

Outils de développement,  
programmation événementielle et  
IHM

Cyrille Bertelle  
UFRST Le Havre  
25, rue Ph. Lebon  
76058 Le Havre Cedex  
Cyrille.Bertelle@univ-lehavre.fr

22 mars 2004

# Plan général du cours

1. Introduction
  - objectifs
  - la nécessité d'une méthodologie
  - modèles, langage et outils retenus
2. Le modèle objet avec UML
  - Les raisons d'une méthodologie objet
  - Les différentes descriptions par diagrammes d'UML
  - présentation rapide de BlueJ
3. Java, langage de développement objet
  - Caractéristiques de Java
  - Classes et objets
  - Types et structures de contrôle
  - Exceptions
  - Héritage
  - Entrée/Sorties
  - Les threads
4. Java, graphisme de base
  - Applets
  - Awt
5. Les concepts et les modèles généraux pour la construction d'IHM
6. Swing : une implantation du modèle MVC en Java

# Chapitre 1

## Introduction

### 1.1 Objectif du cours

**Objectif :**

- Conception et développement d'applications et d'interfaces homme-machine (IHM).

**Pourquoi et comment développer des IHM ?**

- Développement de l'informatique et de son utilisation
  - Augmentation du nombre d'utilisateurs aussi bien en utilisation domestique qu'en utilisation professionnelle
  - Avant : utilisation réservée à des spécialistes ... une interface minimale
  - Aujourd'hui : large gamme d'utilisateurs ... nécessite des interfaces ergonomiques sinon les applications sont délaissées
  - Aujourd'hui : multiples échanges de données (par exemple, via Internet) nécessitant des méthodes de manipulation. Un moteur de recherche est une interface.
- Développement du hardware
  - Permet l'utilisation abondante de graphismes et d'interactivités qui deviennent incontournables aux systèmes d'interfaçage.
- Des modèles et des méthodes sont nécessaires.
  - Le graphisme et l'interactivité à outrance ne fait pas tout !
  - On vise à augmenter la facilité et l'efficacité de l'utilisation et NON la distraction (sauf si c'est le but visé) !
  - Besoin de développer des méthodologies
  - Modèle cognitif des utilisateurs
  - Interface "intelligente"
  - Interface multi-modale

**Démarche de conception d'une interface :**

- Inspirée des techniques utilisées en génie logiciel
- Implémentation dans le cadre de la programmation objet : UML et Java

## 1.2 La nécessité d'une méthodologie

- Pb : Comment gérer le développement et la maintenance/évolution d'applications informatiques industrielles ?
  - Mêmes outils de gestion pour le concepteur/développeur et pour le client
  - Mêmes outils pour un spécialiste en informatique et pour un non spécialiste
  - Fournir des spécifications des applications à restituer à des équipes de développement
  - Outils de dialogue avec les clients
  - Faciliter la validation
  - Favoriser la réutilisabilité

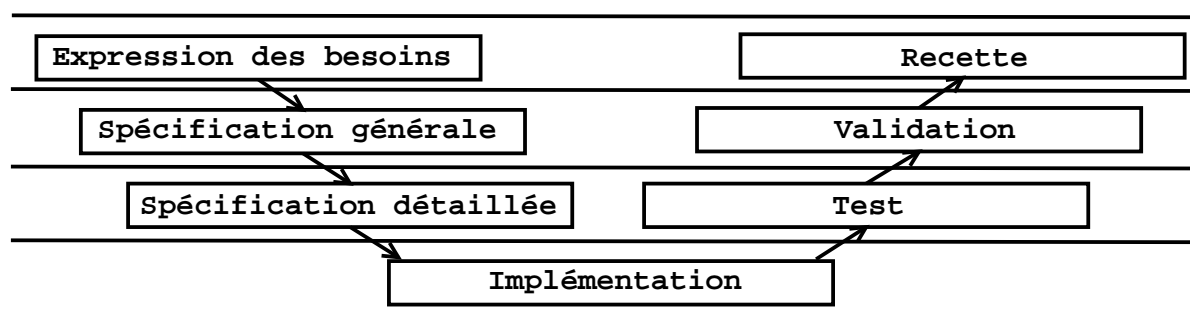


FIG. 1.1: Cycle en V

- Il y a un monde entre l'activité d'un développeur passionné qui travaille seul et le développement industriel d'une application de grande taille qui peut rassembler des centaines de programmeurs sur plusieurs années.
  - Travail isolé : chacun possède ses habitudes et en déduit des méthodes personnelles.
  - Travail industriel
    - Applications de plus en plus complexes
    - Communication nécessaire et efficace entre les participants
    - Nécessité d'un cahier des charges : savoir se mettre d'accord avec les utilisateurs dans un langage clair
    - Concevoir des applications capables d'évoluer
- Modèle de développement en spirale : on doit pouvoir disposer de prototypes puis de versions qui s'améliorent progressivement en respectant un cahier des charges. Il faut donc savoir faire évoluer le développement de manière efficace.

## 1.3 Les modèles, langages et outils retenus

### 1.3.1 UML : Unified Modeling Language

- S'est construit suite à l'identification des différentes phases de conception d'un logiciel et des contraintes associées
- S'est construit suite au développement de recherches méthodologiques depuis 10 ans autour de la conception orientée objets (OO)
- Début 90 :

- Plus de 50 méthodes OO
- 3 méthodes majeures fusionnent sous l'impulsion de leurs concepteurs : James Rumbaugh, Gary Booch et Ivar Jacobson.
- Il en résulte un langage unificateur de description graphique. Ce n'est pas une méthode.
- Novembre 97 : standard UML adopté par l'OMG (Object Management Group)

### **1.3.2 Java**

- Langage de programmation objet multi-plateforme, intégrant des bibliothèques graphiques et de la programmation événementielle.
- Inclus des systèmes d'interfaçage avec le Web.
- Extension aux développements d'applications distribuées ... réseaux d'entreprises.

### **1.3.3 Swing**

- Bibliothèque graphique Java basée sur MVC (Modèle Vue Contrôleur) qui définit des concepts pour le développement d'IHM.

### **1.3.4 BlueJ**

- Environnement de développement dédié à Java de l'Université de Southern Denmark ... C'est une belle interface adaptée à l'enseignement de Java (Gestion interactive des objets, visualisation des diagrammes UML, ...).