

Approches Objet de la Programmation

~ Introduction ~

Didier Verna
EPITA / LRDE

didier@lrde.epita.fr



Plan

Contexte

Programmation Impérative
Programmation Procédurale
Évolution

Paradigmes de Programmation

Notion de Paradigme
Limitations de l'Impératif / Procédural

L'Approche Orientée-Objet

Génèse
Historique

Plan

Contexte

Programmation Impérative
Programmation Procédurale
Évolution

Paradigmes de Programmation

L'Approche Orientée-Objet

Programmation Impérative

Programme

```
instruction 1  
instruction 2  
...  
instruction n
```

- ▶ **Instruction** : ordre destiné à produire un effet de bord
- ▶ **Effet de bord** : modification du contexte environnant
- ▶ **Ordre d'exécution important !**

Contexte

Paradigmes

P00

Programmation Procédurale

- Extension logique de la programmation impérative
- Factorisation d'une séquence d'instructions souvent répétées en *procédure* que l'on peut appeler autant de fois que l'on veut

Programme
...
fais ceci
fais cela
...
fais ceci
fais cela
...

Programme
fais tout:
 fais ceci
 fais cela
...
fais tout
...
fais tout
...

AOP / Introduction – Didier Verna
5/18

Contexte

Paradigmes

P00

Origine

- Paradigmes proches du matériel sous-jacent
modèle de Von Neumann
- Langage machine
 - manipulation de registres
 - échanges avec la mémoire

Mémoire

Unité de contrôle

Unité arithmétique et logique

Accumulateur

Entrée

Sortie



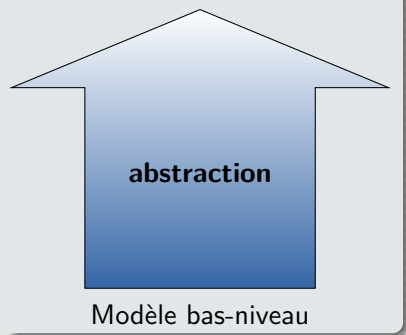
AOP / Introduction – Didier Verna
6/18

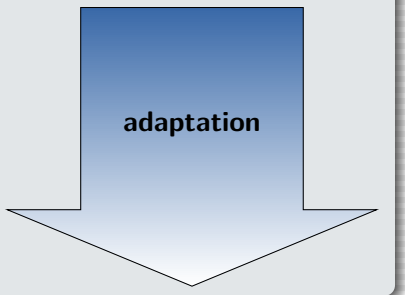
Contexte

Paradigmes

P00

Deux Familles de Langages

Bottom-Up

abstraction
Modèle bas-niveau

Top-Down
Modèle haut-niveau

adaptation

- La programmation impérative / procédurale est bottom-up

AOP / Introduction – Didier Verna
7/18

Contexte

Paradigmes

P00

Plan

Contexte

Paradigmes de Programmation
Notion de Paradigme
Limitations de l'Impératif / Procédural

Approches Alternatives

AOP / Introduction – Didier Verna
8/18

Contexte
Paradigmes
POO

Paradigmes de Programmation

- ▶ **Notion de paradigme :**
 - ▶ Affecte l'expressivité
 - ▶ Affecte la manière de penser
- ▶ **Des tas :** impératif, procédural, orienté-objet, fonctionnel, logique, méta, etc.
- ▶ **Turing-Complétude :** nos langages de programmation ont tous la même puissance d'expression
- ▶ **La question véritable :**
 - « Peut-on exprimer X ? »
 - « Peut-on exprimer X *facilement* ? »



