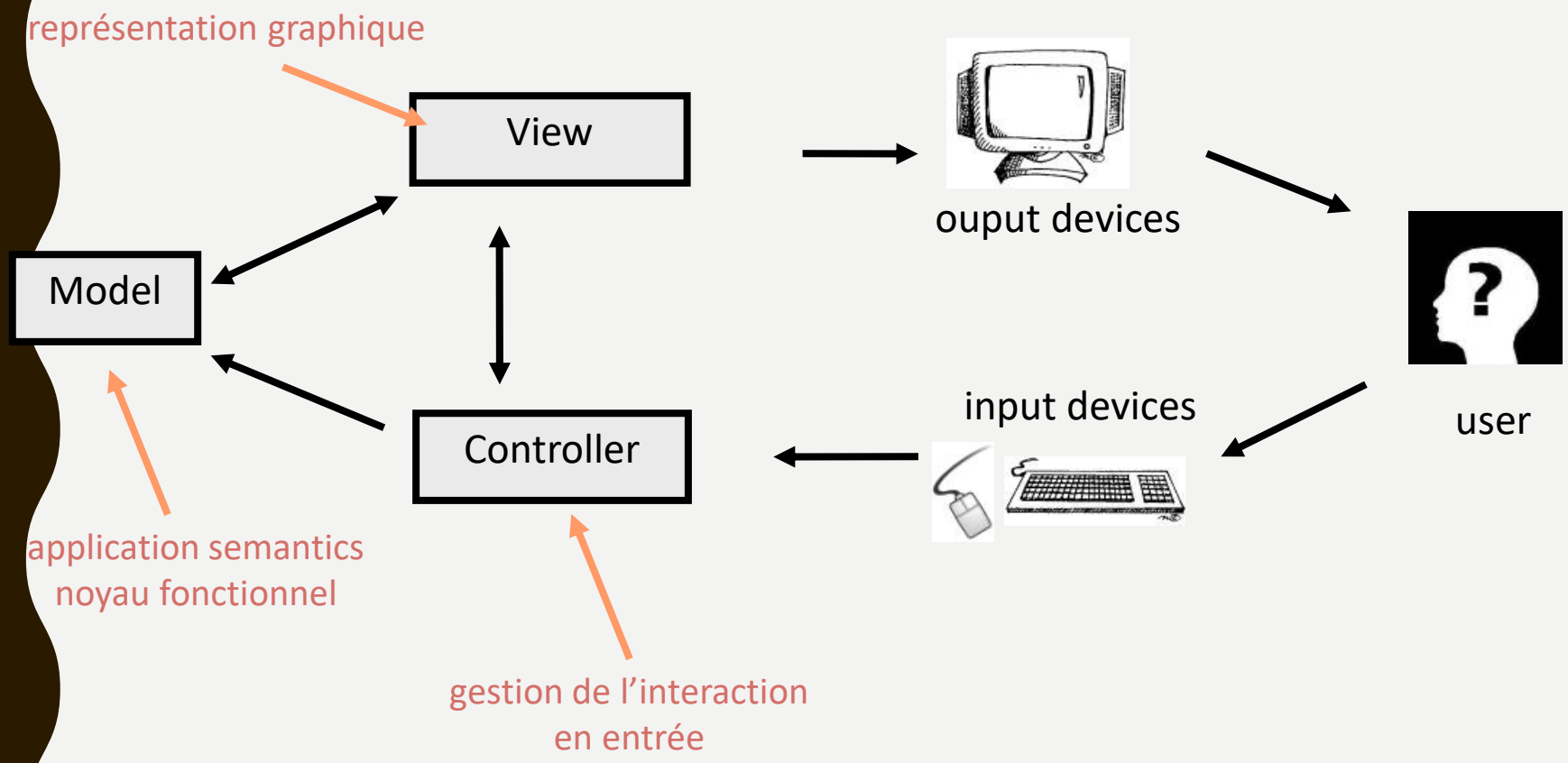
- MVC c’est: Un patron de conception (une solution standardisée à un probléme, indépendante des langages de programmation) Une architecture logicielle (une maniére de structurer une application ou un ensemble de logiciels)
- Introduit en 1979 par Trygve Reenskaug Trés fortement lié aux concepts de la programmation objet (Smalltalk).

