

# Génie des Systèmes Interactifs

## Architecture et Design Patterns

Philippe Palanque, Célia Martinie

ICS-IRIT

{palanque, martinie}@irit.fr

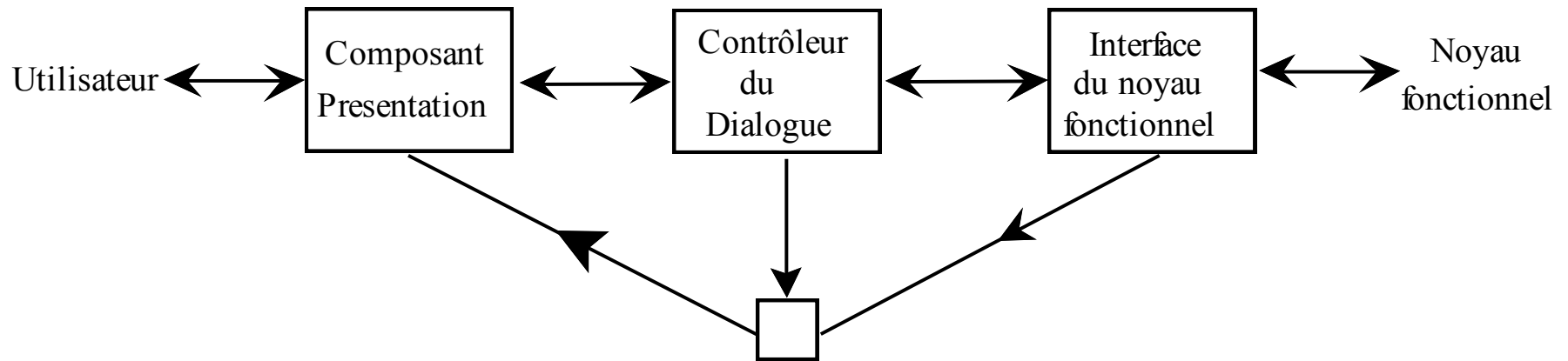


# Plan

- Architectures de référence (concepts)
  - Seeheim
  - Arch/Slinky
- Design patterns
  - Principes
  - Exemples

