# Spécifier le parcours d'enseignement-apprentissage

Le but de cette activité est d'**établir un parcours d'enseignement-apprentissage** : c'est-à-dire transformer les processus d'apprentissage du parcours d'apprentissage, en décidant du **niveau de dévolution** et d'y associer des **tâches** utiles, exigeantes, faisables et engageantes.

***Les BUTS d'apprentissage :***
Pour illustrer cette activité de scénarisation, j'ai choisi de limiter l'étude à un parcours d'apprentissage centré sur le **langage de programmation graphique (comme SCRATCH et LEGO)**. Cet apprentissage se distingue de celui de l'algorithmique, c'est-à-dire de la capacité à résoudre un problème. Il vise uniquement la formalisation de la solution à ce problème.

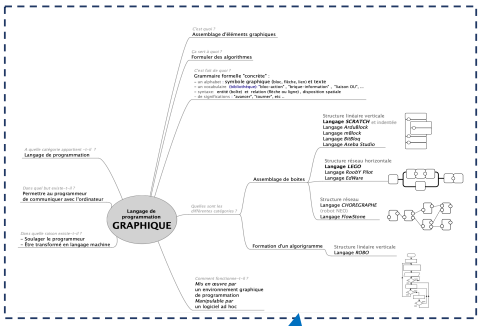
Le fait de **limiter** le champ notionnel à celui de « langage de programmation graphique » n'est pas un choix arbitraire: il s'agit de trouver un distance cognitive acceptable entre les connaissances des élèves et le savoir visé.
En effet, les connaissances en psychologie cognitive* nous disent que:

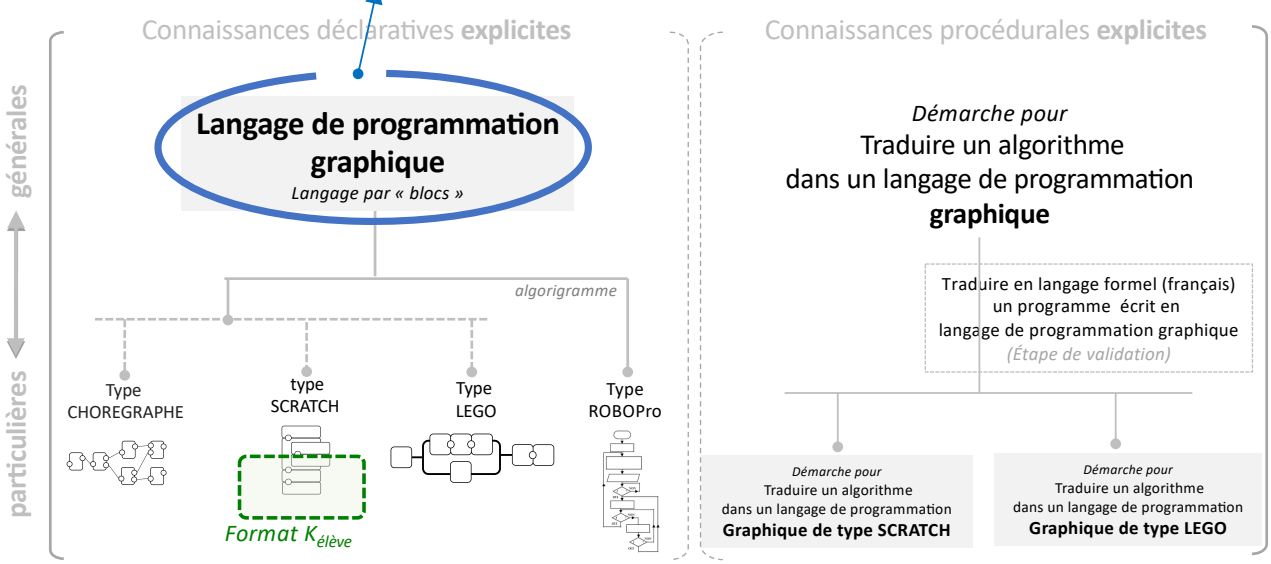
- Le facteur unique le plus important qui influence l'apprentissage est ce que l'apprenant sait déjà. Il faut s'en assurer et lui enseigner en conséquence (Ausubel, 1968).
En l'occurrence ici, le langage SCRATCH (qui est le langage *quasi*-unique présent dans les ressources didactique, comme les manuelles scolaire, ressources numériques ou recommandations institutionnelles)

