



Laboratoire d'Innovation et  
Numérique pour l'Éducation

UNIVERSITÉ CÔTE D'AZUR



**INSPÉ** Institut national  
supérieur du professorat  
et de l'éducation  
Académie de Nice



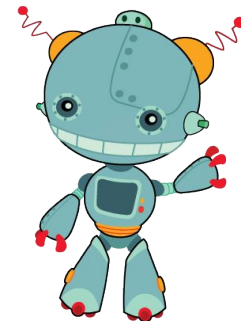
# *Apprendre l'informatique : du code à la pensée informatique*



<https://frama.link/2020Didapro>

@margaridaromero Margarida.Romero@unice.fr

Dir. Laboratoire d'Innovation et Numérique pour l'Éducation  
(LINE)



## **DIDAPRO 8 – DIDASTIC**

L'informatique, objets d'enseignements – enjeux épistémologiques, didactiques et de formation, 5-7 février 2020 @Université de Lille, France

# Un travail en équipe dans le cadre du Laboratoire d'Innovation et Numérique pour l'Education (LINE) et l'INRIA, à l'Université Côte d'Azur



Poste de professeur.e des universités : Didactique des disciplines, créativité et ingénierie coopérative

# Apprendre l'informatique, un défi à relever à plusieurs : Communauté d'Apprentissage de l'Information (CAI)



## Création d'une communauté d'enseignants engagés dans l'apprentissage de l'informatique



## Plan de la présentation

- Apprendre l'informatique ?
- Cadres de compétence numérique
- Quels apprentissages en lien à l'informatique ?
- Pensée informatique et résolution de problèmes
- Analyse d'activités en lien à la pensée informatique
- Communauté d'Apprentissage de l'Informatique (CAI)

# Apprendre l'informatique ?

L'école comme lieu de développement de **citoyen.ne.s** capables de **contribuer et s'épanouir dans la société.**

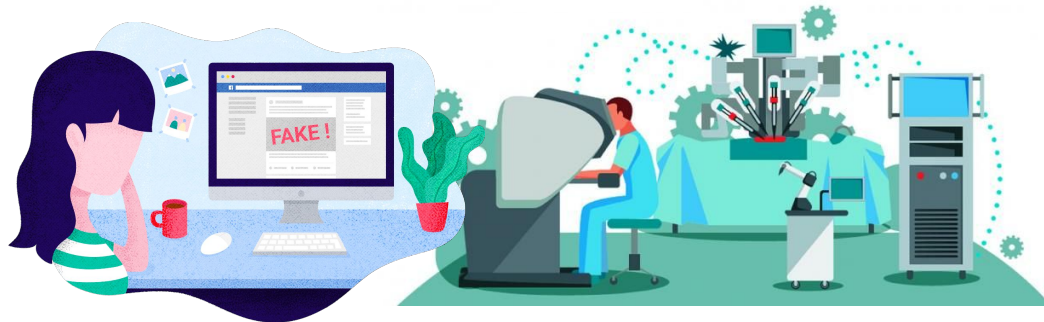
Gilles Dowek : «Préparer les enfants à vivre librement dans un monde d'algorithmes»

**Apprendre l'informatique** à l'école => Emergence d'une **discipline**  
Emergence du questionnement sur la **manière d'apprendre l'informatique**

## Éducation



## Société





# Cadres de compétence numérique

- Education aux médias, **pensée critique** en lien aux usages du numérique
- Connaissances et compétences en lien à des **usages du numérique** (recherche d'information, communication, création de contenu)
- **Connaissances informatiques**
- **Compétences transversales** (pensée informatique, résolution de problèmes, ....), en lien aux usages du numérique

DigComp 2.0



# Référentiel PIX

## 5 DOMAINES - 16 COMPÉTENCES\*



### INFORMATION ET DONNÉES

- › Mener une recherche et une veille d'information
- › Gérer des données
- › Traiter des données



### COMMUNICATION ET COLLABORATION

- › Interagir
- › Partager et publier
- › Collaborer
- › S'insérer dans le monde numérique



### CRÉATION DE CONTENU

- › Développer des documents textuels
- › Développer des documents multimédia
- › Adapter les documents à leur finalité
- › Programmer



### PROTECTION ET SÉCURITÉ

- › Sécuriser l'environnement numérique
- › Protéger les données personnelles et la vie privée
- › Protéger la santé, le bien-être et l'environnement



### ENVIRONNEMENT NUMÉRIQUE

- › Résoudre des problèmes techniques
- › Construire un environnement numérique

\* Déclinaison du référentiel européen DIGCOMP



# Compétence Numérique (Québec)





*L'informatique  
concerne toutes les  
disciplines, elle doit  
être intégrée par les  
disciplines existantes*

*L'informatique doit  
être prise en  
charge par les  
enseignant.e.s en  
mathématiques*

*L'informatique  
doit faire l'objet  
d'une discipline  
propre*

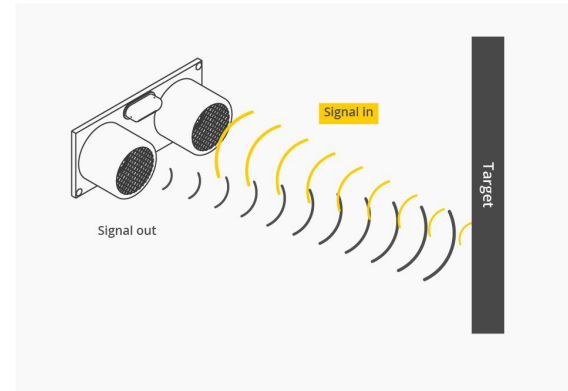
# *Apprendre l'informatique*

- Connaissances en lien aux  **systèmes formels**  (p.ex. code)
  - Connaissances en lien à la  **donnée et ses structurés**
  - Connaissances liés aux  **algorithmes et aux langages**
- Connaissances liées aux  **systèmes physiques**  (p.ex. capteur)
  - Mécatronique : “démarche visant l'intégration en synergie de la mécanique, l'électronique, l'automatique et l'informatique dans la conception et la fabrication d'un produit en vue d'augmenter et/ou d'optimiser sa fonctionnalité” (norme NFE 01-010, 2008)
- Développement de la compétence de  **pensée informatique**  pour résoudre des problèmes en lien aux connaissances en informatique
- Connaissances liées aux  **enjeux de l'informatique dans la société**  (en lien à l'éducation aux médias)

