



**4 juin 2024**  
**Sébastien George**

**Atelier : Modélisations  
de l'activité de programmation**

**Quels aspects de la programmation  
sont présents/absents ?**



**2024**

**10ÈMES RENCONTRES DES JEUNES  
RECHERCHEURS/EUSES EN EIAH**

**LAVAL**

La consigne pour la synthèse  
« quels aspects (contenus, niveaux scolaires, etc.)  
de la programmation sont présents (et éventuellement absents)  
dans ce qui est présenté ? »



# Enseigner l'informatique ?

**4 niveaux** (synthèse issue de la thèse de M. Branthôme, 2023)

Travaux  
de  
l'atelier

- **la théorie informatique**  
ensemble de concepts, principes et propriétés stables issus de la recherche scientifique ;
- **les techniques informatiques**  
ensemble évolutif de procédés méthodiques fondés sur la théorie informatique et utilisés dans la production d'artefacts matériels et/ou logiciels ;
- **les outils informatiques**  
ensemble d'artefacts utilisés à diverses fins par différents groupes sociaux ;
- **les usages informatiques**
  - utilisation des outils informatiques menant à des incidences sociétales dans de nombreux domaines.

**Science  
Informatique  
(programmation)**



# Quel type de langage de programmation ?

*Langage de haut niveau (pas langage machine, ni assembleur)*

Travaux  
de  
l'atelier

- les langages procéduraux : Fortran, Algol, Cobol, Basic, Pascal, C, ...
  - les langages orientés objet : Simula, Smalltalk, C++, Java, Python, ...
  - les langages fonctionnels : Lisp, Scheme, Haskell, ...
  - les langages logiques : Prolog
- Prog. impérative
- Prog. déclarative

(1)



# Concepts fondamentaux de la programmation impérative ?

(concepts algorithmiques)

## Concepts issus de Dowek *et al.* (2011) :

- C1 : affectation, déclaration et portée des variables
- C2 : type scalaires : entier, flottants, booléen, caractère
- C3 : séquence, test et boucles for et while
- C4 : les entrées/sorties
- C5 : la notion de fonction d'argument et de retour
- C6 : la récursivité
- C7 : types composites :  
chaînes de caractères, enregistrements, tableaux, type dynamique (listes, arbres)

Dowek, G., Archambault, J.-P., Baccelli, E., Boldo, S., Bouhineau, D., Cegielski, P., Clausen, T. H., Guessarian, I., Lopes, S., & Mounier, L. (2011). Une introduction à la science informatique pour les enseignants de la discipline en lycée (CRDP Académie de Paris).



# Positionnement des travaux de l'atelier

	<i>C1 : affectation, déclaration et portée des variables</i>								
	<i>C2 : type scalaires : entier, flottants, booléen, caractère</i>								
	<i>C3 : séquence, test et boucles for et while</i>								
	<i>C4 : les entrées/sorties</i>								
	<i>C5 : la notion de fonction d'argument et de retour</i>								
	<i>C6 : la récursivité</i>								
	<i>C7 : types composites : enregistrements, tableaux, ...</i>								
Lille - Cristal			<b>X</b>						
Lyon - LIRIS	x	x	x		x	<b>X</b>	<b>X</b>		
Grenoble - LIG METAH	x	x	x	x	x		x		
Paris - LIP6 MOCAH	<b>X</b>	x	x						
Genève - TECFA	x	x	x	x	x				
La Réunion - LIM	x	x	<b>X</b>		<b>X</b>		<b>X</b>		
Lausanne - HEP	orienté enseignant								

