

Object-Oriented Approaches to Programming

～ Introduction ～

Didier Verna

EPITA / LRE

didier@lrde.epita.fr



lrde/~didier



@didierverna



didier.verna



in/didierverna

Outline

Genesis

History



Plan



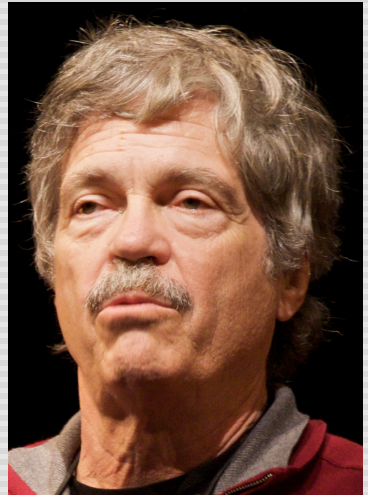
Genesis

History

Terminology

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]

- ▶ **Big mistake!**
"Message-Oriented Programming" would have been more appropriate (*dixit* Alan Kay himself)



Source: Wikimedia Commons

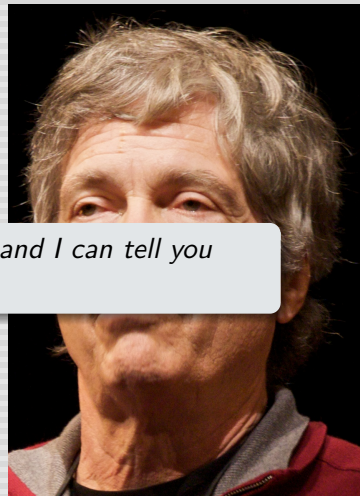
Terminology

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...*

► Big

"Message-Oriented Programming" would have been more appropriate (*dixit* Alan Kay himself)



Source: Wikimedia Commons

Real World Objects

- ▶ *State* (static)
- ▶ *Behavior* (dynamic)

Person

first name / last name / age

—
eats / works / moves

Plane

brand / color

—
boards passengers / moves

- ▶ Every person has a name, but not the same. Every employee has an employer, but a person is not necessarily an employee. Persons and planes move, but not in the same way, *etc.*
- ▶ OOP helps expressing that kind of relations



Plan



Genesis

History



History

- ▶ **Smalltalk** (1971, released in 1980)
 - ▶ Alan Kay et Dan Ingalls
 - ▶ “All Object” and message passing
 - ▶ First “object-oriented” language
 - ▶ Context: DynaBook [Kay, 1972]
- ▶ **Simula** (1962, but mostly 1967)
 - ▶ Ole-Johan Dahl et Kristen Nygaard
 - ▶ Already contains all the concepts
 - ▶ Smalltalk’s greatest source of inspiration (with Lisp)
- ▶ Concepts paternity: Dahl / Nygaard
- ▶ Their goal was different from that of Kay



Source: Wikimedia Commons



Source: Computer Timeline

History

▶ Smalltalk (1971 - released in 1980)

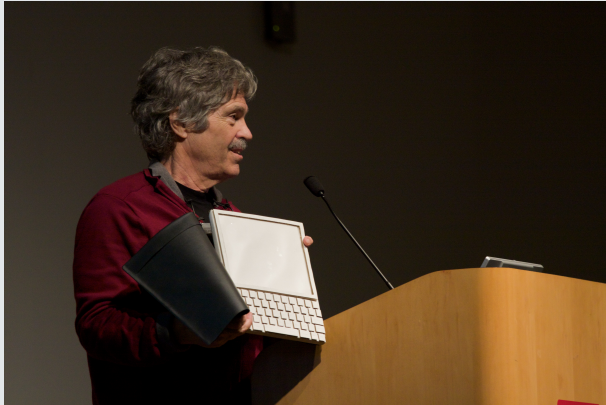
- ▶ Alan Kay
- ▶ "All Objects are Messages"
- ▶ First "object-oriented" language
- ▶ Context: early graphical user interface research

▶ Simula (1961)

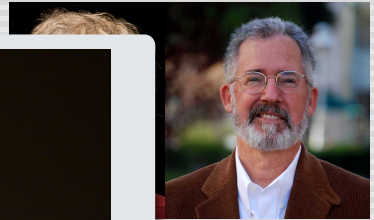
- ▶ Ole-Johan Dahl
- ▶ Already a language for simulation
- ▶ Smalltalk was inspired by it (with Lisp)

▶ Concepts pattern

▶ Their goal was to create a language that could be used to create other languages



Source: Wikimedia Commons



Wikimedia Commons



Computer Timeline

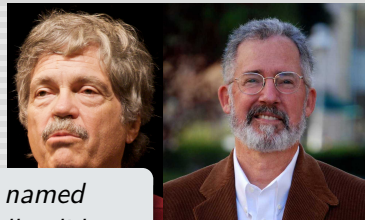
History

- ▶ **Smalltalk** (1971, released in 1980)
 - ▶ Alan Kay et Dan Ingalls
 - ▶ “All Object” and message passing
 - ▶ First “object-oriented” language
 - ▶ Context: Dynabook [Kay, 1970]

- ▶ **Simula** (1961)
 - ▶ Ole-Johan Dahl
 - ▶ Already called “Simula”
 - ▶ Smalltalk (with Lis

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.

- ▶ Concepts paternity: Dahl / Nygaard
- ▶ Their goal was different from that of Kay



Wikimedia Commons



Source: Computer Timeline

Current Object-Oriented Landscape

▶ No proper definition

- ▶ Numerous languages, very different views
- ▶ Fuzzy set of concepts, neither exclusive, nor exhaustive
- ▶ Ill-defined / -articulated concepts [Nierstrasz, 2010]

▶ Standardization effort

- ▶ OMG (Object Management Group)
American non-profit organization created in 1989
- ▶ UML (Unified Modelling Language)
OMG's standardized language for graphical modelling of object-oriented systems (software components, interactions, sequential execution, hardware deployment, etc.)
- ▶ Only deals with classical OOP (UML 🗑️ Agile 🗑️)

▶ Approaches wrap-up

- ▶ AOP1: classical industrial (class-based, static)
- ▶ AOP2: more general and expressive (class-based, dynamic, +++)
- ▶ Also: prototype-based (class-less, cf. Self, JavaScript)

Current Object-Oriented Landscape

► No proper definition

- Numerous languages, very different views
- Fuzzy set of concepts, neither exclusive, nor exhaustive
- Ill-defined / -articulated concepts [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.

► Approaches wrap-up

- AOP1: classical industrial (class-based, static)
- AOP2: more general and expressive (class-based, dynamic, ++)
- Also: prototype-based (class-less, cf. Self, JavaScript)

Current Object-Oriented Landscape

▶ No proper definition

- ▶ Numerous languages, very different views
- ▶ Fuzzy set of concepts, neither exclusive, nor exhaustive
- ▶ Ill-defined / -articulated concepts [Nierstrasz, 2010]

▶ Standardization effort

And with his usual eloquence... 🤔

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

— Edsger Dijkstra

- ▶ Only deals with classical OOP (UML 📄 Agile 📄)

▶ Approaches wrap-up

- ▶ AOP1: classical industrial (class-based, static)
- ▶ AOP2: more general and expressive (class-based, dynamic, +++)
- ▶ Also: prototype-based (class-less, cf. Self, JavaScript)

Plan

Bibliography

Bibliography



Alan Kay.

Dr. Alan Kay on the Meaning of “Object-Oriented Programming”.

Email to 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.