# **ARCHITECTURE LOGICIELLE POUR LES SYSTÈMES INTERACTIFS**

# Modèle de Seeheim (1983)



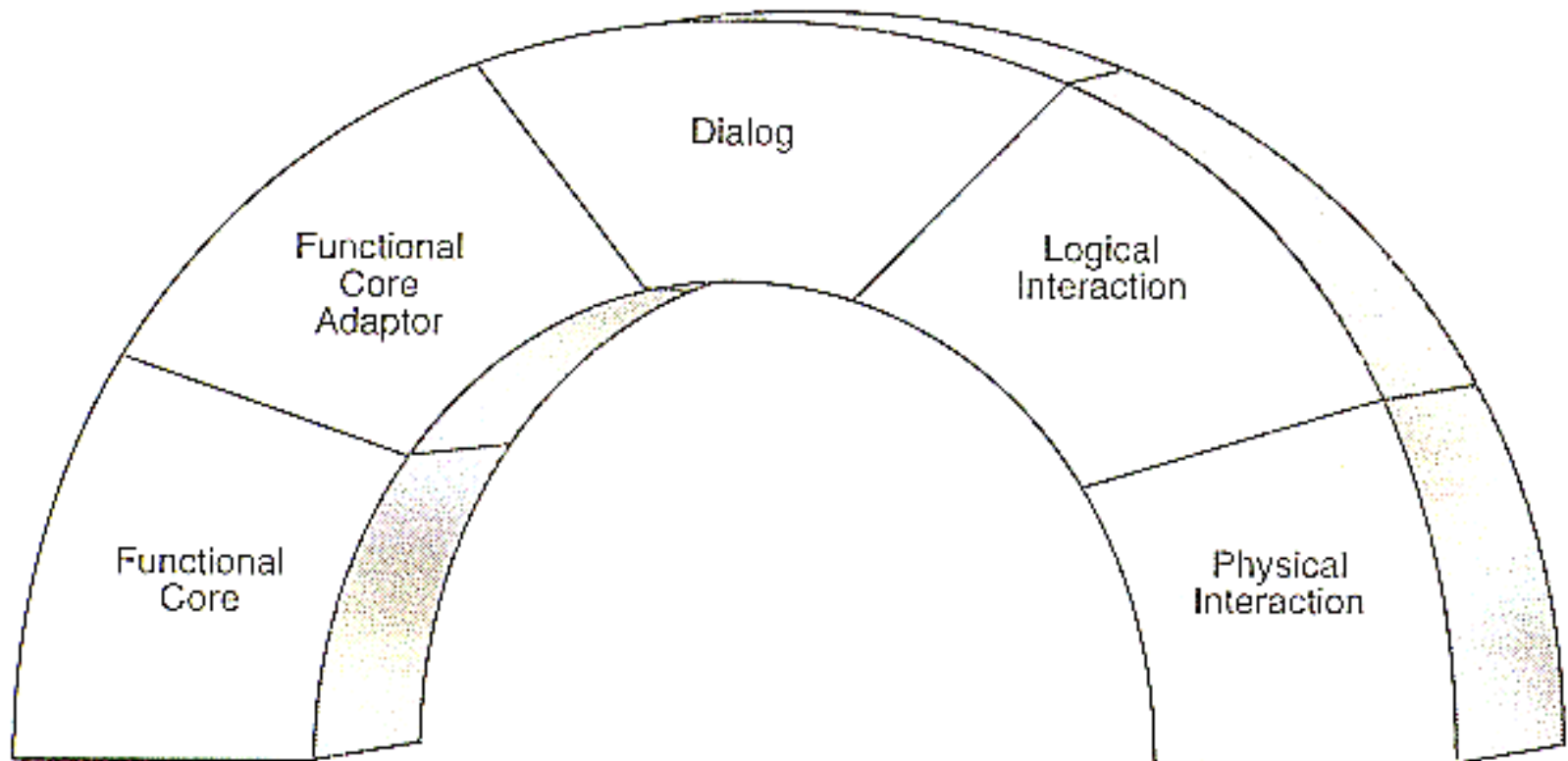
# Modèle de Seeheim

- Présentation: E/S, affichage, interaction
- Dialogue: séquençement
- Interface avec le noyau fonctionnel: adaptateur au domaine