- Apprendre demande d'arrimer les nouvelles connaissances dans une structure cognitive, si possible à des catégories ni trop générales ni trop spécifiques.
En l'occurrence ici, celle associée au concept de langage de programmation graphique (*précurseur* à celui de langage de programmation).

- Choisir l'endroit où placer un nouveau concept dans cette structure est la base de la création d'un plan ou d'une progression.
En l'occurrence ici, la structure-K associée au « langage de programmation graphique » (précurseur à celui de langage de programmation) est une composante de celle de « langage formel » (qui est élaborée à partir des connaissances des élèves).

Quant à la structure cognitive, dont les cartes de connaissances (carte des formats-K et carte heuristique) en sont des représentations externes, elle doit permettre d'éviter les phénomènes de *sous-généralisation* ou de *sur-généralisation*.
En l'occurrence ici, limiter l'enseignement de la programmation au langage SCRATCH peut laisser croire à l'élève que ces propriétés sont représentatives d'un langage de programmation. Autrement dit, l'exemple fait office de catégorie conceptuelle ou procédurale.
On peut aussi voir ici un moyen pour lever les difficultés liées au changement de registres sémantiques dans un langage formel (Duval, 1993, 1995)




(*) : https://fr.wikibooks.org/wiki/Psychologie_cognitive_pour_l%27enseignant/Faire_des_liens#Les_connaissances_antérieures_:_de_l'ordre_primaire_des_notions
Ausubel, D. P. 1968: Educational Psychology: A Cognitive View. Holt, Rinehart and Winston Inc. New York.
Duval R. (1993). Registres de representations semiotiques et fonctionnement cognitif de la pensee. Annales de Didactique et de Sciences Cognitives, ULP, IREM Strasbourg. 5, 37-65.
Duval, R. (1995). Sémiosis et pensée humaine: registres sémiotiques et apprentissages intellectuels. Berne: Peter Lang.

## *Spécifier le parcours d'apprentissage :*

*Analyser la compétence :*

Comme le précisent les recommandations institutionnelles (Eduscol,2018), les connaissances sur les langages de programmation (graphique) font l'objet d'un apprentissage tout au long au cycle 4, de manière spiralaire.

Le programme de la technologie au collège précise que ces connaissances doivent permettre de « lire, modifier et écrire un programme dans un langage graphique donné ».

En l'occurrence ici,

- *LIRE un programme écrit dans un langage donné* nécessite de mobiliser la démarche de « traduction dans un langage formel (le français) ou dans un langage algorithmique naturel (en français)».

MAIS aussi de savoir reconnaitre le langage et savoir identifier et convoquer ses attributs (alphabet, vocabulaire, syntaxe, …).

- *ÉCRIRE un programme dans un langage donné* nécessite de mobiliser la démarche de « traduction d'un algorithme dans un langage de programmation ».

MAIS cela nécessite aussi de connaitre le langage et de convoquer ses attributs (alphabet, vocabulaire, syntaxe ).

- *MODIFIER un programme dans un langage donné* nécessite de mobiliser les deux démarches précédentes et les connaissances déclaratives associées.

Cette première déclinaison suggère de placer l'apprentissage des connaissances associées à la compétence *MODIFIER* après celles de *LIRE* et d'*ÉCRIRE*.

La *LECTURE* d'un programme porte sur un objet construit, résultat d'une démarche d'*ÉCRITURE*. Selon Bruner (2015), il est préférable de comprendre l'objet avant de comprendre sa construction.

Autrement dit, l'élaboration du concept de « langage » est un *précurseur* à l'élaboration de la démarche d'« utilisation du langage ».