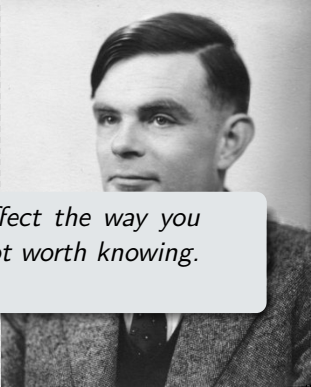

9/18

AOP / Introduction – Didier Verna

Contexte
Paradigmes
POO

Paradigmes de Programmation

- ▶ **Notion de paradigme :**
 - ▶ Affecte l'expressivité
 - ▶ Affecte la manière de penser
- ▶ **Des tas :** impératif, procédural, orienté-objet, fonctionnel, logique, méta
 - A language that doesn't affect the way you think about programming, is not worth knowing.*
- ▶ **Turir** de pr — Alan Perlis
 - puissance d'expression
- ▶ **La question véritable :**
 - « Peut-on exprimer X ? »
 - « Peut-on exprimer X *facilement* ? »




9/18

AOP / Introduction – Didier Verna

Contexte
Paradigmes
POO

Spécificités d'Autres Paradigmes

Fonctionnel

```
map (+4) [1, 2, 3, 4, 5] -- [5, 6, 7, 8, 9]
```

Logique

```
pere(didier,coline).
homme(X) :- pere(X,_).
? - homme(didier).      % yes
```

Méta

```
(defvar program (list '+ 1 2))
(eval program)           ;; 3
```


10/18

AOP / Introduction – Didier Verna

Contexte
Paradigmes
POO

Deux Limitations en Procédural

Convergence d'état

```

graph TD
    Etudiant --> Personne
    Travailleur --> Personne
    Retraité --> Personne
    subgraph Personne
        direction TB
        nom_prenom[nom / prénom]
    end
    
```

Divergence de comportement

```

graph TD
    FaireTourner(faire tourner) --> Cercle
    FaireTourner --> Triangle
    FaireTourner --> Rectangle
    
```

▶ La POO nous permet de nous affranchir de ces deux limitations


11/18

AOP / Introduction – Didier Verna

Contexte Paradigmes POO

Plan

Contexte

Paradigmes de Programmation

L'Approche Orientée-Objet

- Génèse
- Historique

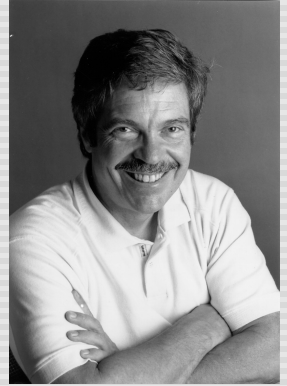
AOP / Introduction – Didier Verna 12/18

Contexte Paradigmes POO

Terminologie

It was probably in 1967 when someone asked me what I was doing, and I said : « It's object-oriented programming. » [...] I thought of objects being like biological cells and/or individual computers on a network, only able to communicate with messages.

— Alan Kay [Kay, 2003]



► **Grosse erreur !**
« Message-Oriented Programming » aurait été plus approprié (dixit Alan Kay lui-même)

AOP / Introduction – Didier Verna 13/18

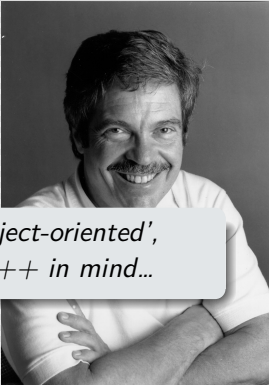
Contexte Paradigmes POO

Terminologie

It was probably in 1967 when someone asked me what I was doing, and I said : « It's object-oriented programming. » [...] I thought of objects being like biological cells and/or individual computers on a network, only able to communicate with messages.

— Alan Kay [Kay, 2003]

Actually I made up the term 'object-oriented', and I can tell you I did not have C++ in mind...



► **Grosse erreur !**
« Message-Oriented Programming » aurait été plus approprié (dixit Alan Kay lui-même)

AOP / Introduction – Didier Verna 13/18

Contexte Paradigmes POO

Les Objets du Monde Réel

- Un *état* (statique)
- Un *comportement* (dynamique)

Personne	Avion
nom / prénom / âge	marque / couleur
mange / travaille / se déplace	prend des passagers / se déplace

► Toute personne a un nom, mais pas le même. Tout salarié a un employeur, mais toute personne n'est pas forcément salariée. Une personne et un avion peuvent se déplacer, mais différemment. etc.