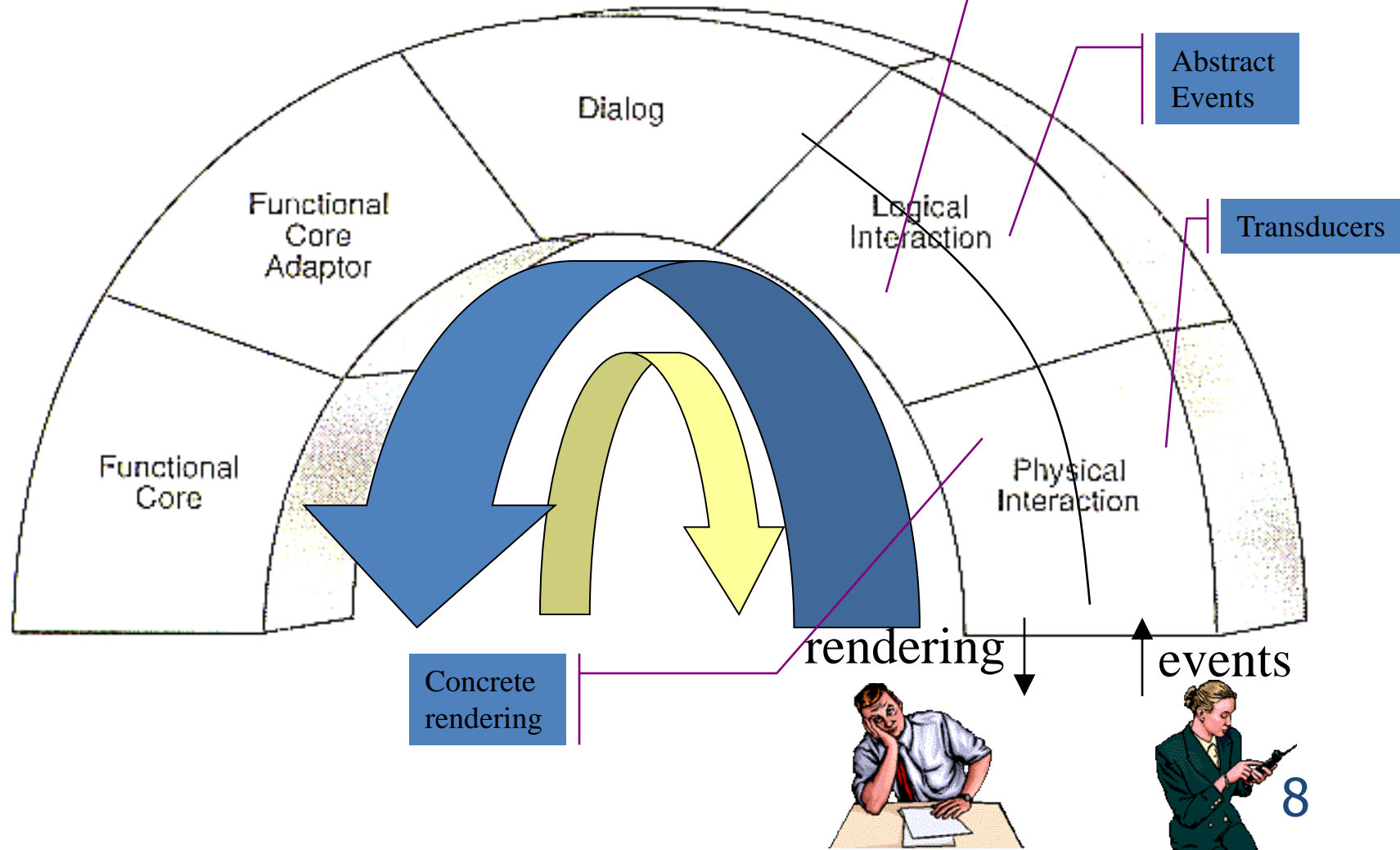
# Noyau fonctionnel (besoins)

- Notification  
Possibilité pour un module externe d'être prévenu lorsque l'état du noyau change
- Prévention des erreurs  
Possibilité de savoir si un appel de fonction est licite selon le contexte
- Annulation  
Possibilité de revenir à des états précédents du noyau