*Identifier et caractériser les tâches d'apprentissages :*

Ces tâches d'utilisation des connaissances, caractérisant la compétence, permettent d'identifier des tâches d'apprentissage pertinentes pour spécifier le parcours d'apprentissage. Ainsi :

Une tâche de LECTURE d'un programme écrit dans un langage de programmation graphique peut correspondre à des tâches d'apprentissage comme :

- une tâche d'étude de cas, supportant un processus d'abstraction déclarative (attributs du concept). En l'occurrence ici, j'étudie un programme écrit en langage SCRATCH dans un environnement SCRATCH (logiciel) pour en extraire son alphabet, son vocabulaire et sa syntaxe.
- une tâche d'étude de cas, supportant un processus d'assimilation déclaratif (reconnaitre un élément d'alphabet ou de syntaxe) ou un processus d'assimilation procédurale (traduire le programme en langage formel algorithmique ou quasi-naturel).

Une tâche d'ÉCRITURE d'un programme dans un langage particulier de programmation graphique peut correspondre à des tâches d'apprentissage comme :

- une tâche de résolution de problème, supportant un processus d'assimilation procédurale. En l'occurrence ici: traduire un algorithme, formulé dans langage formel algorithmique ou quasi-naturel, dans un langage de programmation graphique particulier.

Cependant les travaux issus de la Théorie de la Charge Cognitive montrent qu'il est préférable, dans le cas d'apprentissage de connaissances procédurales nouvelles, de proposer d'abord des exemples de problème résolus, puis des problèmes à compléter avant de proposer des problèmes à résoudre sans aucune aide. Mais aussi d'alterner problèmes résolus et problèmes à résoudre, en intégrant des problèmes à compléter.

Cette préconisation à propos des problème résolus renvoie donc à une autre tâche d'apprentissage:

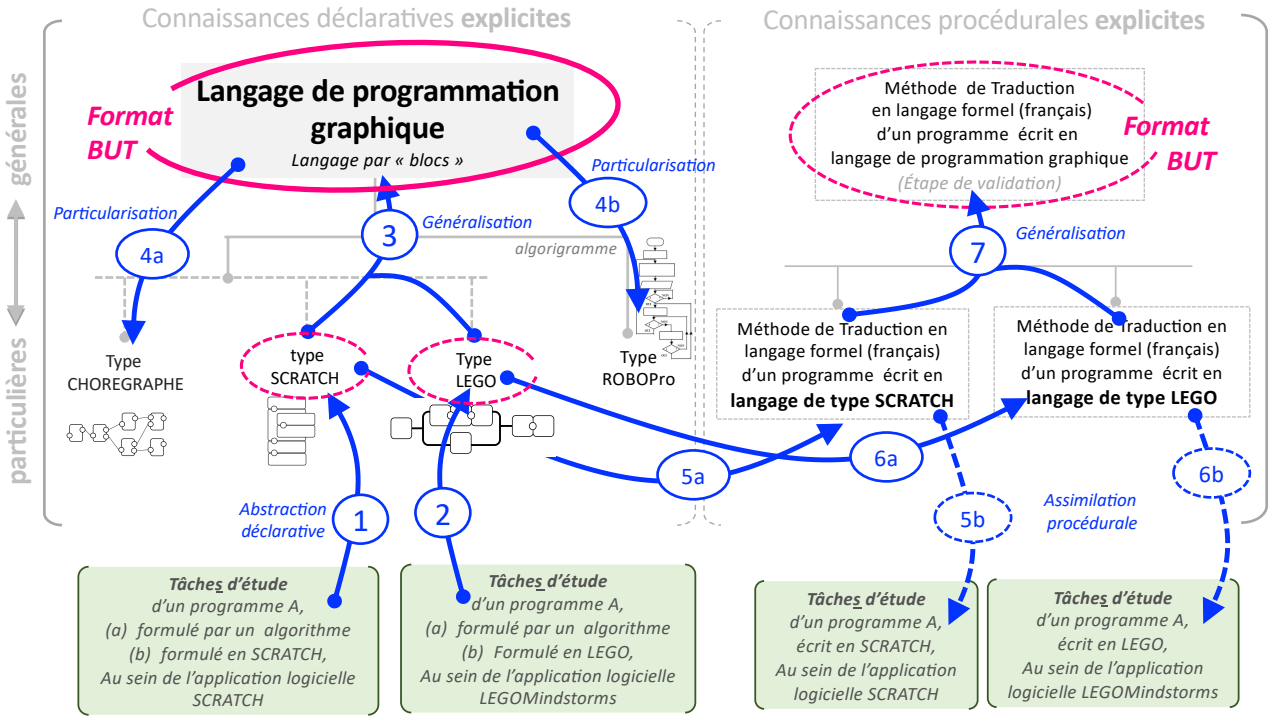
- Une tâche d'étude de cas, supportant un processus d'abstraction procédurale. En l'occurrence ici la démarche de traduction d'un algorithme, formulé dans langage formel algorithmique ou quasi-naturel, dans un langage de programmation graphique particulier.