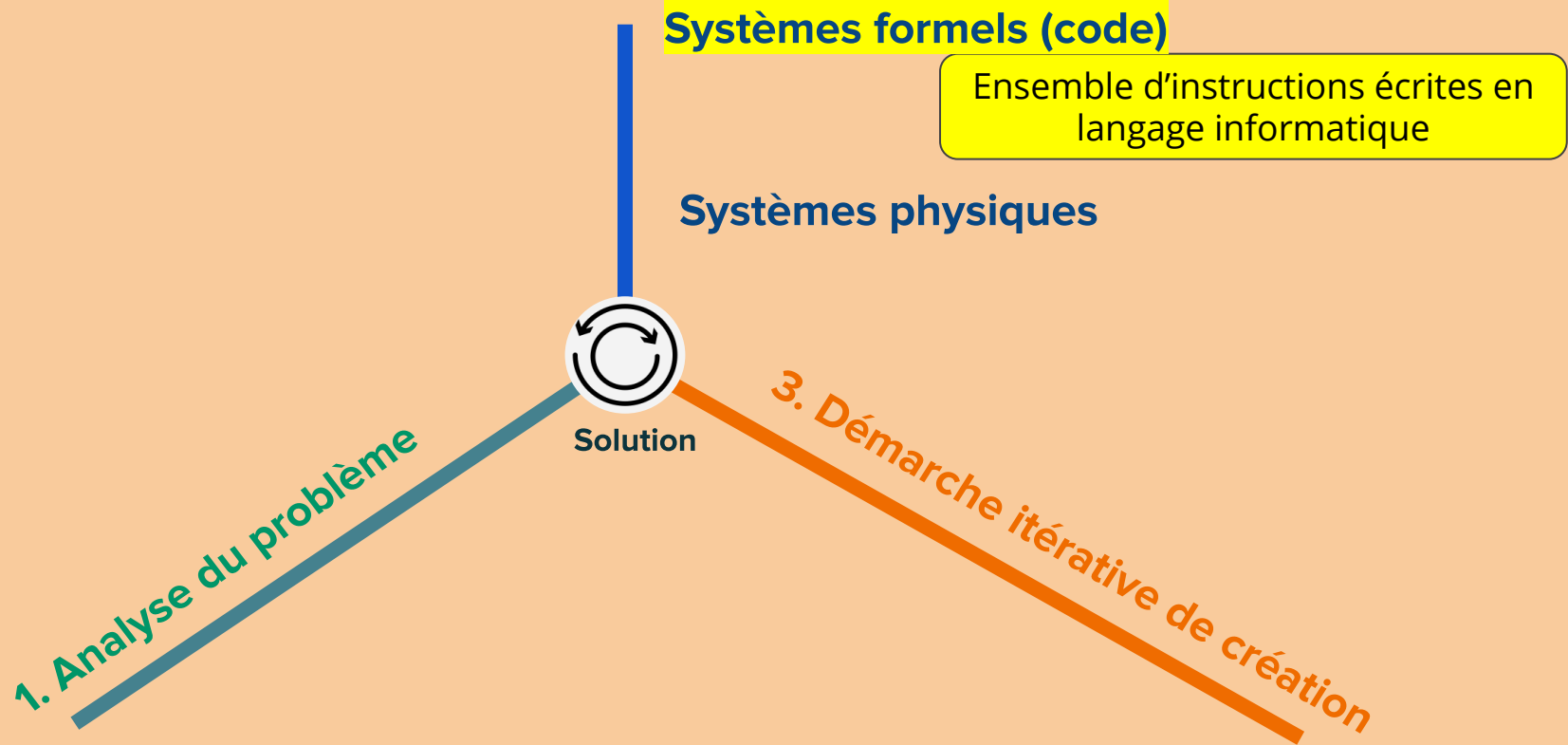
# Connaissances des systèmes formels et physiques (connaissances disciplinaires)

- Connaissances en lien aux **systèmes formels** (p.ex. code)
  - Connaissances en lien à la **donnée et ses structurés**
  - Connaissances liés aux **algorithmes et aux langages**
- Connaissances liées aux **systèmes physiques** (p.ex. capteur)

