

Simon Modeste – Université de Montpellier



# Didactique de l'informatique

DIU Enseigner l'Informatique au lycée

Strasbourg

Mercredi 24 avril 2019

[simon.modeste@umontpellier.fr](mailto:simon.modeste@umontpellier.fr)

# Programme de la journée

- Partie 1. Introduction : didactique de l'informatique
- Partie 2. Difficultés et enjeux sur des points spécifiques
- Partie 3. Introduction à la Théorie des Situations Didactiques

Simon Modeste – Université de Montpellier



# **Partie 1. Introduction.**

## **Didactique de l'informatique**

DIU Enseigner l'Informatique au lycée

Strasbourg

Mercredi 24 avril 2019

# Plan

Objectif : Présenter la nature du questionnement didactique, l'approche choisie, et l'ancrage didactique de l'informatique

- Didactique et épistémologie des sciences
- Épistémologie de l'informatique
- Didactique de l'informatique

# 1. Didactique et épistémologie des sciences et de l'informatique

- La didactique des disciplines scientifiques s'est construite sur une volonté de prendre en compte les savoirs dans la réflexion sur leur enseignement et leur apprentissage, ce qui la distingue d'une approche que l'on pourrait qualifier de pédagogie générale qui viserait à dégager des modalités d'enseignement et d'apprentissage indépendantes des savoirs :

Si l'on devait risquer une définition, on pourrait dire que la didactique d'une discipline est la science qui étudie, pour un domaine particulier (ici les sciences et les mathématiques), les phénomènes d'enseignement, les conditions de la transmission de la "culture" propre à une institution [...] et les conditions de l'acquisition des connaissances par un apprenant. Le point d'entrée dans cette problématique, c'est la réflexion sur les savoirs. (Johsua & Dupin)

- En ce sens, l'épistémologie joue un rôle central dans la constitution même des didactiques des disciplines scientifiques, en permettant de rendre explicites les hypothèses quand à la nature et la structure des savoirs, qui agissent implicitement sur les choix d'enseignement de ces savoirs. En fournissant les éléments relatifs à la genèse, la nature, le rôle et la place des savoirs, l'épistémologie contribue de manière significative au travail didactique d'ouverture de choix raisonnés et fondés scientifiquement pour leur enseignement et leur apprentissage.

[Les didactiques des disciplines scientifiques] se sont fondées par un choix radical qui devait les distinguer progressivement des autres approches concernant l'enseignement scientifique : la volonté – et l'affirmation de la possibilité – d'un abord raisonné, systématique, scientifique et spécifique des phénomènes d'enseignement dans ces domaines, visant à délimiter théoriquement et pratiquement les domaines du possible de ceux de l'"inaccessible". (Johsua & Dupin)



L'épistémologie permet également de prendre en compte et de discuter les frontières et les articulations entre les disciplines scolaires, l'introduction de nouveaux savoirs et de mutations d'une discipline dans l'enseignement.

# Didactique et épistémologie des mathématiques

Dans ce cadre, la didactique des mathématiques s'est fortement appuyée sur des travaux d'ordre épistémologique. L'article fondateur de Artigue (1990a) présente cette relation entre didactique et épistémologie des mathématiques au sens d'une "connaissance des processus par lesquels les concepts mathématiques se forment et se développent" (Artigue, 1990a, p. 243). Elle ajoute :

Au-delà de l'analyse conceptuelle, l'épistémologie intervient à ce niveau sur un plan plus général car ce que vise l'enseignement des mathématiques, ce n'est pas simplement la transmission de connaissances mathématiques, c'est plus globalement celle d'une culture. Il s'agit de faire entrer les élèves dans le jeu mathématique. Mais, qu'est ce jeu mathématique ? Quels sont les processus généraux de pensée qui le gouvernent ? C'est l'analyse épistémologique [...] qui est au premier chef concernée par ces questions. (Artigue, 1990).