Eduscol, 2018). Ressources d'accompagnement du programme de technologie au cycle 4 . https://eduscol.education.fr/cid99549/ressources-technologie-c4.html
Bruner, J. (2015). Le développement de l'enfant: savoir faire, savoir dire. Presses universitaires de France.

**Spécifier le parcours d'apprentissage :**

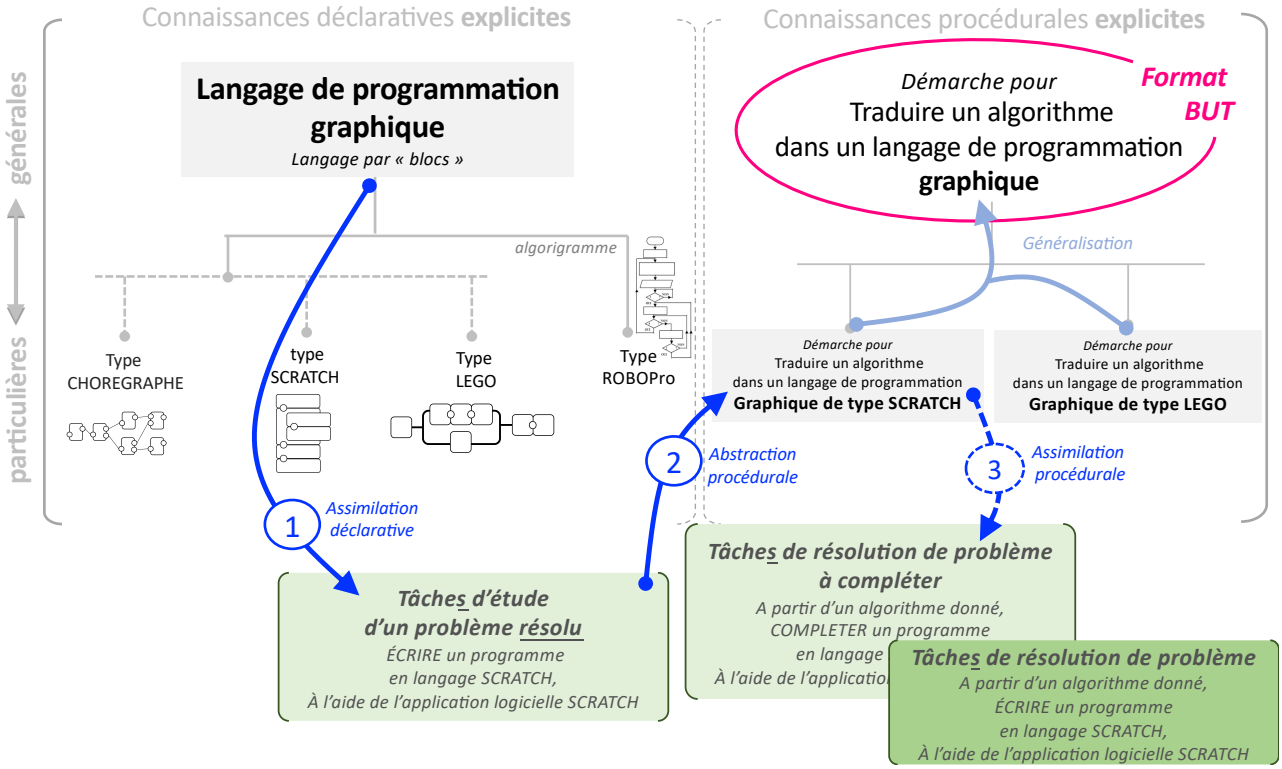
A partir des analyses précédentes, j'ai choisi de retenir un parcours d'apprentissage qui s'appuie sur le langage SCRATCH. La synthèse de Romero et al. (2018) permet d'établir l'intérêt de cet outil (et ses limites). Je fais le choix d'articuler ce parcours peut autour de deux phases :