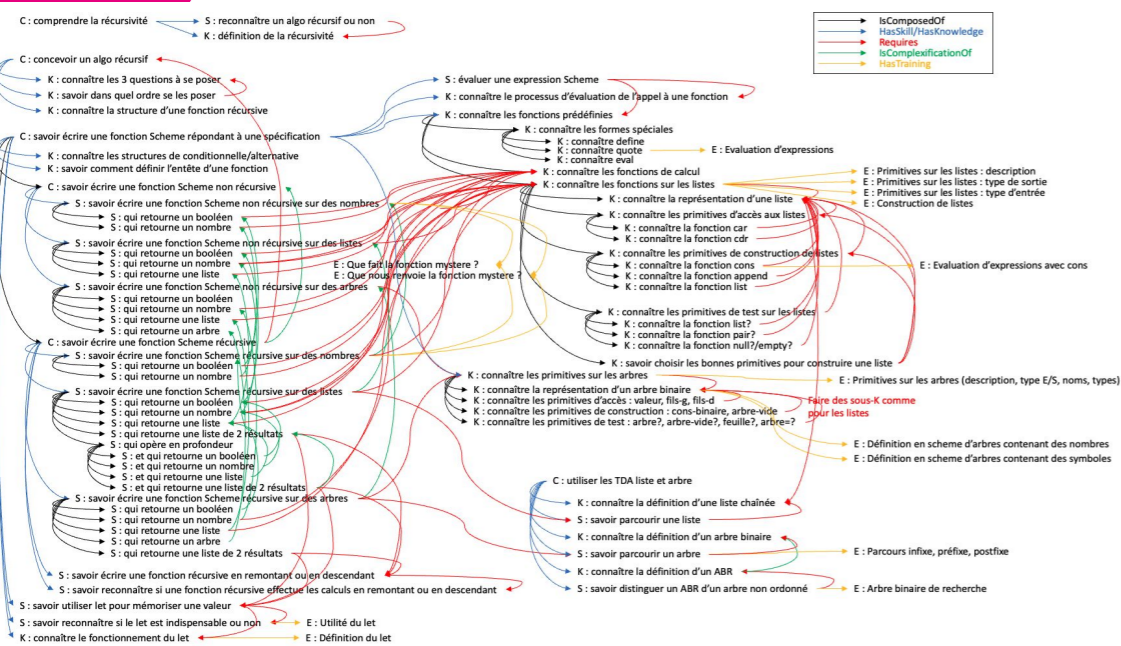


# Différents référentiels

	Référentiel
Lille - Cristal	fondé sur des schèmes
Lyon - LIRIS	meta-modèle COMPER (compétences)
Grenoble - LIG METAH	tableau à double entrée
Paris - LIP6 MOCAH	référentiel PIAF (centré compétences)
Genève - TECFA	modèle praxéologique/type de tâches
La Réunion - LIM	hierarchie ad hoc
Lausanne - HEP	combinaison de modèles



# Différents référentiels pour les concepts à enseigner (1/2)



- 01\_Expressions:
  - 01\_Expressions/Anticiper : competence.md, 00.md, 01.md...
  - 01\_Expressions/Decomposer : competence.md, 00.md, 01.md...
  - 01\_Expressions/Evaluer : competence.md, 00.md, 01.md...
  - 01\_Expressions/Generaliser : competence.md, 00.md, 01.md...
  - 01\_Expressions/Modeliser : competence.md, 00.md, 01.md...
- 02\_Sequences:
  - 02\_Sequences/Anticiper : competence.md, 00.md, 01.md...
  - 02\_Sequences/Decomposer : competence.md, 00.md, 01.md...
  - 02\_Sequences/Evaluer : competence.md, 00.md, 01.md...
  - 02\_Sequences/Modeliser : competence.md, 00.md, 01.md...

LIRIS (ontologie)

LIM (hiérarchie)

Compétences

MOCAH (papier)

METAH (Tableau à double entrée)  
compétences

**Compétence 1 : Définir des abstractions / généraliser**

**C 1.1 Nommer des objets et (séquences d') actions**

**Définition**  
Être capable de donner des noms à des objets, des actions et des séquences d'actions.

**Contexte**  
Il est important de prendre l'habitude de donner des noms aux objets / actions existants afin de pouvoir y référer et faciliter l'expression des solutions.