Dans cette direction, celle de la vigilance épistémologique, de la prise de distance par rapport à l'objet d'étude, l'analyse épistémologique permet [...] de prendre la mesure des disparités existant entre savoir "savant" [...] et savoir "enseigné". En effet, alors que l'école vit sur la fiction consistant à voir dans les objets d'enseignement des copies simplifiées mais fidèles des objets de la science, l'analyse épistémologique, en nous permettant de comprendre ce qui gouverne l'évolution de la connaissance scientifique, nous aide à prendre conscience de la distance qui sépare les économies des deux systèmes. (Artigue, 1990)

Plus largement, les principales théories développées en didactique des mathématiques relèvent de cette approche, intégrant l'épistémologie propre à la discipline mathématique. Elles permettent de prendre en charge les aspects langagiers, logiques, de validation, cognitifs, institutionnels, etc. spécifiques aux mathématiques, à leur enseignement et à leur apprentissage. La spécificité de la preuve en mathématiques, de la relation des mathématiques aux autres disciplines, du statut des objets, de leur construction et de leur existence y sont prises en charge.

Les théories et travaux développés dans le cadre de la didactique des mathématiques permettent, enfin, de s'appuyer sur ces éléments d'épistémologie non seulement pour produire l'analyse de systèmes didactiques (c'est-à-dire de transmission des savoirs) mais aussi, et surtout, pour construire et proposer des approches, des situations d'apprentissage, des organisations didactiques susceptibles de produire (ou au moins de favoriser) les apprentissages mathématiques attendus.

C'est dans ce même esprit que nous envisagerons la didactique de l'informatique :

celui de prendre en compte les spécificités de l'informatique et de ses concepts pour :

- appréhender les phénomènes d'enseignement et d'apprentissage
- Organiser et contrôler les apprentissage des élèves

# Didactique de l'informatique et des technologies du numérique

Depuis les années 1980 et les premiers enseignements de l'informatique, s'est développée une recherche sur la place de l'informatique dans l'enseignement en tant qu'objet et outil (Baron & Bruillard, 2011). Une recherche en didactique de l'informatique s'est constituée (Arsac, 1989 ; Orange, 1990 ; Baron & Bruillard, 2001 ; Caraballo & Cicala, 2006). Elle s'intéresse à la fois à l'outil et à l'objet informatique, aux technologies et à la science informatiques, aux usages des logiciels comme à la programmation, au numérique, etc.

Les récents actes du colloque DIDAPRO (2015, 2017) montrent cette diversité et illustrent bien qu'il est impossible de réduire l'informatique à un seul de ces aspects, ainsi que de les détacher complètement les uns des autres (Tort & Bruillard, 2010 ; Drot-Delange & Bruillard, 2012).

L'enseignement d'informatique proposé en classes de première et terminale au lycée en France vient apporter de nombreux éléments issus de la sciences informatique (jusqu'alors uniquement enseignés en début d'université – au moins ces dernières années).

Ces contenus nécessitent plus qu'une simple « adaptation » au lycée, et surtout, ils nous amènent ici à faire un choix :

Nous concentrer plutôt sur les aspects relatifs à la science informatique (par rapports à ceux liés aux technologies, logiciels et leurs usages – même si les liens doivent être développés et les connexions ne peuvent être ignorées).

En particulier, une entrée didactique, au sens précédemment développé, en articulation avec les questions épistémologiques nous semble utile.

Dans cette direction, il semble que plusieurs outils et théories didactiques issus de la didactique des sciences ou de la didactique des mathématiques peuvent s'avérer pertinents et effectifs pour traiter certaines questions de didactique de l'informatique.

→ Notre point de vue : prendre en compte l'épistémologie de l'informatique



# Prise en compte de l'épistémologie de l'informatique

Les liens étroits existants entre informatique et mathématiques (comme sciences), et plus profondément entre leurs épistémologies, permettent de soutenir l'hypothèse selon laquelle nombre d'approches et de cadres théoriques de la didactique des mathématiques peuvent s'avérer efficaces pour explorer les questions didactiques liées à l'informatique.

## 2. Épistémologie de l'informatique

Partant du constat que l'épistémologie de l'informatique peut nourrir la didactique de l'informatique et est même nécessaire pour éclairer certaines questions, on peut identifier deux grandes questions :

- l'épistémologie de l'informatique (comme champs de production de savoir), pour mieux comprendre ses spécificités

Qu'est que faire de l'informatique ? Quelle est la nature de l'informatique ?