**Phase 1 :** Élaborer les connaissances associées à la compétence « LIRE un programme écrit dans un langage de programmation graphique »




**Phase 2 :** Élaborer les connaissances associées à la compétence « ÉCRIRE un programme dans un langage de programmation graphique ».

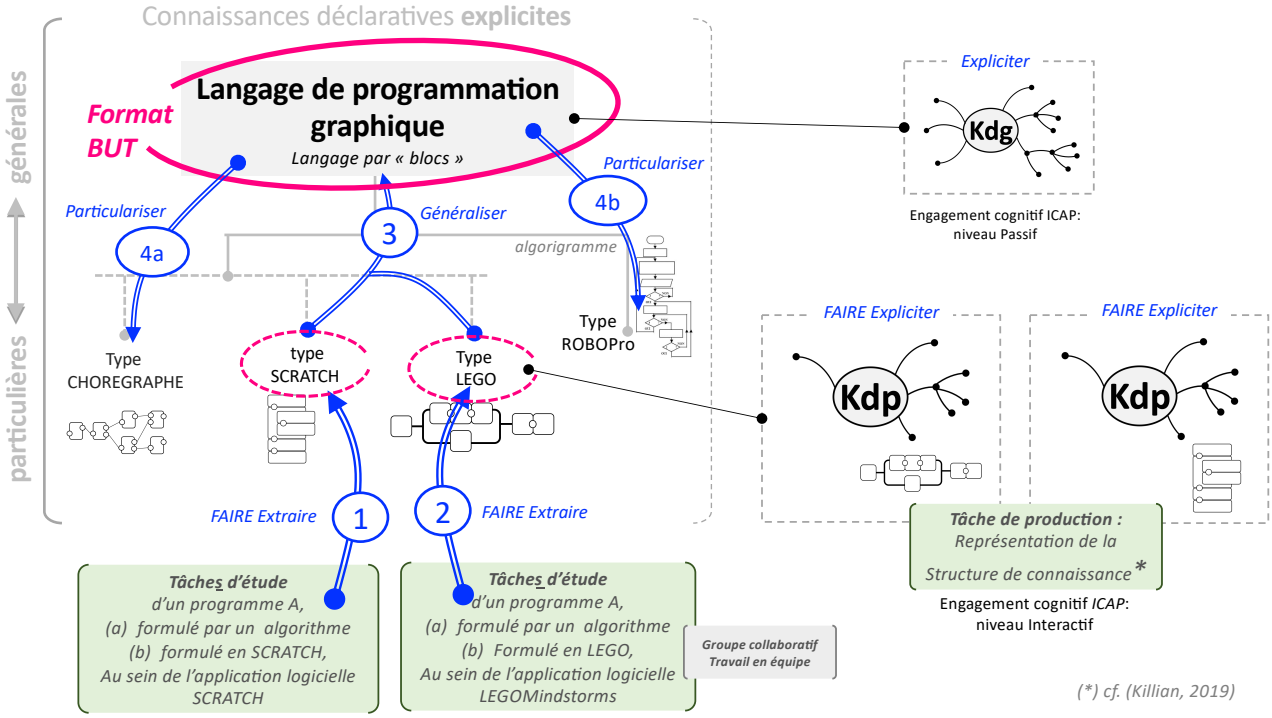
Pour des raisons de lisibilité, je fais apparaitre uniquement le parcours associé au langage de type SCRATCH. L'approche est analogue pour les autres types de langages.