PRÉSENTATION DU MODÈLE MVC

- Modèle -Vue -Contrôleur est le modèle multi-agent utilisé par **Smalltalk** [GOLDBERG 84]
- Chacun des trois composantes de la triade MVC est un objet à part entière.
- **Cause** : difficultés de conception des applications fortement interactives
- **Réponse** : modularisation
- **Principe de base**
  - Model : modélisation (données et comportement), maintient son état et notifie de ses modifications
  - View : Vue présente l ’information aux utilisateurs en utilisant les données issues des modèles.
  - Control : Contrôleur se charge de l ’interaction avec l’utilisateur. contrôleur écoute l’utilisateur et demande des modifications au modèle

MODÈLE MVC 1987 (SMALLTALK)

un système interactif en un ensemble d'agents, chaque agent comportant trois facettes



MVC: STRUCTURE DU MODÈLE

- Organiser, structurer une application interactive en séparant:
  - Les données et leurs traitements: **Le Modèle**
  - La représentation des données: **La Vue**
  - Le comportement de l'application: **Le Contrôleur**

- **Le modèle, la vue et le contrôleur communiquent par échange de messages.**

MVC: LE MODÈLE

Noyau Fonctionnel de l'application

- Représente les données
- Fournit les accès aux données
- Fournit les traitements applicables aux données
- Expose les fonctionnalités de l'application

MVC: LA VUE

Sorties de l'application

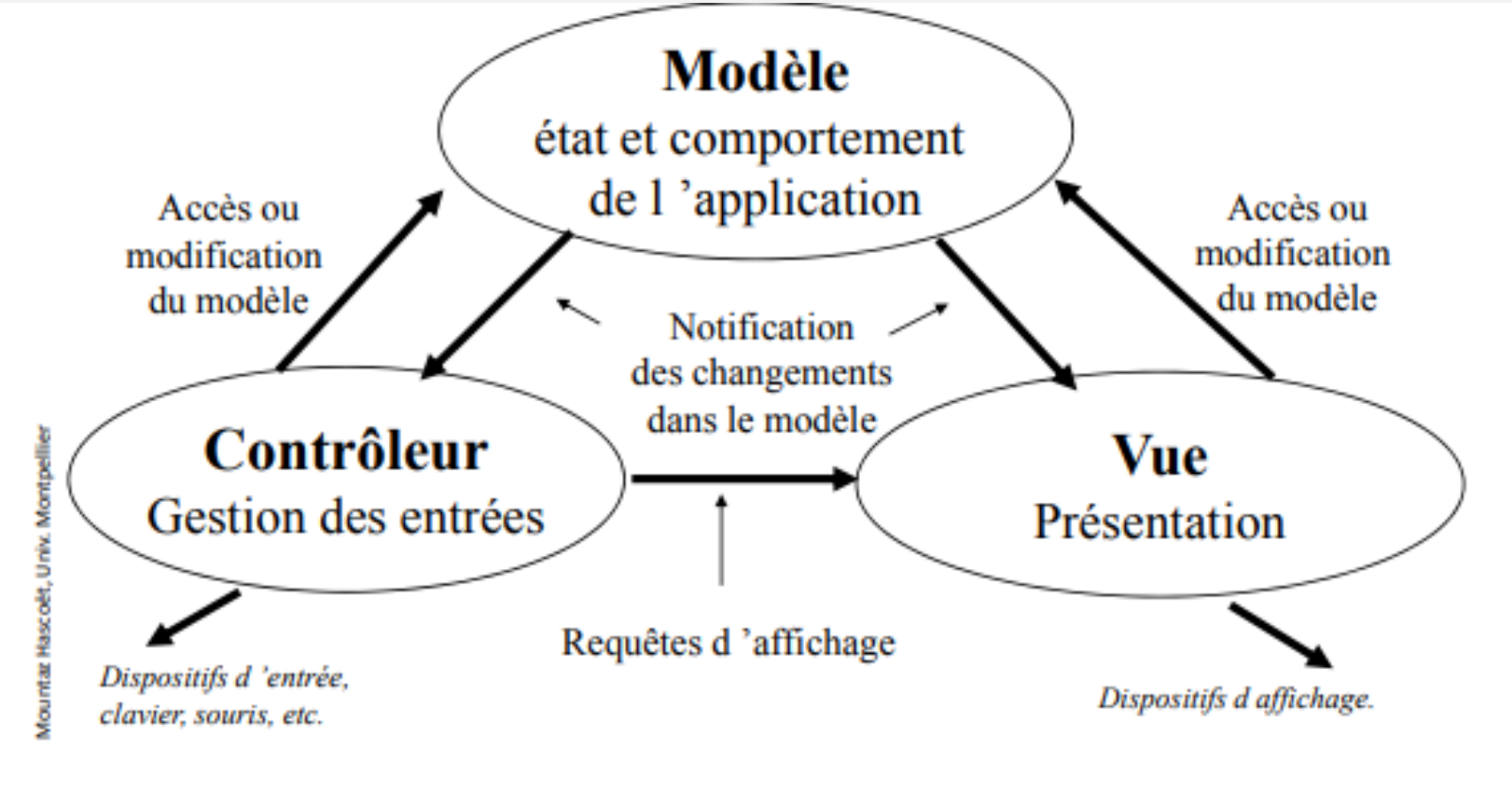
- Représente la (ou une) représentation des données du modèle
- Assure la consistance entre la représentation qu'elle donne et l'état du modèle / le contexte de l'application