► La POO nous aide à exprimer ce type de relations

AOP / Introduction – Didier Verna 14/18

Contexte Paradigmes POO

Historique

Smalltalk (1971, publique en 1980)

- ▶ Alan Kay et Dan Ingalls
- ▶ « Tout objet » et envoi de message
- ▶ Premier langage dit « orienté-objet »
- ▶ Contexte : DynaBook [Kay, 1972]



Simula (1962, mais surtout 1967)

- ▶ Ole-Johan Dahl et Kristen Nygaard
- ▶ Contient déjà tous les concepts
- ▶ Considérablement inspiré Smalltalk
- ▶ La paternité des concepts est attribuée à Dahl / Nygaard
- ▶ Leur but était pourtant différent de celui de Kay



AOP / Introduction – Didier Verna 15/18

Contexte Paradigmes POO

Historique

Smalltalk (1971, publique en 1980)

- ▶ Alan Kay et Dan Ingalls
- ▶ « Tout objet » et envoi de message
- ▶ Premier langage dit « orienté-objet »

Most programming systems were named "Zeus", "Odin", or "Thor", and hardly did anything. I figured that "smalltalk" was so innocuous a label that if it ever did anything nice, people would be pleasantly surprised.

Simula

- ▶ Considérablement inspiré Smalltalk
- ▶ La paternité des concepts est attribuée à Dahl / Nygaard
- ▶ Leur but était pourtant différent de celui de Kay



AOP / Introduction – Didier Verna 15/18

Contexte Paradigmes POO

Paysage Orienté-Objet Actuel

- ▶ **Aucune réelle définition**
 - ▶ Nombreux langages, visions très différentes
 - ▶ Ensemble flou de concepts, ni exclusif, ni exhaustif
 - ▶ Concepts mal définis / articulés [Nierstrasz, 2010]
- ▶ **Effort de standardisation**
 - ▶ OMG (Object Management Group)
association à but non lucratif américaine créée en 1989
 - ▶ UML (Unified Modelling Language)
langage standardisé de l'OMG pour la modélisation graphique de systèmes orientés-objet (composants logiciels, interactions, déroulement séquentiel, déploiement matériel etc.)