# Arch/Slinky (1991)



# Modèle Architectural



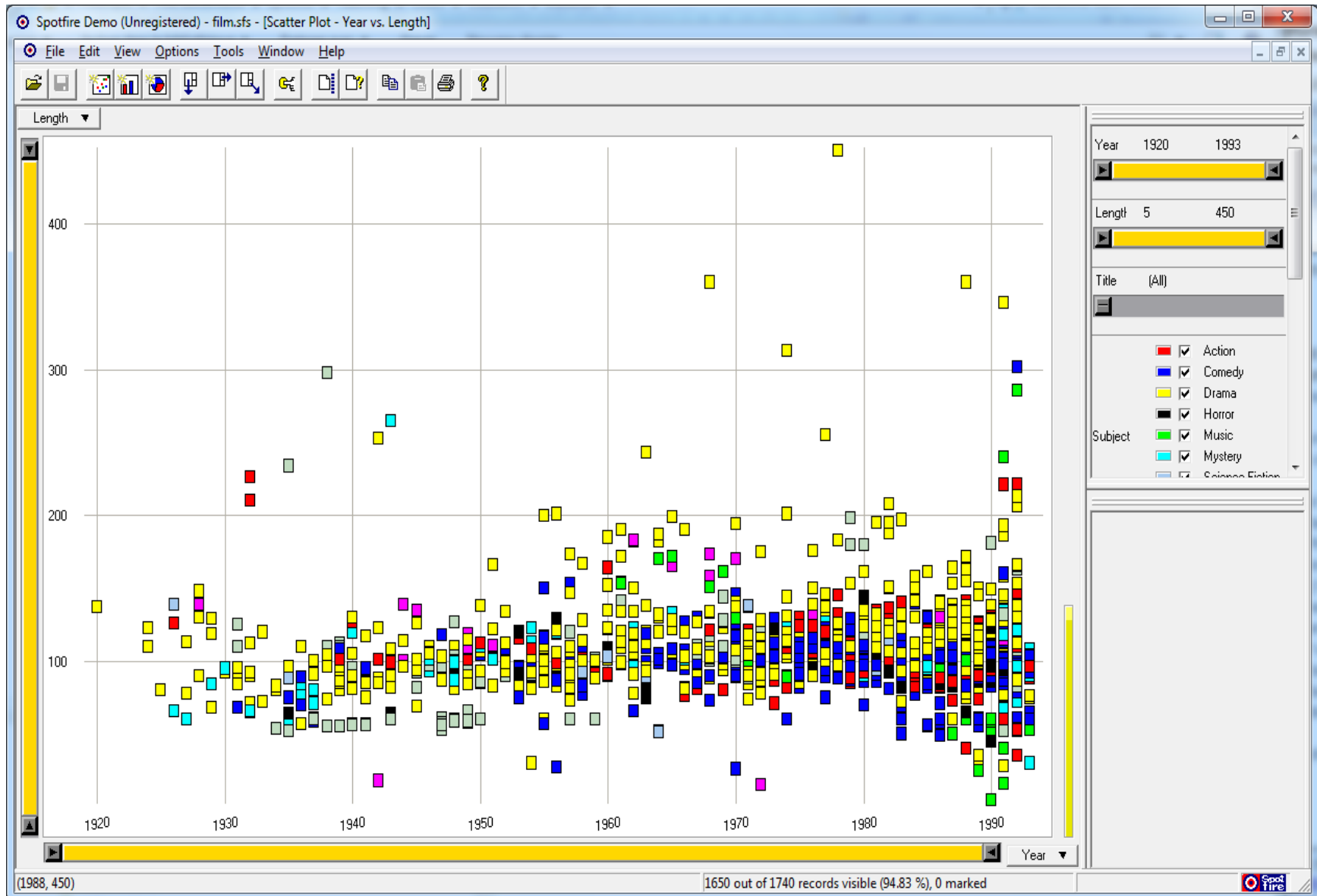


# Exemple: Les quatres boutons

Trois implémentations

1. Sans Arch
2. Présentation et Dialogue au sein d'une même JFrame (dialogue apparent)
3. Trois objets
  - Interface au noyau fonctionnel
  - Dialogue
  - Interaction logique (interaction physique gérée par la boîte à outils SWING)

# Example 2: SpotFire



# **DESIGN PATTERNS POUR LES SYSTÈMES INTERACTIFS**

# Introduction

- Un concept pour augmenter la réutilisabilité du logiciel orienté-objet
- Donner des archétypes de solution à des problèmes récurrents dans la conception de logiciel
- Solutions issues de l'expérience dans des projets réels, validées par l'usage
  - « On n'invente pas un Pattern, on le découvre »

# Inspiration Architecturale

- « Un pattern décrit un problème qui se présente fréquemment dans notre environnement, et décrit l'essence d'une solution à ce problème, de telle sorte que l'on puisse réutiliser cette solution un million de fois sans jamais la refaire deux fois de la même manière »
  - Christopher Alexander

# Bibliographie

- Design Patterns, Elements of Reusable Object-Oriented Software
  - E. Gamma, R. Helm, R. Johnson, J. Vlissides, Addison-Wesley, le «Gang of Four», 1995
- A System of Patterns : Pattern-oriented software architecture
  - Buschmann, F., R. Meunier, H. Rohnert, P. Sommerlad, and M. A Stal, Wiley 1996
- Smalltalk-80: the language and its implementations
  - Goldberg, A. and Robson, D.. Addison Wesley; 1983.
- Java

# Éléments d'un Pattern

- Nom du pattern
  - augmenter le vocabulaire du design, augmenter le niveau d'abstraction
- Description du problème
  - contexte d'application
- Solution
  - en termes d'un modèle générique (la plupart du temps OO)
- Conséquences
  - avantages, inconvénients, compromis