- l'épistémologie des concepts de l'informatique, en particuliers ceux qui sont/seront enseignés au lycée.

Quelle est la raison d'être de tel ou tel concept ? Quel est son rôle ?

# Science et technique

L'informatique est à la fois science et technique, ce qui lui confère une nature particulière :

Elle étudie à la fois des phénomènes scientifiques (« physique » ou « mathématiques ») et des objets techniques conçus par l'homme (machines, conventions/normes) et soumis à ces phénomènes

# Validation formelle et expérimentale

Elle partage des à la fois des modes de validations proches de la preuve mathématique (étude d'algorithmes d'algorithmes, propriétés de programmes...) mais aussi une dimension expérimentale (robotique, logiciel, complexité/efficacité « en pratique »...).

Il est nécessaire de prendre cela en compte dans l'enseignement et il peut sembler raisonnable de faire exister ces deux dimensions dans la classe (et de les faire comprendre).

# Les quatre « piliers » de l'informatique

« Quand on enseigne une discipline, il faut veiller à en respecter les équilibres internes, afin que les contenus enseignés donnent une image fidèle de la discipline elle-même. » (Dowek, 2011)

Dowek identifie quatre concepts qui structurent l'informatique :

- Algorithmes
- Machine
- Langage
- information

« Il semble plus pertinent d'énumérer, non les sous-disciplines, mais les concepts utilisés dans ce domaine de la connaissance, qui ont sans doute une stabilité temporelle plus grande que les diverses sous-disciplines qui se recomposent sans cesse. Les relations entre ces différents concepts sont également sans doute plus révélatrices de la structure profonde de l'informatique, que les relations entre ces diverses sous-disciplines. » (Dowek)

Cela est aussi très adapté à résister aux évolutions de programmes scolaires et de traverser les niveaux et les filières pour maintenir une cohérence de la discipline scolaire elle-même.

→ On retrouve dans les curriculums ISN puis aujourd'hui NSI... Ils permettent de structurer un curriculum équilibré mais aussi de mettre en lumière les concepts clés de l'informatique.

# Algorithme

Un **algorithme** est une « recette » qui permet de résoudre un certain problème de manière systématique (déterministe ou non).

Un problème → plusieurs algorithmes

Un algorithme → plusieurs programmes.

Les algorithmes opèrent sur des objets, souvent représentés sous forme de données symboliques.



# Machine

Une **machine** est un système matériel (souvent avec un protocole d'interaction qui permet d'échanger des données).

« Aujourd'hui, de nombreuses machines exploitent les propriétés physiques des semi-conducteurs, mais rien n'impose que cela soit toujours le cas et la recherche en informatique explore de nombreuses alternatives » (Dowek).

Réseaux comme machine constituée de milliards d'ordinateurs interconnectés. → répartition des calculs.

Ordinateurs : machines paramétrables

→ Machines universelles pouvant exécuter n'importe quel algorithme opérant sur des données symboliques (n'importe quel programme).

# Langage

Écrire des algorithmes à destination de machines paramétrables nécessite un **langage**.

Nécessité d'un langage spécifique pour qu'un algorithme (sous forme de programme) puisse être exécuté par une machine paramétrable.

Les premiers langages de programmation universels (lambda-calcul, MT) sont antérieurs de peu aux premières machines universelles. Ils sont liés à la nécessité d'exprimer sans ambiguïté et de communiquer (liens à la logique).

Les langages informatiques permettent de faire d'un algorithme l'objet d'un algorithme.

Il existe des langages formels autres que des langages de programmation : spécifier, raisonner sur des programmes, exprimer des requêtes... → notion très générale de langage (formel) en informatique.

# Information

Représenter des données de manière symbolique et codage de **l'information**

Lié à la nécessité de représenter l'information sous forme symbolique pour la rendre manipulable par les algorithmes décrits pour les machines (les programmes).