AOP / Introduction – Didier Verna 16/18

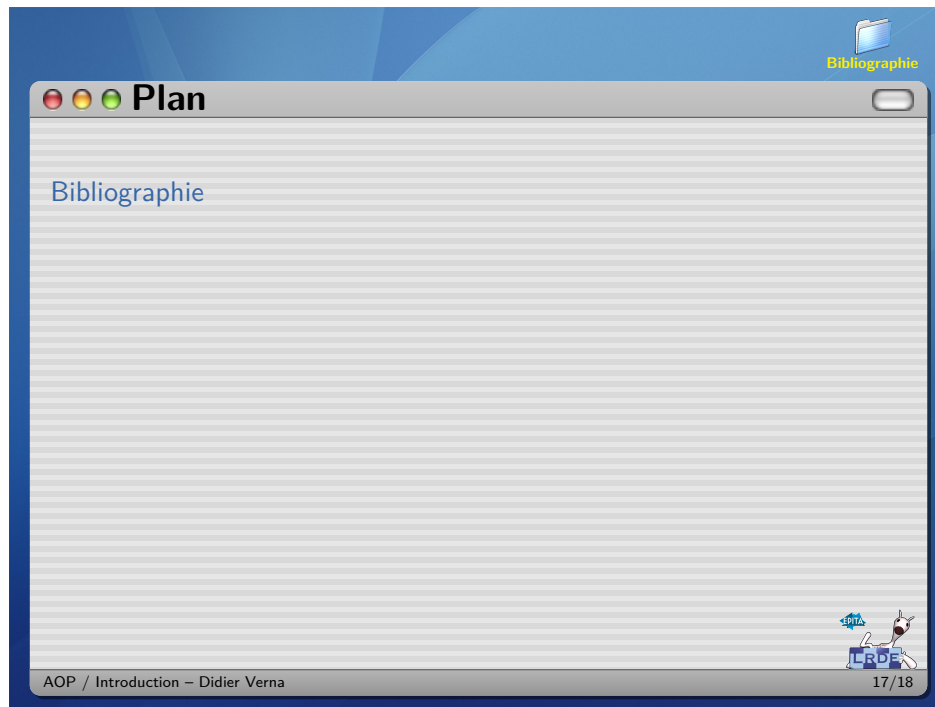
Contexte Paradigmes POO

Paysage Orienté-Objet Actuel

- ▶ **Aucune réelle définition**
 - ▶ Nombreux langages, visions très différentes
 - ▶ Ensemble flou de concepts, ni exclusif, ni exhaustif
 - ▶ Concepts mal définis / articulés [Nierstrasz, 2010]
- ▶ **Effort de standardisation**
 - ▶ *Object-oriented programming is an exceptionally bad idea which could only have originated in California*
— Edsger Dijkstra
 - ▶ systèmes orientés-objet (composants logiciels, interactions, déroulement séquentiel, déploiement matériel etc.)



AOP / Introduction – Didier Verna 16/18

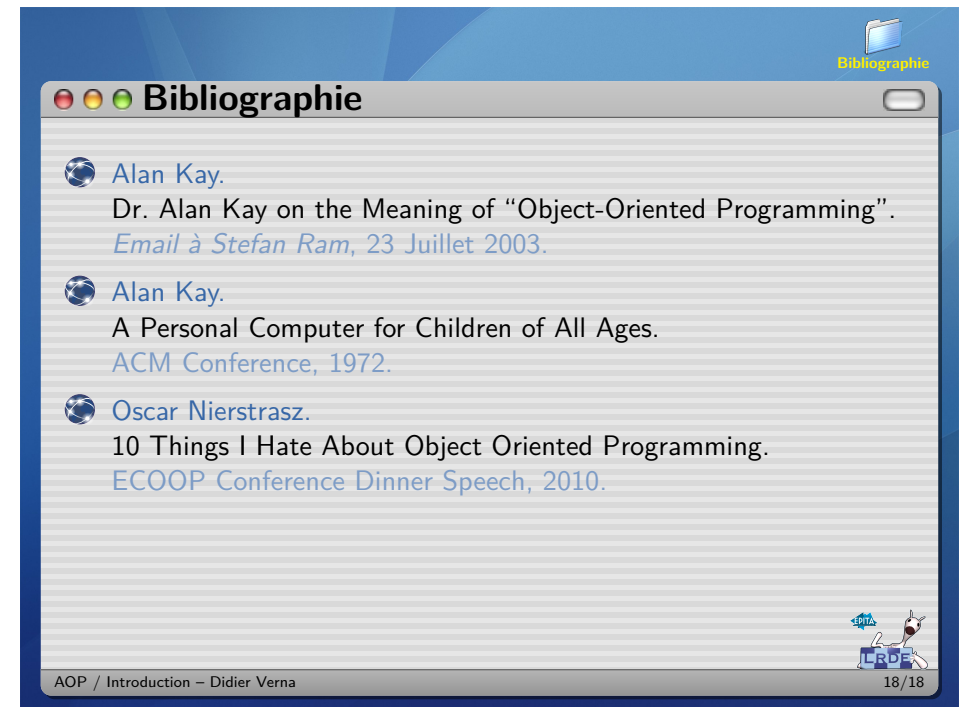


Plan

Bibliographie

AOP / Introduction – Didier Verna

17/18



Bibliographie

-  [Alan Kay.](#)
Dr. Alan Kay on the Meaning of “Object-Oriented Programming”.
Email à Stefan Ram, 23 Juillet 2003.
-  [Alan Kay.](#)
A Personal Computer for Children of All Ages.
ACM Conference, 1972.
-  [Oscar Nierstrasz.](#)
10 Things I Hate About Object Oriented Programming.
ECOOP Conference Dinner Speech, 2010.

AOP / Introduction – Didier Verna

18/18