# Types de patterns

- **de comportement (Behavioral)**

Solutions pour l'interaction et la répartition des responsabilités entre les classes ou objets

- **de création (Creational):**

Solutions pour l'instanciation des objets

- **de structure (Structural)**

Solutions pour la composition des classes ou objets



# Les patterns ne sont pas

- Des bibliothèques de classes réutilisables
  - listes chaînées, arbres, ...
- Des applications toutes faites
  - traitent uniquement un sous-problème bien défini
- Des classes génériques (templates)

# Design Pattern 1: MVC

Goldberg, A. and Robson, D.. Addison  
Wesley; 1983 Smalltalk

# Le Design Pattern MVC par l'exemple

The screenshot shows a Microsoft PowerPoint presentation titled "DesignPatternMVC.ppt" in compatibility mode. The main slide, titled "Design Pattern MVC par exemple", is partially obscured by a "Contrôle du volume" (Volume Control) window. The volume control window has four tabs: "Contrôle du volume", "Sons wave", "Synthé. SW", and "Lecteur CD". Each tab contains a "Balance" slider, a "Volume" slider, and a "Muet" (Mute) checkbox. The "Contrôle du volume" tab is currently selected. To the right of the slide, there is a "Rechercher" (Search) sidebar with a search bar and a "Rechercher dans" (Search in) dropdown menu. The sidebar also shows "Collections sélectionnées" (Selected collections) and "Les résultats devraient être" (The results should be). The presentation's taskbar at the bottom shows the current slide is "Diapositive 7 sur 22" (Slide 7 of 22) and the language is "Français (France)".

Design Pattern MVC par exemple

Contrôle du volume

Options ?

Contrôle du volume    Sons wave    Synthé. SW    Lecteur CD

Balance:    Balance:    Balance:    Balance:

Volume:    Volume:    Volume:    Volume:

☐ Tous muets    ☐ Muet    ☐ Muet    ☐ Muet

Avancé

Rechercher :

Rechercher dans :

Collections sélectionnées

Les résultats devraient être :

Tous types de fichiers multimédia

Diapositive 7 sur 22    Français (France)

# Le Design Pattern MVC par l'exemple

The screenshot shows a Windows XP desktop environment. The main application is Microsoft PowerPoint, displaying a presentation titled "Le Design Pattern MVC par l'exemple". The presentation is on slide 9, which features a blue background and the title "Le Design Pattern MVC par l'exemple" in large blue text. A small graphic of overlapping colored squares is to the left of the title. The PowerPoint interface includes a ribbon with tabs like "Accueil", "Insertion", "Création", etc., and a taskbar on the left with icons for various applications. A "Mélangeur de volume" (Volume Mixer) window is open in the foreground, showing volume sliders for "Périphérique", "Haut-parleurs / Casque", "Sons Windows", "Windows Live Messenger", and "Microsoft Sync Center". The taskbar at the bottom shows the date and time as "19:44 mercredi 04/03/2009".

