

# Histoire et actualité de l'informatique

Évolution, problèmes, perspectives

Sacha Krakowiak

Université de Grenoble

Séance 3  
Avancées, espoirs, ... déceptions

Années 1970-80

## Où en est-on vers la fin des années 60 ?

### ❖ Les avancées

Les langages de programmation de haut niveau

Les systèmes d'exploitation

Les nouvelles générations d'ordinateurs

Le développement d'une industrie informatique

Constructeurs de machines

Sociétés de service

### ❖ Les limites

Le défi de la production de logiciel

Programmes incorrects, délais non respectés, budgets dépassés ...

Des domaines d'application encore restreints

Encore peu d'impact sociétal

Une organisation centralisée

## Les années 1970-80 : avancées, espoirs

### ❖ La naissance du génie logiciel

La «crise du logiciel», espoirs et mythes

### ❖ La quête du «langage idéal»

### ❖ Les bases de données

### ❖ L'intelligence artificielle, espoirs et déceptions

### ❖ Les circuits intégrés et les microprocesseurs

### ❖ Les premiers impacts sociétaux

### ❖ La montée des réseaux

Prochaine séance

### ❖ Les ordinateurs personnels

### ❖ Les nouveaux champs d'application

## La crise du logiciel et les débuts du génie logiciel

### ❖ Une prise de conscience (fin des années 60)

Les grands projets informatiques

ne tiennent pas les délais

dépassent leur budget

répondent souvent mal aux attentes

Le constat obligé

écrire des programmes corrects est *difficile*

on en est encore à un stade artisanal

Les conclusions

il faut développer des méthodes et des outils

il faut mieux former les gens

il faut passer à un stade industriel

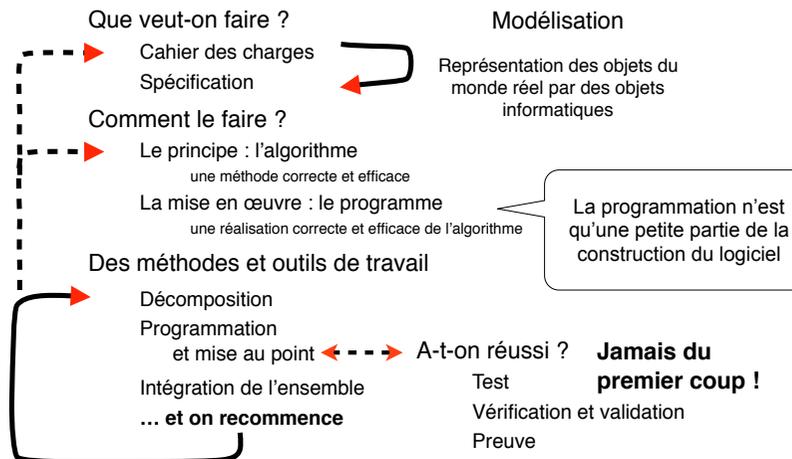
### ❖ La naissance du génie logiciel

Les deux conférences *Software Engineering*

(Garmisch, 1968 ; Rome, 1969)

## Le logiciel : comment ça se fabrique ?

### ❖ Les étapes de la création



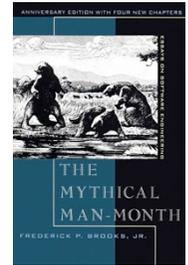
## Les débuts du génie logiciel : mythes et déconvenues

### ❖ Le mythe de la solution «par les masses»

Si le projet est en retard, ajouter de la force de travail

**Faux !** cela ne fait qu'aggraver le problème

Frederick Brooks, *The Mythical Man-Month* (1975)



### ❖ Le mythe de la solution «par les outils»

On résoudra les problèmes en créant des outils plus raffinés

**Très discutable !** la qualité des équipes est le facteur dominant

### ❖ Le mythe de la «programmation automatique»

Les programmes du futur seront créés automatiquement

Le métier de programmeur va disparaître

**Faux !** Le métier va se transformer, mais dans le sens d'une plus grande qualification

## À la recherche du «langage idéal» ...

### ❖ Qu'est-ce qu'un «bon» langage de programmation ?

Un langage sûr

qui empêche de faire les erreurs les plus courantes  
exemple : ne pas ajouter des pommes et des oranges ...