## Systèmes



# Code VS Programmation



# Introduction à la programmation

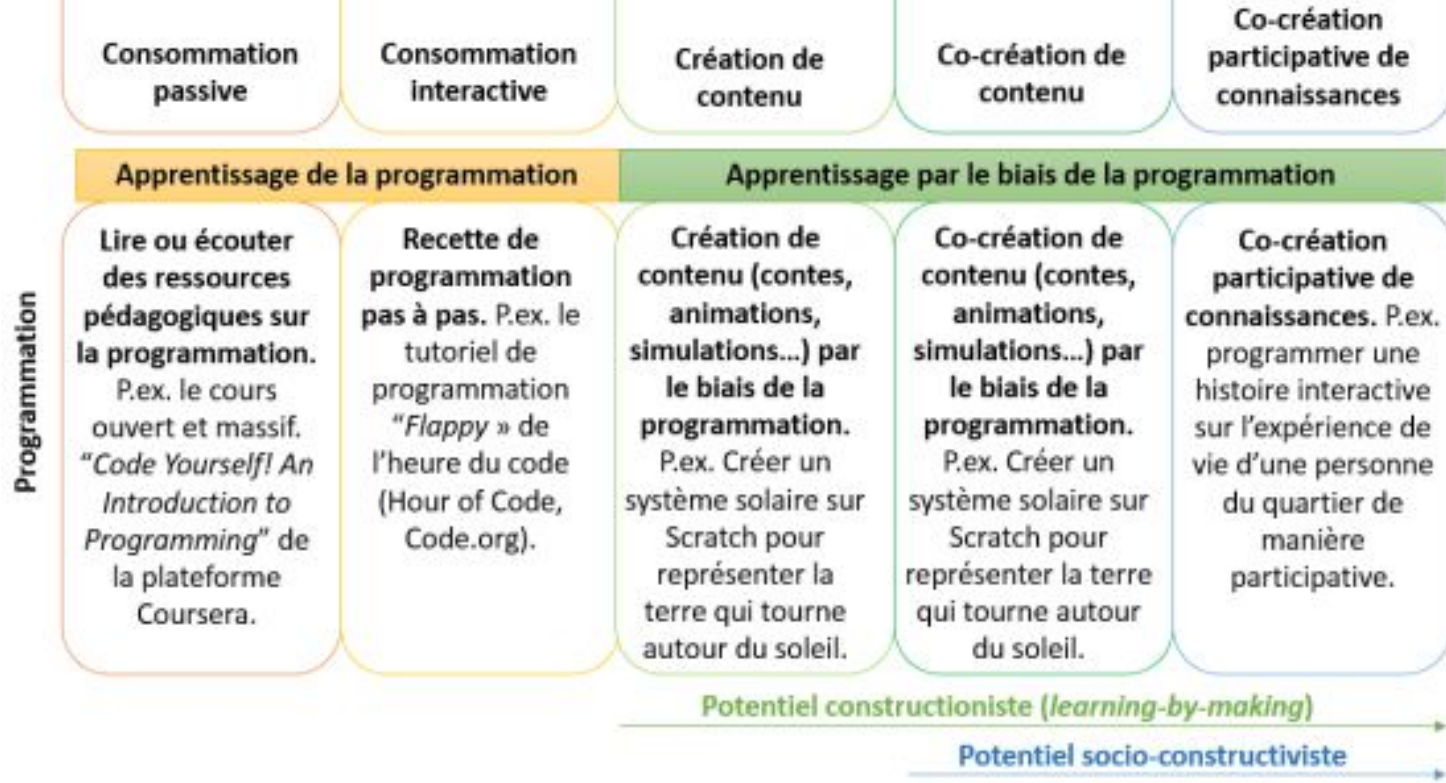
programmation



Après l'analyse d'un besoin ou d'un problème, la **programmation** vise à modéliser et à développer une solution par le biais d'un programme informatique.

La programmation s'exprime par le **code**, qui est un ensemble d'instructions écrites en langage informatique.

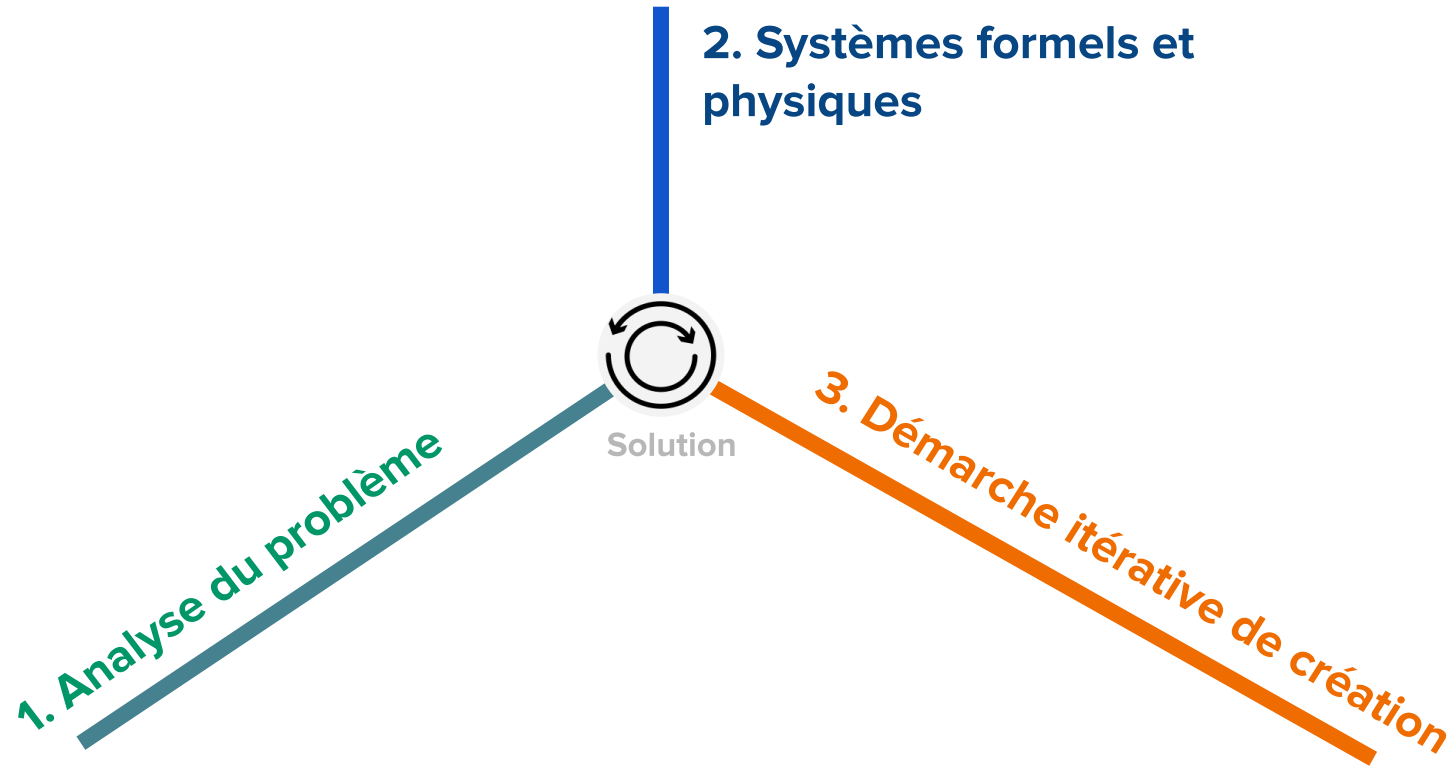
La programmation nous permet de donner des **instructions** à des appareils numériques programmables comme les ordinateurs ou les robots.



De l'apprentissage de la programmation à l'apprentissage par le biais de la programmation. 5 types d'usages en apprentissage de la programmation