9

## Le Design Pattern MVC par l'exemple

Mélangeur de volume

Périphérique

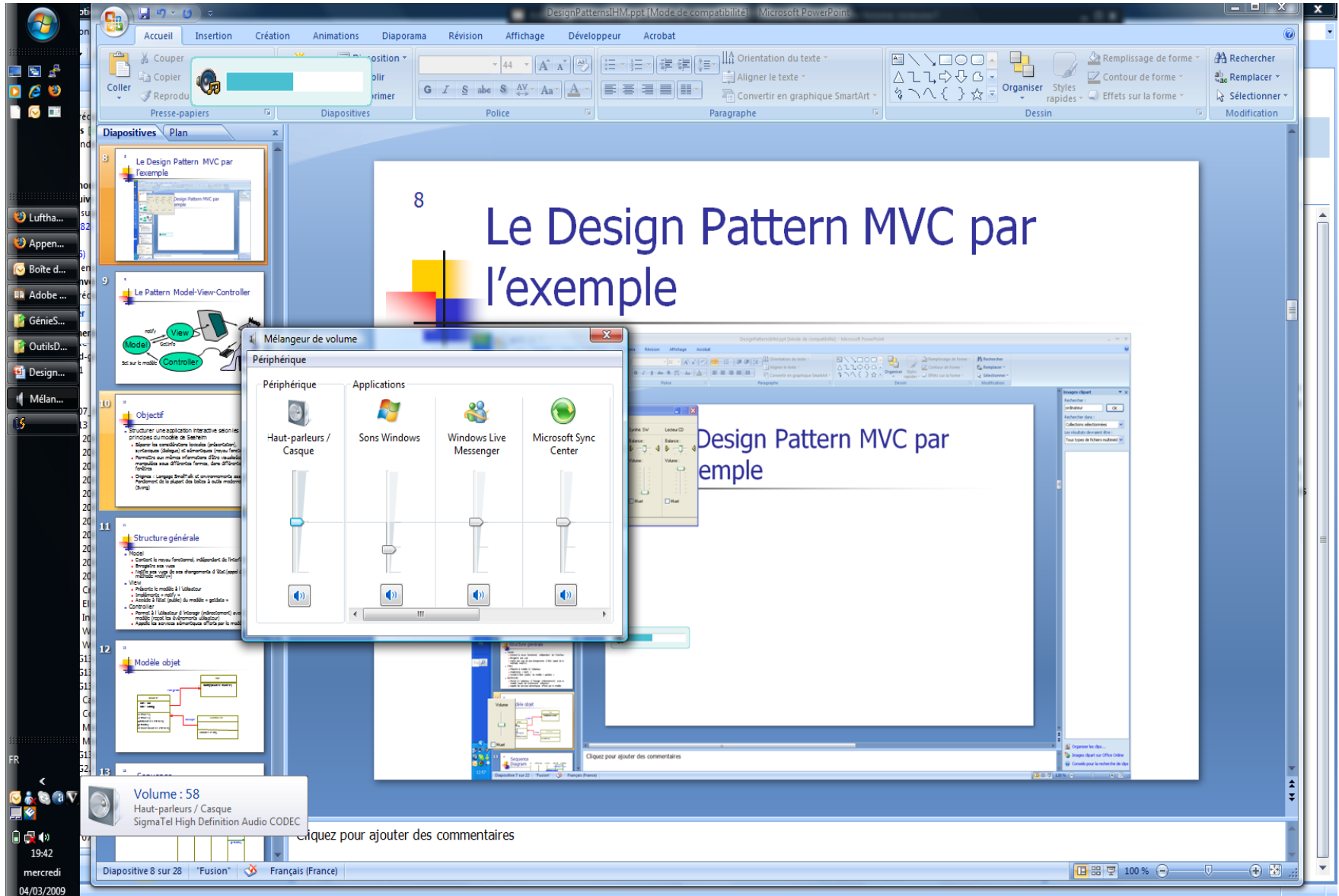
Périphérique Applications

Haut-parleurs / Casque Sons Windows Windows Live Messenger Microsoft Sync Center

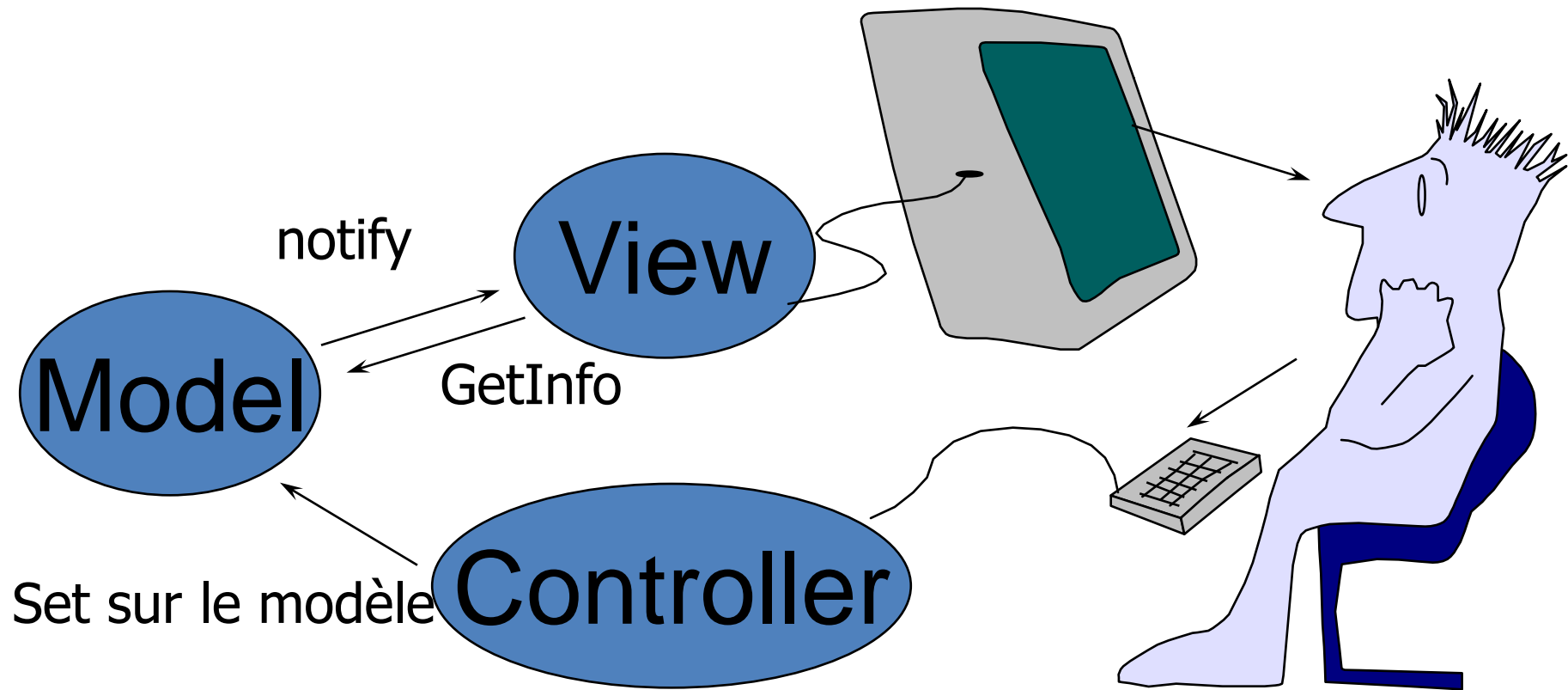
Cliquez pour ajouter des commentaires

Diapositive 9 sur 30 "Fusion" Français (France)

# C'est super mais faut pas rêver



# Le Pattern Model-View-Controller



# MVC

- Modèle = données de l'application (structures et fonctions),
- Vue = informations présentées à l'utilisateur à partir des données du modèle,
- Contrôleur = responsable de l'interaction avec l'utilisateur