« La taille du plus court programme qui engendre un message est une manière de définir la quantité d'information contenue dans ce message. Cette théorie de la compression aboutit donc à une théorie quantitative de l'information »

Bases de données : codage de l'information pour son traitement (algorithmes) mais aussi son stockage.

# Unité de l'informatique

« Ces quatre concepts, loin de définir quatre branches de l'informatique, sont souvent utilisés ensemble. C'est parce que nous voulons faire exécuter des algorithmes par des machines paramétrables que nous avons eu besoin d'exprimer ces algorithmes dans des langages formels et les données sur lesquelles ces algorithmes calculent comme des informations. » (Dowek)

→ L'informatique prend son sens et trouve sa spécificité dans l'articulation de ces 4 concepts.

# Pensée informatique

« Ce souci du caractère algorithmique de la description des objets, du langage dans lequel ces descriptions sont exprimées, des flux d'information et des instruments sont plus généralement caractéristiques d'une 'pensée informatique'. Cette pensée a bien entendu de nombreux précurseurs dans l'histoire, comme Galilée qui a à la fois introduit l'utilisation d'instruments – la lunette – et proposé d'utiliser un nouveau langage – le langage mathématique – dans les sciences de la nature, mais ce qui semble nouveau, en revanche, est de prêter attention simultanément, dans un même geste, à ces quatre questions. » (Dowek)

# Pensée informatique d'après Knuth

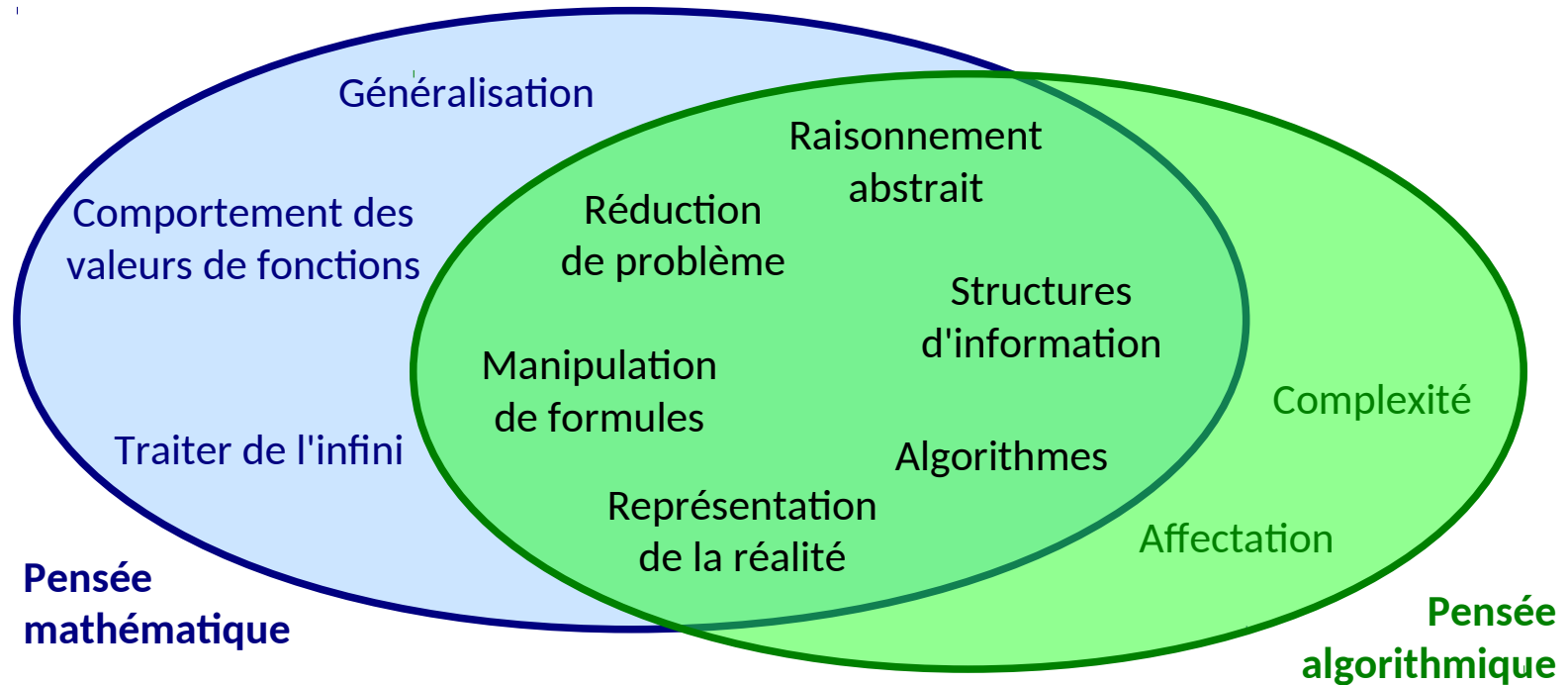
Pour Knuth, la pensée algorithmique est celle de l'informatique.