# ***Programmation engage la pensée informatique***

***La pensée informatique n'est pas limitée à la programmation***





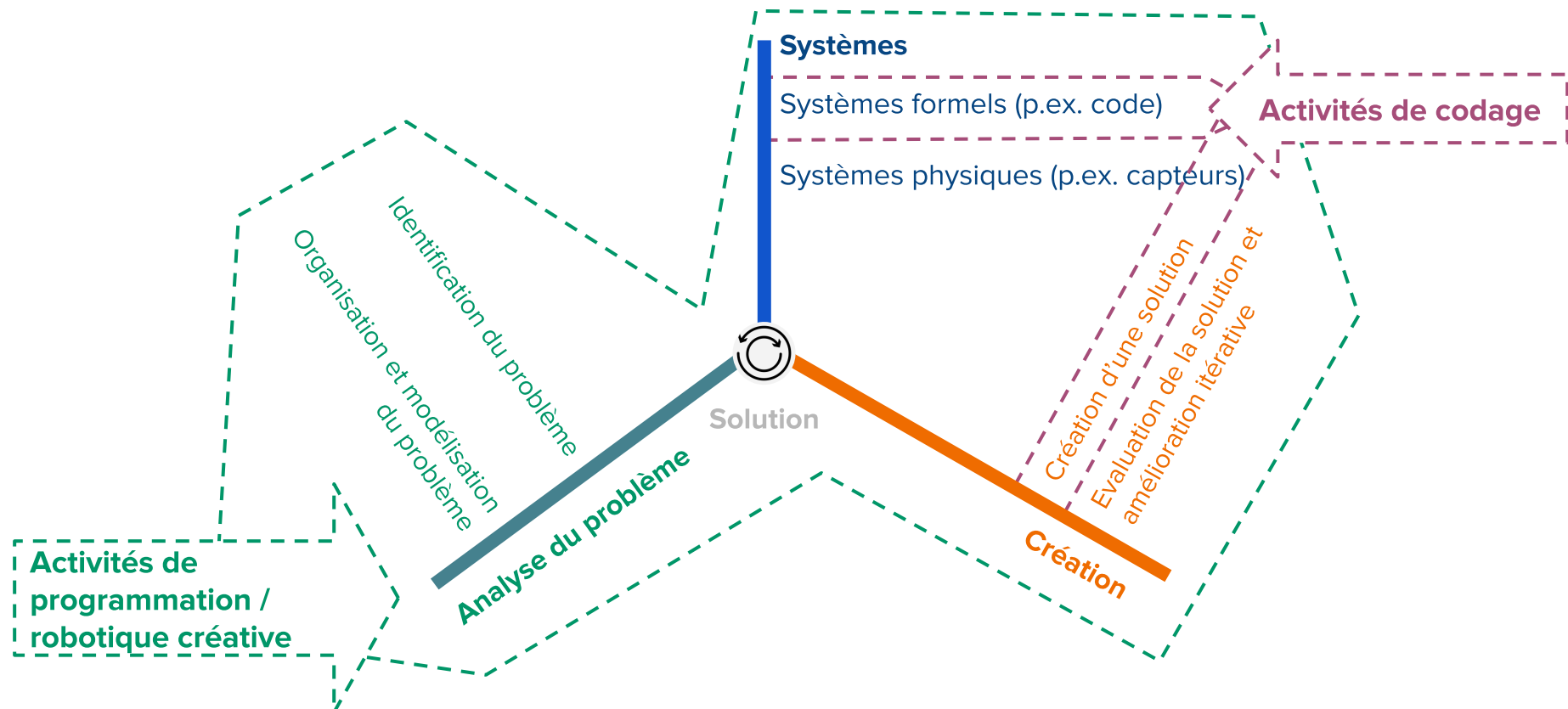
# ***Pensée informatique***

“La pensée informatique conduit à résoudre des problèmes, à concevoir des systèmes et à comprendre le comportement humain, en s'appuyant sur les concepts fondamentaux de la discipline et en y incluant une large collection d'outils intellectuels qui reflètent l'étendue de la science qu'est l'informatique” (Wing, 2008)

“la maîtrise d'objets informatiques et la participation à des activités sociales dans un monde en réseau” (Drot-Delange et Bruillard, 2012)

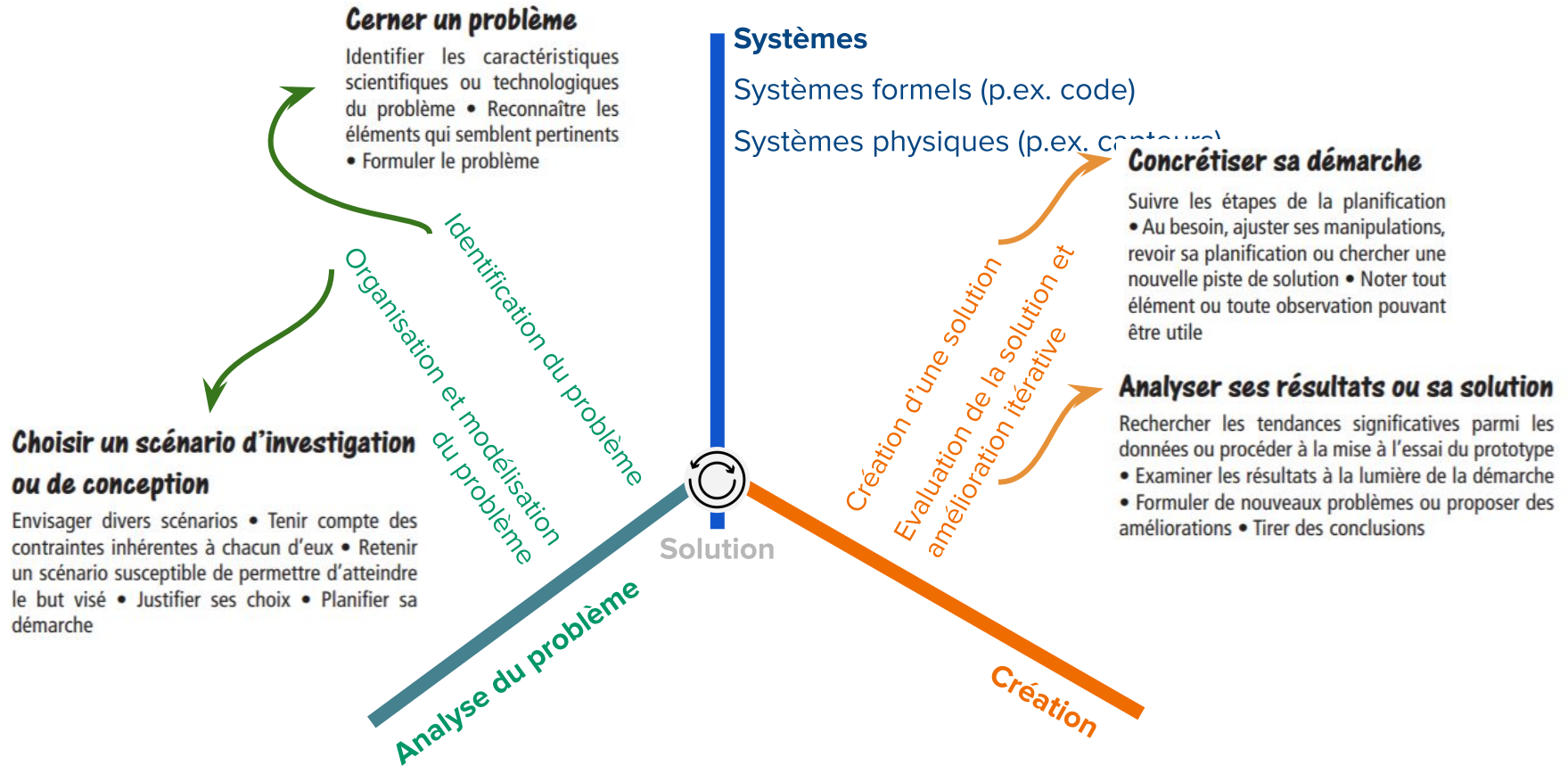
“ensemble de stratégies de pensée cognitive et métacognitive liées à la modélisation de connaissances et de processus, à l'abstraction, à l'algorithmique et à l'identification, la décomposition et l'organisation de structures complexes et de suites logiques” (Romero, Lille & Patino, 2017).