## Enchaînement

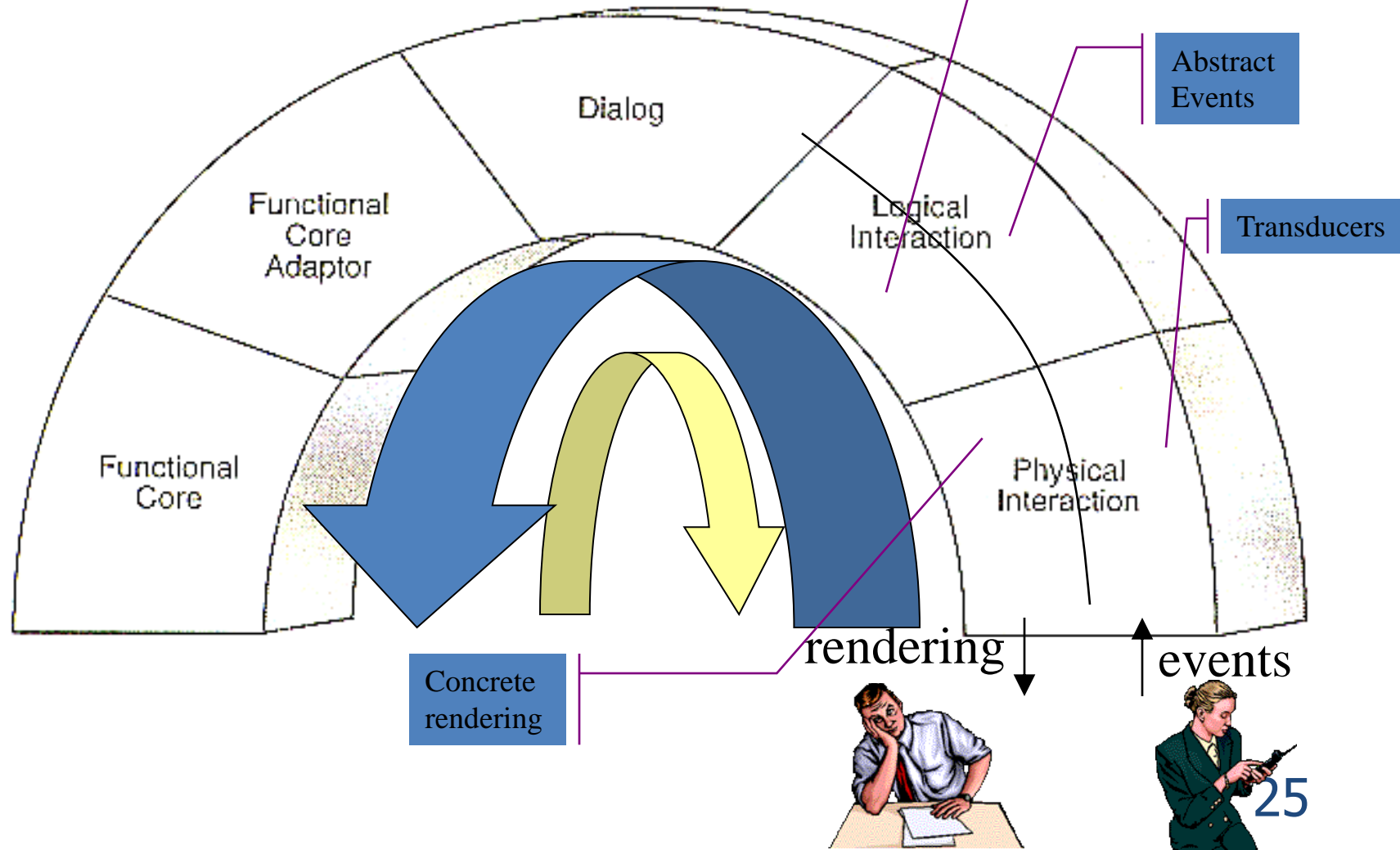
1. Requête envoyée à l'application analysée par le contrôleur
2. Contrôleur demande au modèle approprié d'effectuer les traitements
3. Modèle prévient la(les) vue(s) adapté(es)
4. La(les) vues demandent au modèle les nouvelles valeurs

# Objectif

- Structurer une application interactive selon les principes du modèle ARCH
  - Séparer les considérations lexicales (présentation), syntaxiques (dialogue) et sémantiques (noyau fonctionnel)
  - Permettre aux mêmes informations d'être visualisées et manipulées sous différentes formes, dans différentes fenêtres
  - Différents chemins entre input et output
  - Origines : Langage SmallTalk et environnements associés. Fondement de la plupart des boîtes à outils modernes (Swing)



# Modèle Architectural



# Structure générale

- Model
  - Contient le noyau fonctionnel, indépendant de l'interface
  - Enregistre ses vues
  - Notifie ses vues de ses changements d'état (appel de la méthode «notify»)
- View
  - Présente le modèle à l'utilisateur
  - Implémente «notify»
  - Accède à l'état (public) du modèle « getdata »
- Controller
  - Permet à l'utilisateur d'interagir (**indirectement**) avec le modèle (reçoit les événements utilisateur)
  - Appelle les services sémantiques offerts par le modèle (modifs)