MVC: LE CONTRÔLEUR

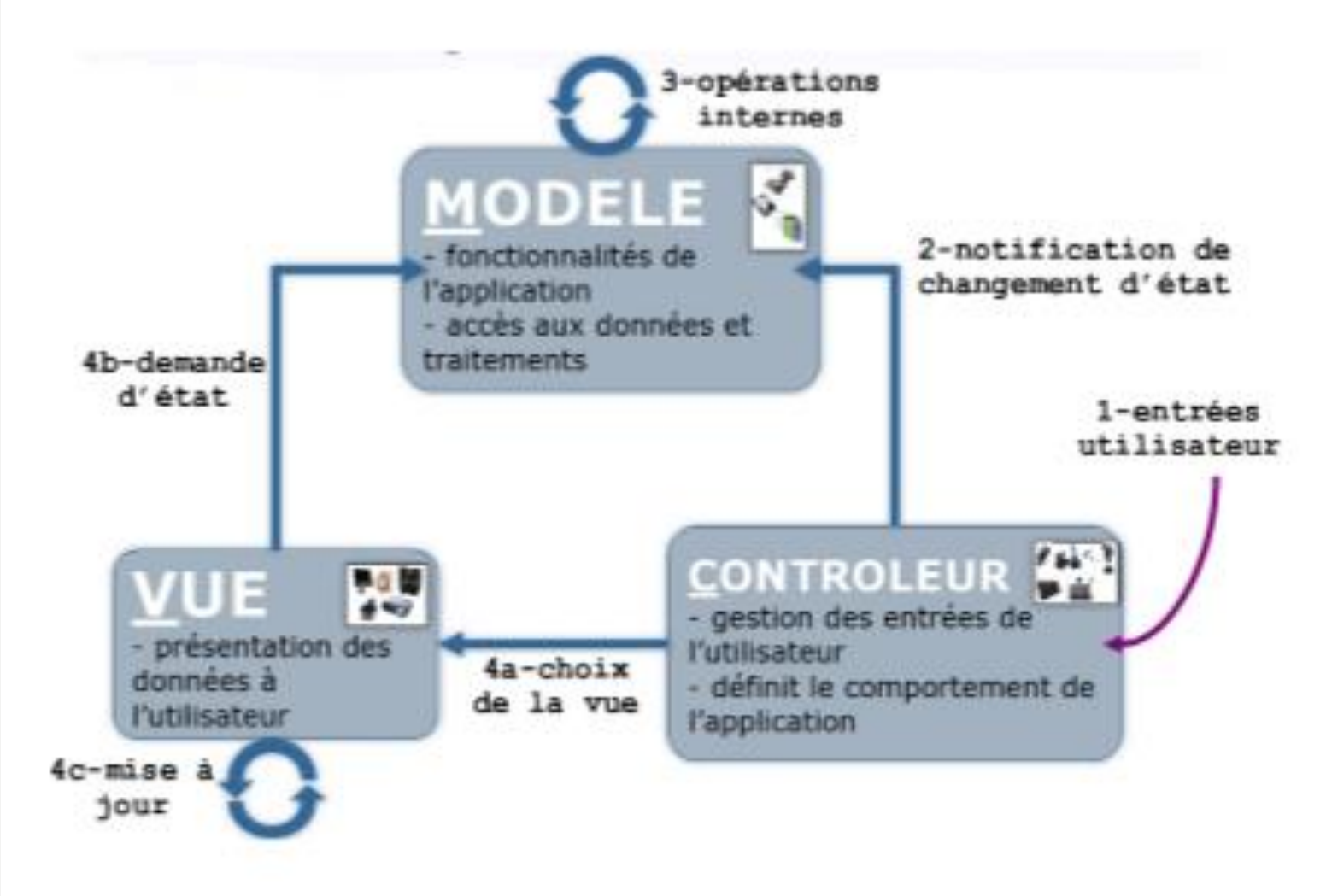
Comportement et gestion des entrées de l'application

- Représente le comportement de l'application face aux actions de l'utilisateur
- Fournit la traduction des actions de l'utilisateur en actions sur le modèle
- Fournit la vue appropriée par rapport aux actions de l'utilisateur et des réactions du modéle