Opérationnalisation de la compétence **pensée informatique** (Wing, 2006) en trois axes et 6 composantes ([Romero, Lepage, & Lille, 2017](#)).

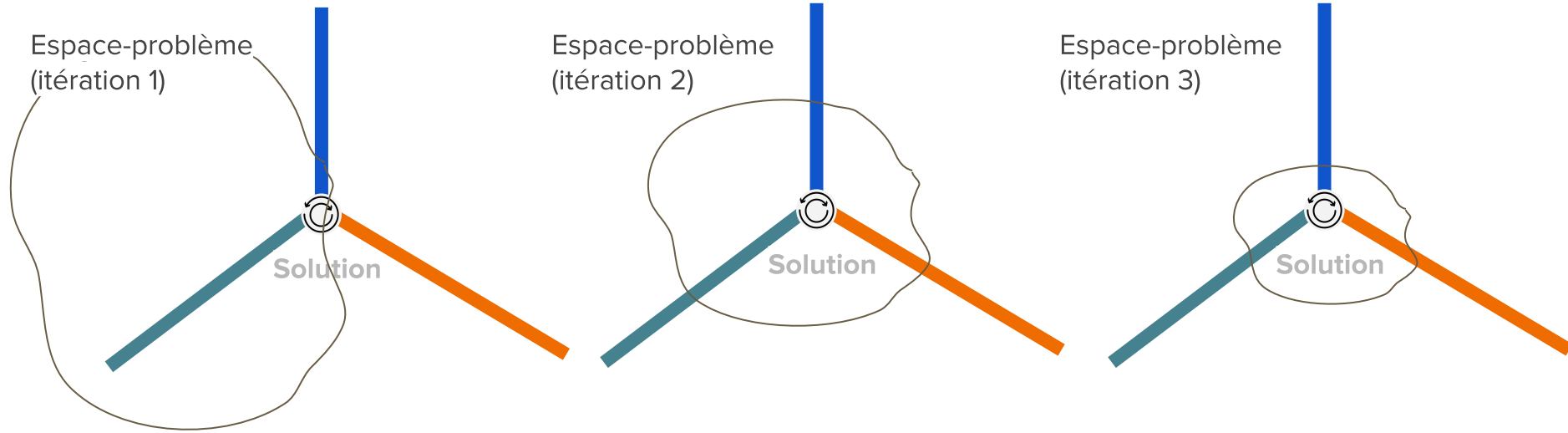


<i>Categories</i>	<i>CT6 Components</i>	<i>CT6 Subcomponents and Definition</i>
Problem analysis	Identify the problem (COMPO1)	Analysing information and understanding the problem and its components
	Organise and model the problem (COMPO2)	Creating models based on possible solutions
Systems	Formal systems (COMPO3)	Sequencing – understanding the series of steps for a task
		Loops – figuring out ways to run the same steps without repeating them from scratch
		Parallelism – enabling processes/steps to run simultaneously
		Events – triggering something to happen based on a previous step or action
		Algorithms – describing them in simple terms and understanding “blocks, objects, operators and instructions” [34]
		Conditional logic – making decisions based on if-then-else logic
		Building algorithms – modelling a process at a level where it can be processed by a machine
		Code optimisation – reusing and modifying code
	Physical systems (COMPO4)	Understanding features of the technology used and its benefits and limitations
Creation	Devise a solution (COMPO5)	Identifying the programming features to be used to accomplish a task
		Using programming techniques to complete a task
		Completing the programming task successfully
	Evaluate solutions and conduct iterations (COMPO6)	Testing the solutions to see if they work; if they don't work, analysing the reasons behind it and finding solutions to problems that arise

# Pensée informatique et résolution de problèmes



# *Pensée informatique et résolution de problèmes*



# Analyse du projet *makered* **ArcadeTable** (Davidson)



Choix des systèmes dans un écosystème riche (fablab, makerspace)

## Systèmes

Systèmes formels (p.ex. code)

Systèmes physiques (p.ex. capteurs)

Program emulation station  
Program joystick  
Extract and transfer roms

Layout design  
Joystick and buttons  
Fit the monitor

Identification et modélisation du problème

Identification du problème

Analyse du problème

Solution

Création d'une solution  
Évaluation de la solution et amélioration itérative

Création

Concept of arcade table  
Concept of play  
Usability issues

Prototype  
Measurements  
Hardware connections

Prototype  
Measurements  
Hardware connections  
Setting-up of the emulation station

# Analyse du jeu en pensée informatique *Crabs & Turtles* (Tsarava, Moeller, & Ninaus, 2018)



Organisation et modélisation du problème

Identification du problème

Understanding the game instructions and rules .

Observe and identify restrictions

Analyse du problème

Solution

Systèmes

Systèmes formels (p.ex. code)

Systèmes physiques (p.ex. capteurs)

Introduction of coding concepts: sequences, constants/variables, conditionals, events, loops

Création

Création d'une solution

Evaluation de la solution et amélioration itérative

Design a game strategy

Build the respective sequence of commands

Adapt strategy by:

- identifying previous mistakes
- observing others' successful moves
- considering the visual feedback provided by rewards

# Analyse de la tâche **grue intelligente** (Kamga, dir. Romero, Barma)



Organisation et modélisation  
du problème

Identification du problème

Analyse du problème

Solution

**Systèmes**

Systèmes formels (p.ex. code)

Systèmes physiques (p.ex. capteurs)