Un langage expressif

Un langage rigoureux

une "sémantique" bien définie (on sait précisément ce qu'on fait)  
idéalement : on peut prouver que le programme fait bien ce qu'on veut qu'il fasse

Un langage élégant et lisible

un programme est fait autant pour être lu (et compris) que pour être exécuté

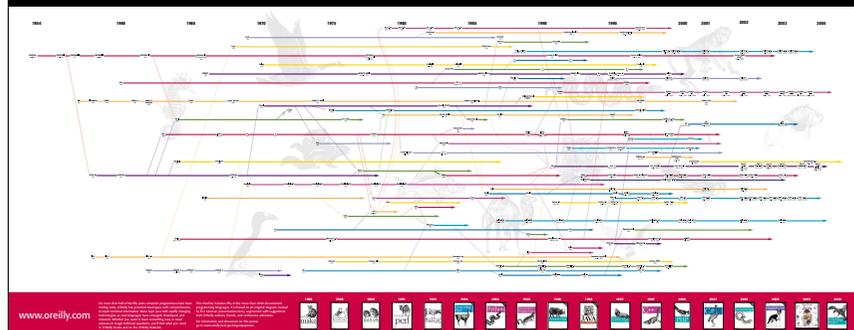
### ❖ «Le» langage idéal n'existe pas ...

... mais ce n'est pas le plus important

## Les langages de programmation

History of Programming Languages

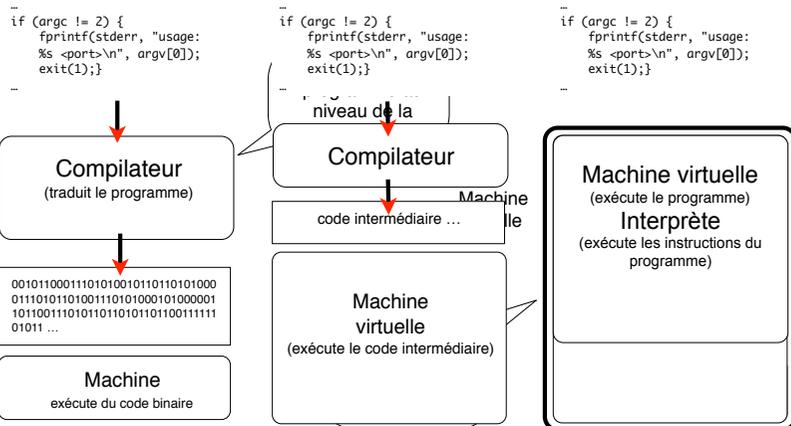
OREILLY



Voir [www.oreilly.com/news/graphics/prog\\_lang\\_poster.pdf](http://www.oreilly.com/news/graphics/prog_lang_poster.pdf)  
Plus de 8 000 langages recensés ...

## Les langages : comment ça marche ?

Problème : combler l'écart entre le programme en langage de haut niveau et le code binaire exécutable de la machine



## Nouveaux langages des années 1970-80 (1)

### ❖ Dans la lignée d'Algol 60

Pascal (Niklaus Wirth, 1970), suivi de Modula-2 (1978)

Initialement destiné à l'enseignement ...

... mais aura une carrière plus large

Ada (Jean Ichbiah, 1979)

Une appel d'offres du ministère de la défense des USA

Une nouvelle tentative vers un langage universel

Utilisé dans les systèmes embarqués

Sûreté  
d'abord

### ❖ Retour vers la machine

BCPL (Martin Richards, 1967-69)

Premier langage à utiliser une machine virtuelle ...

... mais très proche de la machine réelle

C (Dennis Ritchie, 1971)

Issu de BCPL

Efficace, mais peu sûr : mieux vaut être expert

Efficacité  
d'abord

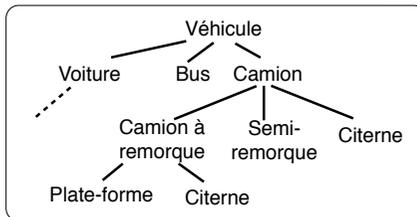
## Nouveaux langages des années 1970-80 (2)

### ❖ Langages à objets

*Modéliser le monde réel*

Simula (Dahl-Nygaard, 1967)

Smalltalk (Alan Kay et al.,  
Xerox PARC, 1978-80)



### ❖ Langages logiques

*Modéliser le raisonnement*

Prolog (Alain Colmerauer -  
Robert Kowalski, 1972)