Il entend « algorithmique » dans un sens large, voire d'informatique



# Pensée algorithmique comme la pensée de l'informatique

Il tente de caractériser ce qui est commun et différent de la pensée mathématique :



# Complexité et affectation

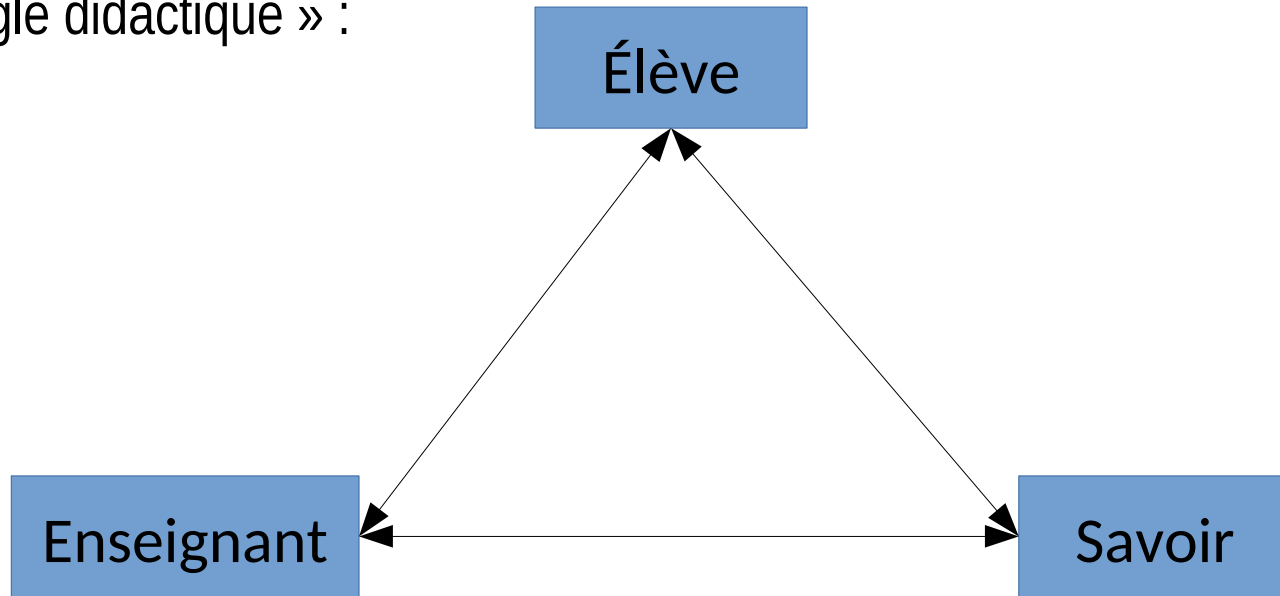
- Complexité comme spécifique de l'activité algorithmique.
- Affectation réfère à la notion de variable informatique.
  - Révélateur d'une orientation vers la résolution effective et systématique des problèmes, au-delà de la constructivité mathématique.
- Knuth insiste aussi sur la nature discrète des problèmes rencontrés à l'interaction des mathématiques et de l'informatique.