Création d'une solution

Evaluation de la solution et  
amélioration itérative

**Création**

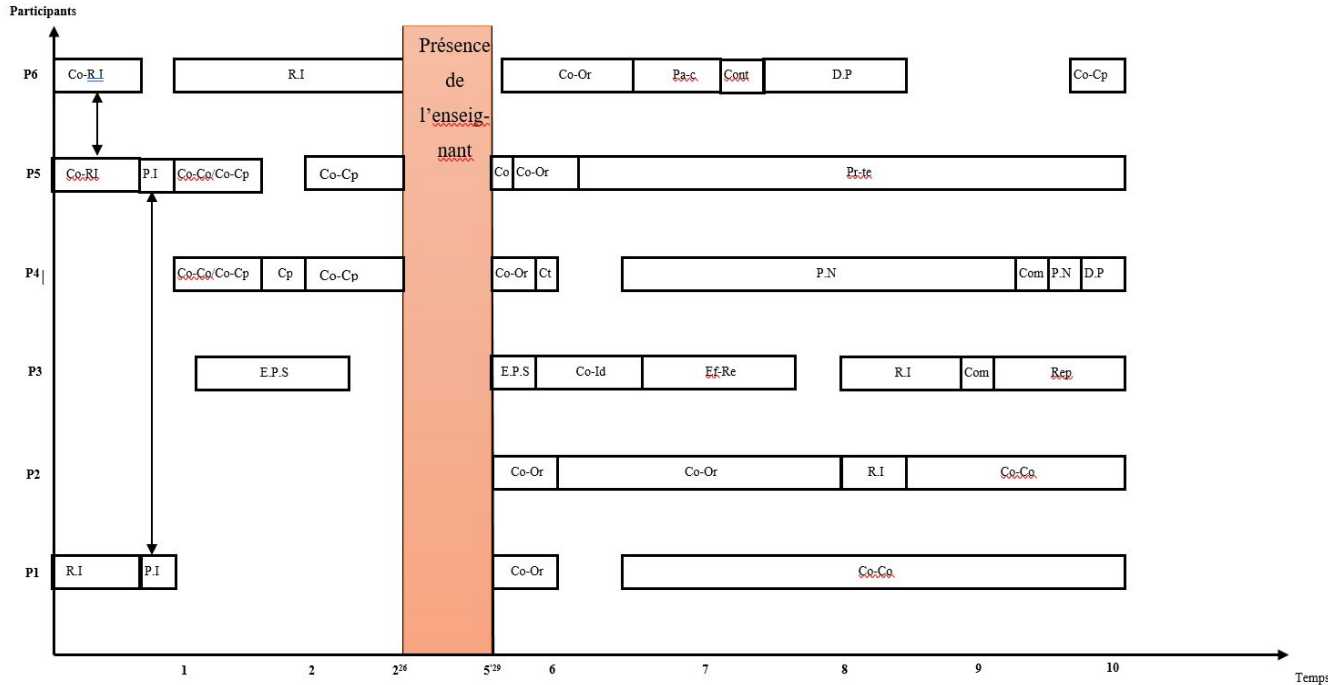
C1: Modelise the smart crane and identify the position of sensors, engine and other components  
C2: Understand the the activity's instructions

C5: plan the construction of Crane and the programming  
C6: modify the structure of crane or the programm when the solution is not good

C3 : Use Scratch to programme the crane  
C4 : Use the Lego's Block for the construction of crane

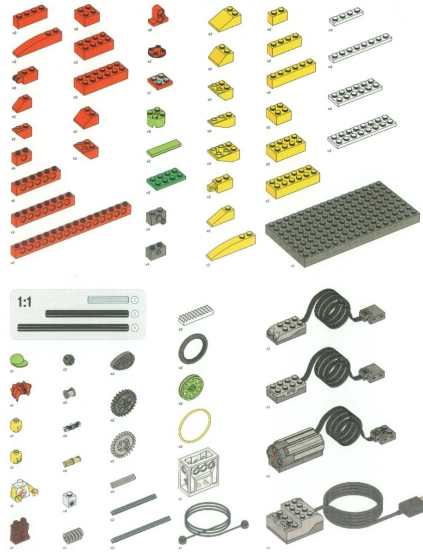


# Identification des difficultés des futur(e)s enseignant(e)s du primaire et du préscolaire en lien avec la compétence de résolution collaborative de problèmes (RCP) (Kamga, dir. Barma, Romero)



**Co-R.I** ; Corecherche de l'information; **R.I** : Recherche de l'information; **P.I** : Partage de l'information; **Co-Or** : Co-organisation; **Or** : Organisation; **Pa-C** : Partage de connaissances; **Co-co** : Coconstruction; **Co** : Construction; **D.P** : Demande de point de vue des autres membres de l'équipe; **Cont** : Contrôle; **Cp** : Compréhension; **Co-Cp** : Cocompréhension; **E.P.S**: essai des pistes de solution; **Ef-Re**: Effacer et recommencer; **Rep**: Reproduire un exemple trouvé sur internet

# Réduction des degrés de liberté créative



Une tâche développée pour l'analyse de la pensée informatique dans le cadre du projet ANR CreaMaker (Romero, Vieville, Cassone, Basiri, DeBlois, Barma et al).



# Analyse de la tâche **CreaCube** (projet [ANR CreaMaker](#), Romero, Vieville, Cassone, Bassiri et al)



Understanding the problem-situation  
Concept of autonomous vehicle

Organisation et modélisation du problème  
Jentification du problème  
**Analyse du problème**