**À noter**  
Certaines valeurs (objets) peuvent être connues et ne pas changer (par exemple un nom de famille, un numéro de rue), dans ce cas celles-ci sont appelées constantes. Certaines valeurs ne sont pas connues et / ou peuvent changer en fonction du contexte (par exemple la température), et sont donc appelées variables. Ce concept de variable est abordé dans la compétence C 1.6 ci-dessous.

**Exemples illustrant la compétence**

- Étant donné une histoire avec différents personnages présentant des similitudes (par exemple représentant tous des sportifs), mais distinguables, être capable donner des noms afin de les identifier de manière univoque, et de les intégrer dans une histoire.
- Un apprenant réalise une action, un autre la voit et doit simplement la raconter à un troisième, qui ne voit pas le premier apprenant. Ce troisième apprenant doit alors être capable de reproduire cette action.
- Deux apprenants s'accordent sur quatre mots correspondant à quatre (séquences d') actions simples. Un troisième apprenant arrive. Le premier dit un des quatre mots et le deuxième exécute les commandes correspondantes devant le troisième apprenant, qui doit être capable d'apprendre le langage défini par les quatre mots (et par la suite d'exécuter les actions demandées).
- Après avoir défini les étapes nécessaires pour s'habiller, être capable d'appeler cette séquence d'actions "s'habiller".