### **3. Didactique de l'informatique : questions et enjeux**

# Questions didactiques générales (à instancier pour l'informatique)

- « Triangle didactique » :



Les questions qui intègrent ces trois composantes sont d'ordre didactique.  
Les réponses sont alors spécifiques aux savoirs en jeu.

La nature de la discipline informatique (son épistémologie) amène à se poser des questions spécifiques.

Chaque concept peut lui aussi donner lieu à une réflexion didactique quand à son apprentissage. Il est alors souvent intéressant de comprendre le rôle de ce concept enseigné dans le « savoir savant », l'histoire de sa naissance et de son développement et son origine dans les système d'enseignement .

# Transposition didactique

Du savoir savant au savoir enseigné.

Quand un savoir savant, c'est à dire le savoir des spécialistes du domaine, devient objet d'apprentissage, il subit des transformations.

Ce phénomène est appelé **transposition didactique**

Savoir savant → Savoir à enseigner → Savoir enseigné

# Transposition didactique

Dans cette direction, celle de la vigilance épistémologique, de la prise de distance par rapport à l'objet d'étude, l'analyse épistémologique permet [...] de prendre la mesure des disparités existant entre savoir "savant" [...] et savoir "enseigné". En effet, alors que l'école vit sur la fiction consistant à voir dans les objets d'enseignement des copies simplifiées mais fidèles des objets de la science, l'analyse épistémologique, en nous permettant de comprendre ce qui gouverne l'évolution de la connaissance scientifique, nous aide à prendre conscience de la distance qui sépare les économies des deux systèmes. (Artigue)

- Exemple 1.

Les 4 concepts de l'informatique, par Dowek, relèvent d'un phénomène de transposition didactique :

- Lecture a posteriori de la structure du savoir savant qui vient aujourd'hui structurer les programmes du lycée

- Exemple 2. Machine de référence

Ngyen :

- Absence d'une machine de référence pour enseigner algorithmique et programmation
- Pourtant, dans le savoir savant / enseignement avancé : machine de référence réelle, virtuelle ou idéale ou langage de programmation
- Situations qui partent de la machine calculatrice jusqu'à des machines dotées de caractéristiques supplémentaires.

Chiprianov :

- Proposition d'une machine de référence pertinente pour l'enseignement secondaire (ni machine de Turing, ni ordinateur...)

- Exemple 3.

## Algorithmes et programmes en maths au lycée

- Entrée-sortie vs saisie-affichage

- Contraint par les langages utilisés (Algobox) et par les formalisations attendues (cf programme de maths, documents d'accompagnement, formatage lié au bac)

- Évolution des attentes du bac ces deux dernières années

- Signe d'une superposition des concepts d'algorithme et de programme



# Point de vue sur l'apprentissage

- Erreur en didactique
  - Point de vue de l'erreur comme attestant d'une connaissance de l'élève utilisée hors de son domaine de validité.
  - L'erreur a un sens
  - L'erreur est constitutive de l'apprentissage : elle permet de comprendre les limites d'une connaissance
  - Comment l'élève peut identifier l'erreur ?

# Difficultés et obstacles

- Certains concepts sont plus difficiles que d'autres à apprendre.
- Certains portent une complexité conceptuelle particulière. On les identifie souvent à la régularité des réponses erronées qu'ils provoquent chez les élèves, dans des contextes institutionnel différents (ex : récursivité?).
- On peut alors parler d'obstacles.
- En général, on peut retrouver des traces historiques liées à cette complexité.
- D'autres obstacles relatifs à des savoirs donnés sont produits par le système d'enseignement lui-même (exemple : ordre des notions rencontrées, choix d'un langage de programmation spécifique – entrée-sortie en Algobox, *cloisonnement des disciplines dans la réforme du lycée ?*).  
On parle d'obstacles didactiques.

# Situations d'apprentissages

- Hypothèse (socio)constructiviste:  
L'enseignant ne transmet pas directement des connaissances...  
... mais organise l'apprentissage de l'élève qui construit ses connaissances dans les situations d'apprentissages qui lui sont proposées.  
L'interaction avec les pairs dans ces situations peut contribuer à l'apprentissage.