

Approches Objet de la Programmation

～ Introduction ～

Didier Verna

didier@didierverna.net



didierverna.net



[@didierverna](https://twitter.com/didierverna)



[didier.verna](https://facebook.com/didier.verna)



[in/didierverna](https://in.linkedin.com/company/didierverna)

 Plan

Genèse

Historique



Plan



Genèse

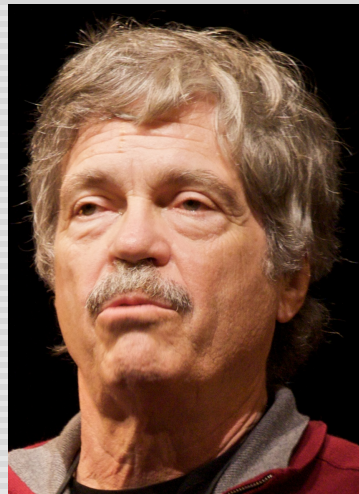
Historique

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)



Source : Wikimedia Commons

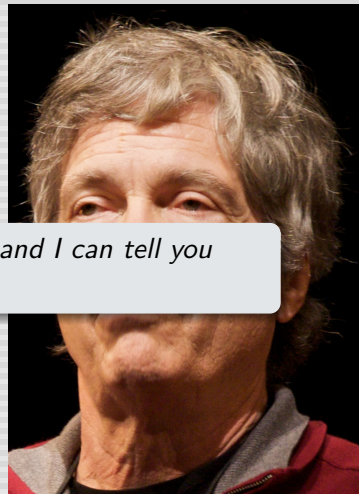
Terminologie

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.

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

► Gro

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



Source : Wikimedia Commons

Les Objets du Monde Réel

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

Personne

nom / prénom / âge

—
mange / travaille / se déplace

Avion

marque / couleur

—
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

 Plan

Genèse

Historique

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
 - ▶ Principale source d'inspiration (avec Lisp) pour Smalltalk
- ▶ Paternité des concepts : Dahl / Nygaard
- ▶ Leur but était pourtant différent de celui de Kay



Source : Wikimedia Commons



Source : Computer Timeline

Historique

▶ Smalltalk (1971 - publique en 1980)

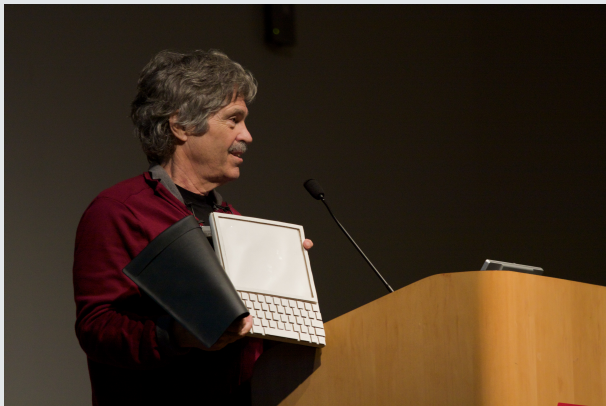
- ▶ Alan Kay
- ▶ « Tout ol
- ▶ Premier l
- ▶ Contexte

▶ Simula (1960)

- ▶ Ole-Joha
- ▶ Contient
- ▶ Principal pour Sm

▶ Paternité des

▶ Leur but étai



Source : Wikimedia Commons



Wikimedia Commons



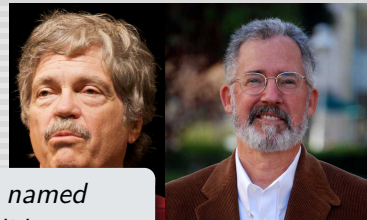
Computer Timeline

Historique

- ▶ **Smalltalk** (1971, publique en 1980)
 - ▶ Alan Kay et Dan Ingalls
 - ▶ « Tout objet » et envoi de message
 - ▶ Premier langage dit « orienté-objet »
 - ▶ Contexte : Smalltalk-80 (Kay, 1979)

- ▶ **Simula** (1961)
 - ▶ Ole-Johan Dahl
 - ▶ Contient Smalltalk
 - ▶ Principal langage pour Sm

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.



Wikimedia Commons



Source : Computer Timeline

- ▶ Paternité des concepts : Dahl / Nygaard
- ▶ Leur but était pourtant différent de celui de Kay

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.)
- Ne traite que de la POO classique (UML 🗑️ Agile 🗑️)

► Bilan des Approches

- AOP1 : classique industrielle (à base de classe, statique)
- AOP2 : plus générale et expressive (à base de classe, dynamique, +++)
- Mais aussi : par prototypes (sans classe, cf. Self, JavaScript)

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]

► Tim Rentsch [Rentsch, 1982]

My guess is that object-oriented programming will be in the 80s what structured programming was in the 70s. Everyone will be in favor of it. Every manufacturer will promote its products as supporting it. Every manager will pay lip service to it. Every programmer will practice it (differently). And no one will know just what it is.

► Bilan des Approches

- AOP1 : classique industrielle (à base de classe, statique)
- AOP2 : plus générale et expressive (à base de classe, dynamique, ++)
- Mais aussi : par prototypes (sans classe, cf. Self, JavaScript)

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

Et avec son éloquence habituelle... 😊

Object-oriented programming is an exceptionally bad idea which could only have originated in California.

— Edsger Dijkstra

- Ne traite que de la POO classique (UML 📄 Agile 📄)

► Bilan des Approches

- AOP1 : classique industrielle (à base de classe, statique)
- AOP2 : plus générale et expressive (à base de classe, dynamique, +++)
- Mais aussi : par prototypes (sans classe, cf. Self, JavaScript)



Bibliographie

Plan



Bibliographie

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.



Tim Rentsch.

Object Oriented Programming.

ACM SIGPLAN Notices, vol. 17, issue 9, 1982.