domaines


Choix ▾

choix 1

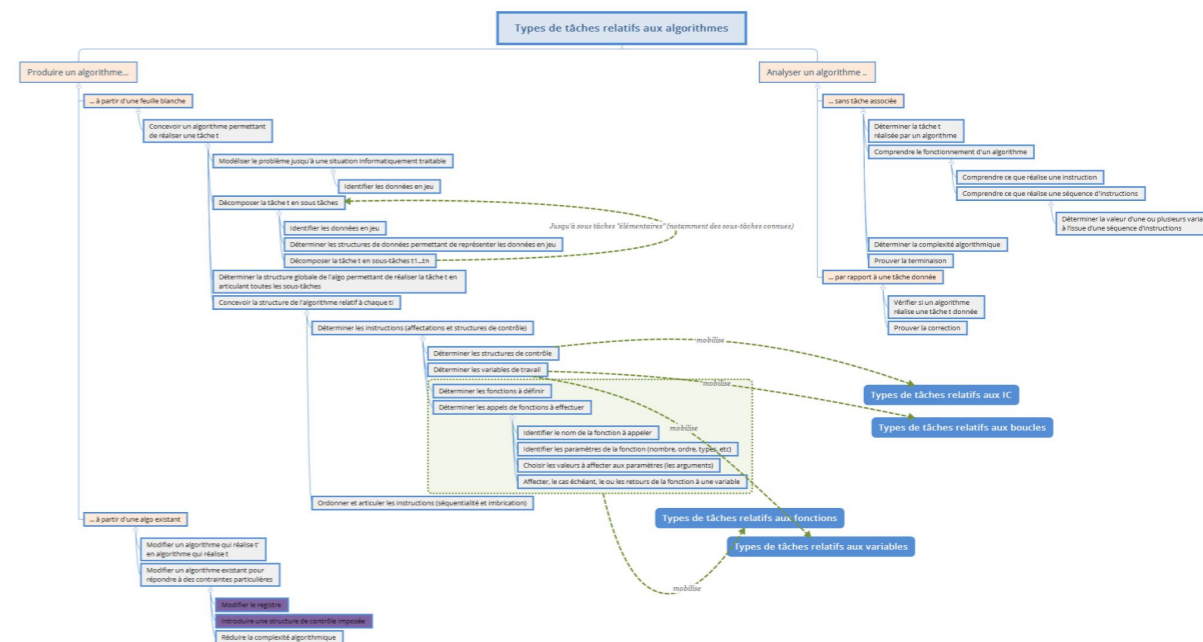
choix 2

choix 3



# Différents référentiels pour les concepts à enseigner (2/2)

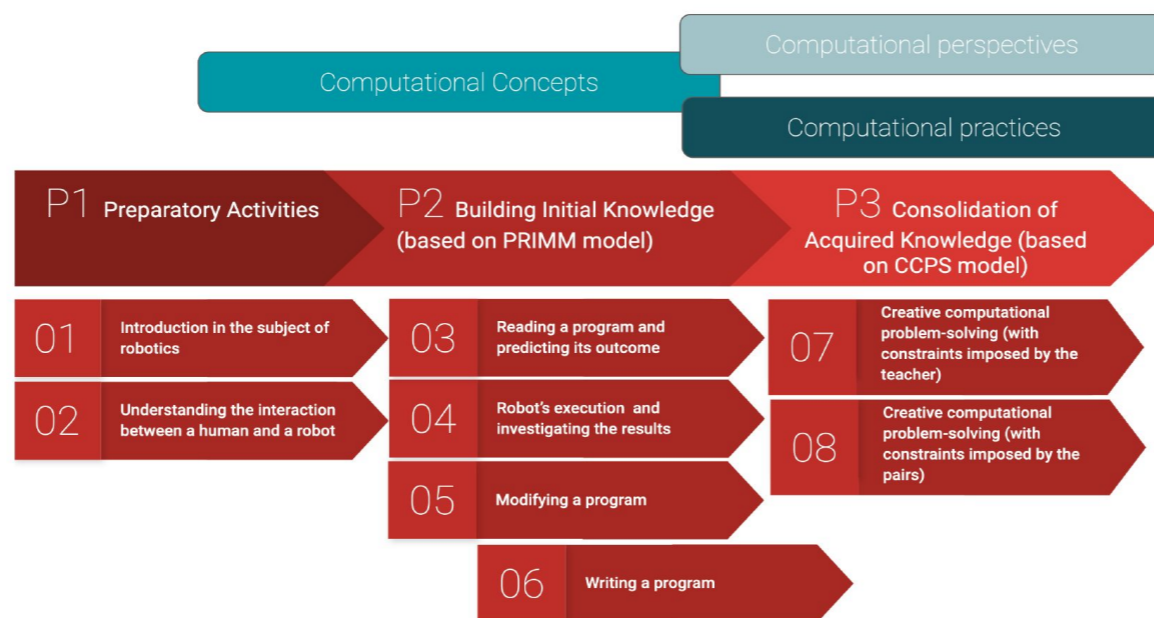
- Description de schèmes
  - But
  - Règles de conduite de l'action
    - i. prise d'information
    - ii. actions
    - iii. contrôle
  - Invariant opératoires
    - i. concepts et propositions tenues pour vraies dans l'action
  - Inférences (adaptation à la situation concrète)



## Schémas d'actions (*Cristal*)

## Type de tâches – arbres (*TECFA*)

## Combinaison de modèles (*HEP*)



# Niveaux scolaires

	Niveaux
Lille - Cristal	Niveaux primaire / collège
Lyon - LIRIS	Licence 1 Math-Info à l'Université
Grenoble - LIG METAH	Cycle 4 (5e, 4e et 3e)
Paris - LIP6 MOCAH	Primaire/collège
Genève - TECFA	France-Suisse 12-18 ans
La Réunion - LIM	Lycée et Licence
Lausanne - HEP	Formation d'enseignants / élèves secondaire

*Quel que soit le niveau, enseignement à des novices (sauf licence info)*



# Quels aspects de la programmation sont absents ?

- Programmation bas niveau (langage machine, assembleur)
- Programmation déclarative (langage fonctionnel ou logique)
- Programmation impérative orientée objet
- Programmation Web
  
- Technique de production de programmes (procédés, IDE, framework, ...)
- Modélisation (objet/UML, BD, IHM, 3D, ...)
  
- Globalement, les activités sont davantage orientées exercices/puzzle que développement d'une application
  
- ...