Solution

**Systèmes**

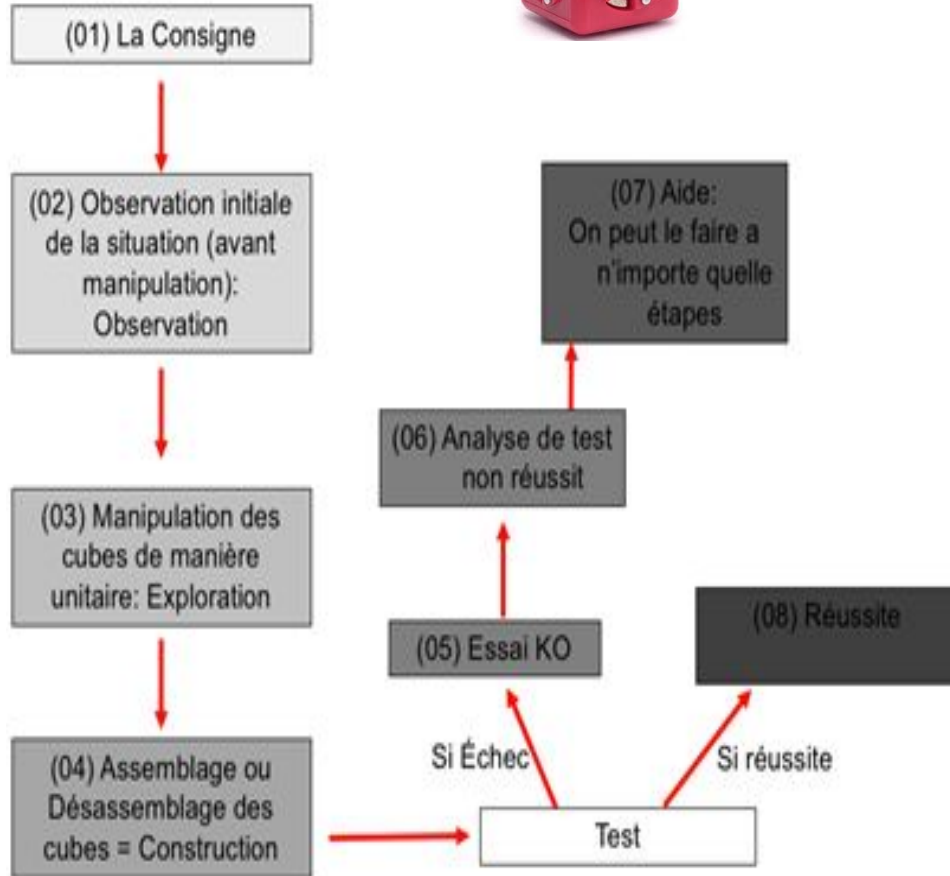
Systèmes formels (p.ex. code)

Systèmes physiques (p.ex. capteurs)

Magnets  
Sensors  
Actuators  
Electric circuit  
Cubes assembled as a system

**Création**  
Création d'une solution  
Evaluation de la solution et amélioration itérative

Creating a solution by assembling by inverting the distance sensor signal

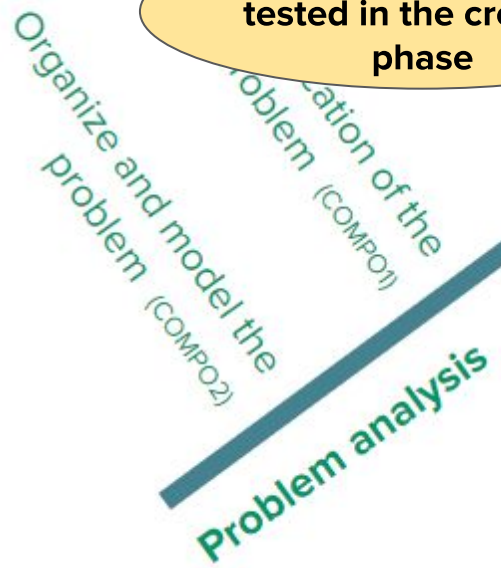
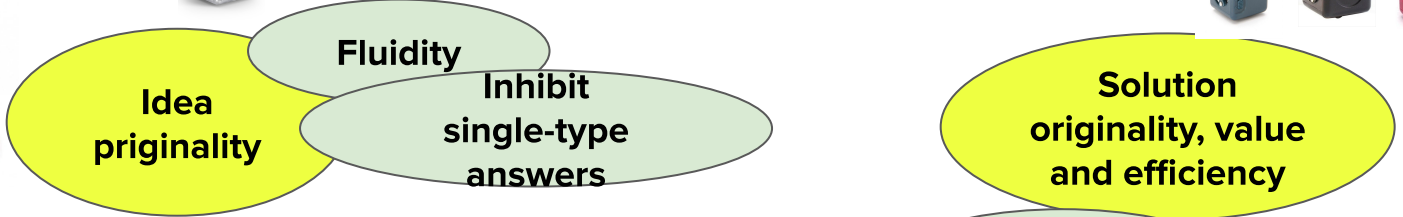


Un modèle de tâche bien établi permettant de comparer les observables en lien à la pensée informatique en contexte individuel et collaboratif.

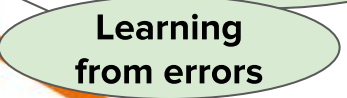
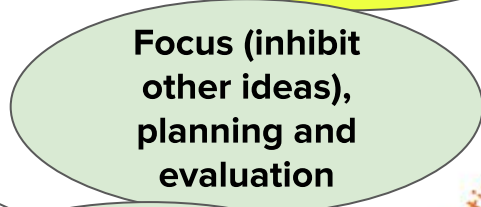
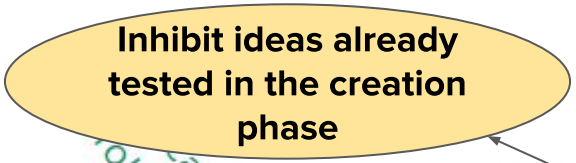
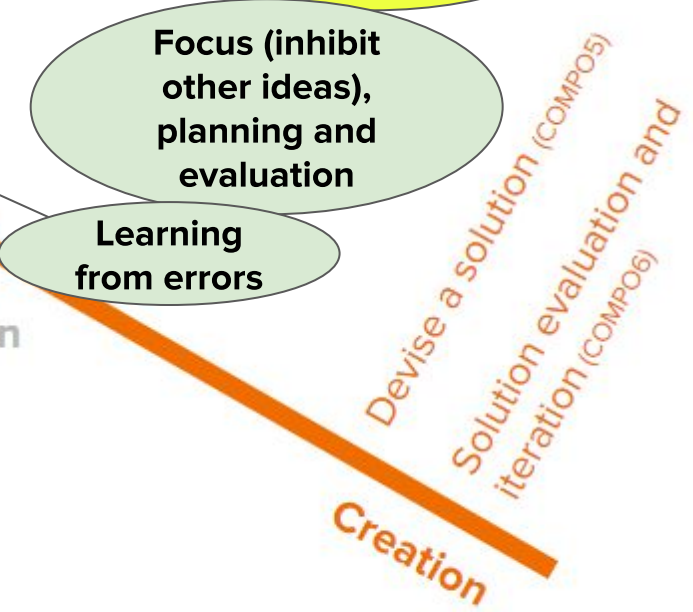
D'un point de vue didactique, la situation vise placer l'apprenant dans une démarche où il doit **mobiliser et adapter des connaissances pour en élaborer de nouvelles** (Brousseau, 1998; Brousseau et Warfield, 2014).

La situation-problème « place l'apprenant devant une série de décisions à prendre pour atteindre un objectif qu'il a lui-même choisi ou qu'on lui a proposé, voire assigné » (Perrenoud, 1995, p.6).