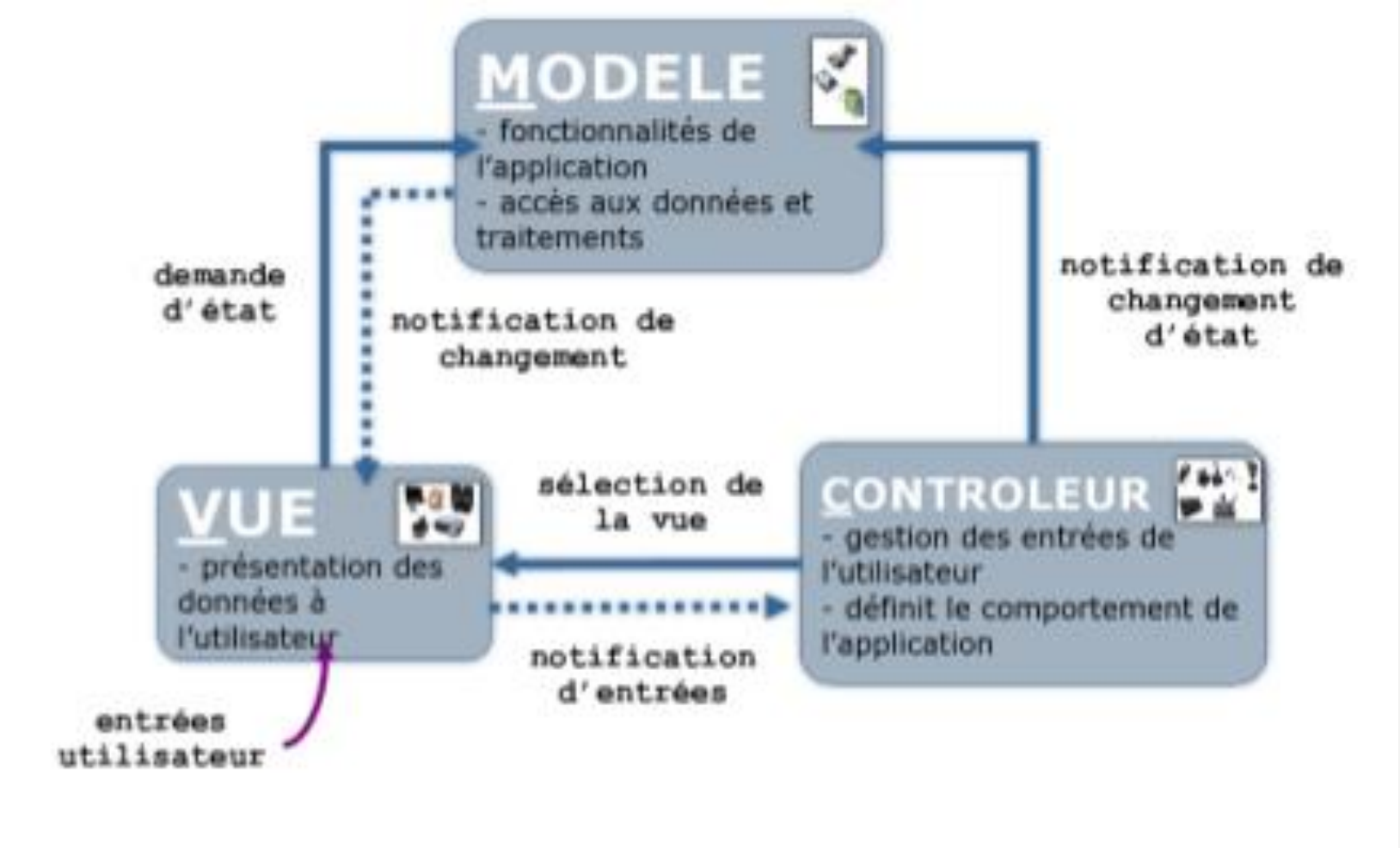
COMMUNICATION ENTRE LES COMPOSANTS MVC



MVC: EN RÉSUMÉ



MVC: VISION PLUS CONCRÈTE



AVANTAGES ET INCONVÉNIENTS

Avantages

- Structure ‘propre’ de l’application
- Indépendance ‘données’ ‘représentation’ ‘comportements’
- Modulaire et réutilisable

Inconvénients

- Mise en place complexe dans le cas d’applications importantes
- Mises à jour potentiellement trop nombreuses ‘Spaghettis’ dans le code
- Contrôleur et Vue restent souvent fortement liés au Modèle
- complexité de communication entre les composants

MERCI POUR VOTRE ATTENTION