Pierre est fils de Paul  
Julie est fille de Paul  
Marie est fille de Pierre  
Cécile est fille de Pierre  
Émile est fils de Julie  
A enfant de B = A fils de B **ou** A fille de B  
A cousin germain de B = A enfant de X  
**et** B enfant de Y **et** X enfant de Z  
**et** Y enfant de Z **et** X différent de Y  
Quels sont les cousins germains d'Émile ?  
--> Marie, Cécile

## Les systèmes d'exploitation des années 1970

### ❖ Pas de révolution ...

Les concepts de base sont déjà définis

### ❖ ...mais consolidation

Unix, l'anti-Multics (Ken Thompson et Dennis Ritchie, Bell Labs,  
1971-72)

Solutions simples et efficaces (au départ, sur mini-ordinateur PDP 11)

Un langage, C, devenu standard pour la programmation de systèmes

Simplification pour l'utilisateur

Le début d'une longue carrière

### ❖ Des systèmes pour les nouvelles machines

(détails : séance suivante)

Les ordinateurs personnels : CP/M, MS/DOS et la suite

Les «stations de travail» en réseau : toujours Unix

## Les bases de données : modèles

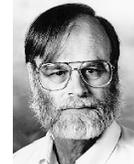
- ❖ Gérer et exploiter de grandes quantités de données
- ❖ Question 1 : assurer la conservation et l'accès rapide  
Stockage et organisation des données sur les disques
- ❖ Question 2 : pouvoir répondre à des questions complexes  
Modèle de données : entités et relations  
Premières tentatives (années 1960) : hiérarchique, réseau  
Une grande avancée : le modèle relationnel (Edgar Codd, 1970)

Nom	Prénom	Lieu	Service	...
Martin	Jeanne	5	Études	
Duval	Jacques	3	Après vente	
Bernard	Pierre	4	Commercial	
Lefèvre	Paule	1	Après vente	
...	...			

Lieu	Ville	Adresse	...
1	Bordeaux	...	
3	Paris	Centre	
4	Paris	Défense	
5	Grenoble	...	

## Les bases de données : transactions

- ❖ Les aléas potentiels du traitement ...  
Incohérence des données  
Panne matérielle ou logicielle  
Interférence entre traitements indépendants simultanés  
Perte de résultats
- ❖ Une solution : les transactions (Jim Gray, 1981)



**Atomicité** : une transaction s'exécute complètement ou pas du tout

**Cohérence** : une transaction qui part d'un état cohérent aboutit à un état cohérent

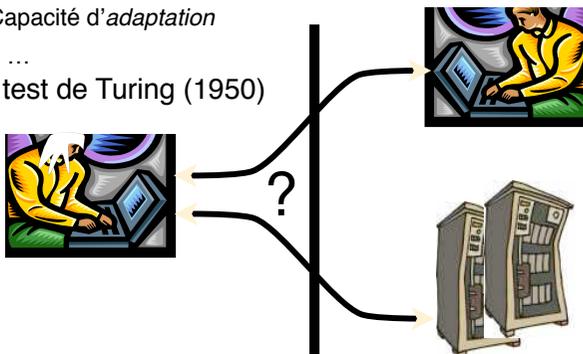
**Isolation** : des transactions indépendantes n'interfèrent pas entre elles

**Durabilité** : une fois validés, les résultats seront préservés

## Les machines peuvent-elles penser ?

- ❖ Qu'est-ce que l'intelligence ?  
Capacité de *raisonnement* (déduire, décider, agir en utilisant des *connaissances*)  
Capacité d'*apprentissage*  
Capacité d'*adaptation*  
...

- ❖ Le test de Turing (1950)



## L'intelligence artificielle : espoirs, déceptions

- ❖ Les débuts de l'IA (1960-72)  
Les jeux : dames, échecs  
Représentation de la connaissance, formalisation du raisonnement  
Eliza, ou comment masquer son ignorance  
Un problème «dur» : la reconnaissance des formes
- ❖ Espoirs déçus (1973-80)  
Une douche froide : le rapport Lighthill (1973)  
De la poudre aux yeux : le programme japonais 5ème génération (1980)  
Controverses philosophiques
- ❖ Renaissance (1980-87)  
Les systèmes experts, ou comment exploiter des connaissances
- ❖ Nouvelles déceptions, nouveau rebond (1987-93)