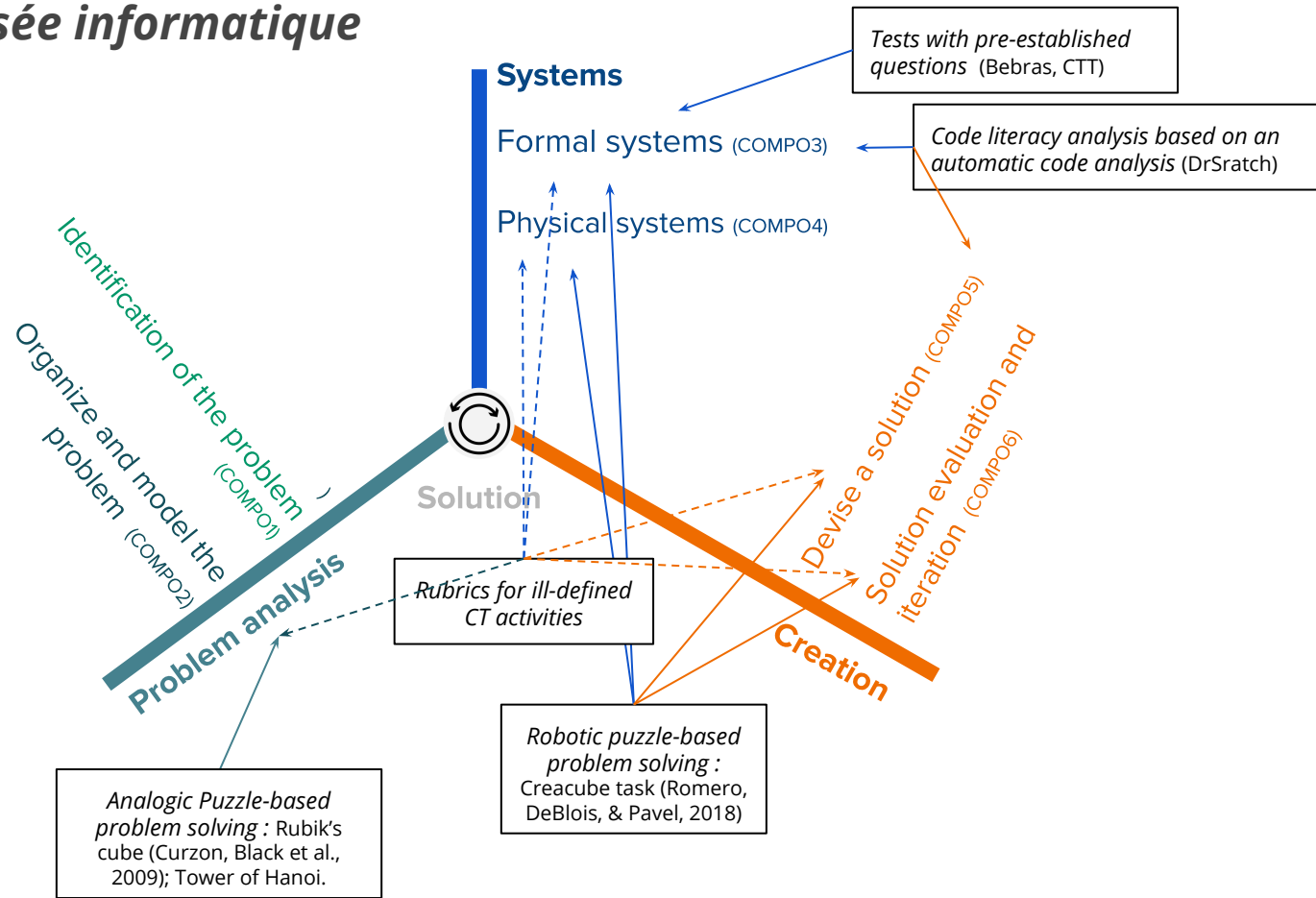
Face à un artefact nouveau, ce sont les prises de décisions de l'apprenant qui permettront d'**explorer par la manipulation** et de poser une réflexion sur ses actions afin de repérer le fonctionnement de l'artefact inconnu. Bélanger et al., (2014) ont observé comment la **variété de procédures exploitées dans une tâche de résolution de problèmes** comme autant de manifestations de créativité au moment d'arrimer et d'organiser le système de connaissances des élèves aux contraintes d'un problème.



Solution



# Evaluation de la pensée informatique





# Références des études de cette présentation

Romero, M. (2019). [La programmation n'est pas que technologique. Programmer : une démarche sensible, culturelle et citoyenne pour résoudre des problèmes](#) *Spectre* (49).

Romero, M., Lepage, A., & Lille, B. (2017). [Computational thinking development through creative programming in higher education](#). *International Journal of Educational Technology in Higher Education*, 14(1), 42.

Menon, D., Viéville, T., & Romero, M. (2019). [Computational thinking development and assessment through tabletop escape games](#). *International Journal of Serious Games*, 6(4), 3-18.

Romero, M. (2016). De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative. *Formation et profession*, 24(1), 87-89.  
<https://doi.org/10.18162/fp.2016.a92>



# Apprendre l'informatique, un défi à relever à plusieurs : Communauté d'Apprentissage de l'Information (CAI)



## Création d'une communauté d'enseignants engagés dans l'apprentissage de l'informatique



**Thierry Massart**  
Université Libre de Bruxelles



**Sébastien Hoarau**  
Université de la Réunion



**Olivier Goletti**  
Université catholique de Louvain



**Kim Mens**  
Université catholique de Louvain



**Margarida Romero**  
LINE, Université Côte d'Azur



**Vassilis Komis**  
Université de Patras



**Patricia Corieri**  
La Scientothèque



**Thierry Vieville**  
LINE, INRIA,  
Université Côte d'Azur



**Maryna Rafalska**  
LINE, Université Côte d'Azur



**Gabriel Parriaux**  
HEP Vaud



**Leïla Meziane**  
La Scientothèque



**Jennifer Christophe**  
La Scientothèque



Laboratoire d'Innovation et  
Numérique pour l'Éducation

UNIVERSITÉ CÔTE D'AZUR



INSPÉ Institut national  
supérieur du professorat  
et de l'éducation  
Académie de Nice



# *Apprendre l'informatique : du code à la pensée informatique*

Merci de votre  
attention

**@margaridaromero**

Margarida.Romero@unice.fr

Dir. Laboratoire d'Innovation et Numérique pour l'Éducation (LINE)