**Spécifier le parcours d'enseignement-apprentissage :** (Étude limitée au volet déclaratif de la phase 1)

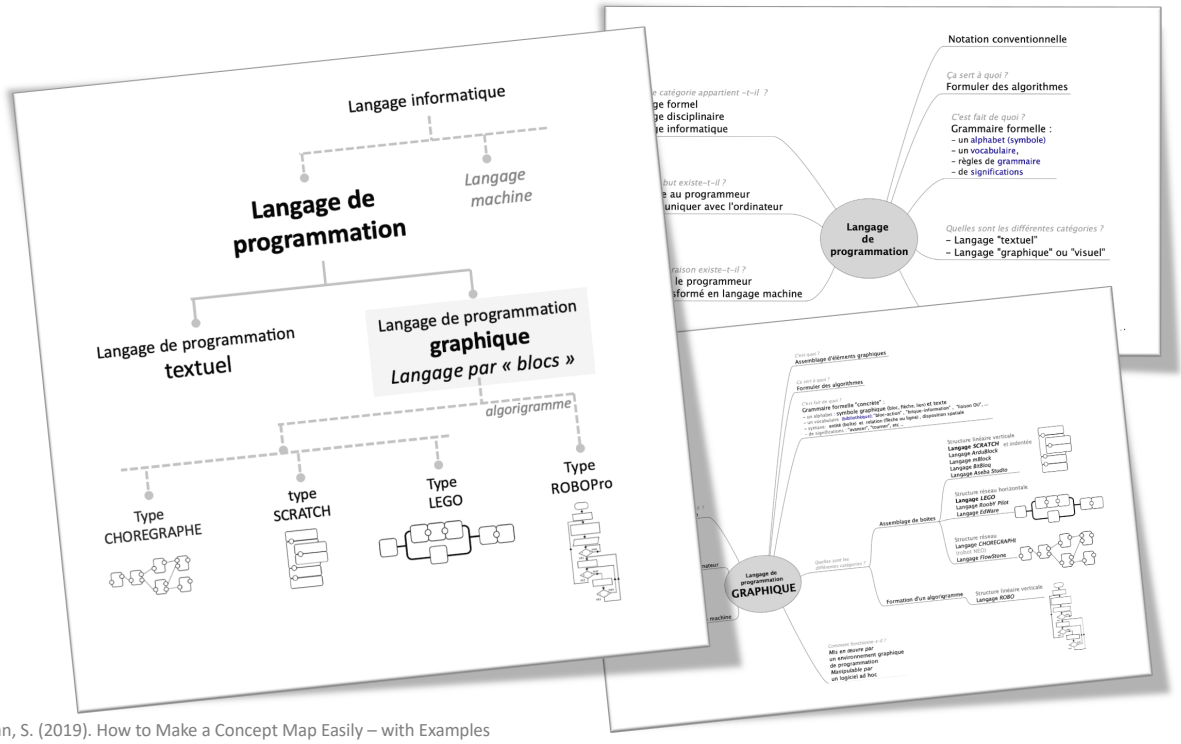
Le choix qui est fait ici est de dévoluer :

- à l'élève l'élaboration des connaissances particulières (Kdp), via un des processus *FAIRE Extraire* et *FAIRE Expliciter. (processus 1 et 2). L'enseignant soutiendra ces processus par des tâches d'étayage écrite et orale (via des guidances et un dialogue d'aide).*
- à l'enseignant l'élaboration des connaissances générale (Kdg), via le processus emblématique *FAIRE comprendre, organisé autour des processus EXPLICITER-FAIRE Généraliser (processus 3) EXPLICITER-Particulariser, (processus 4a et 4b). Ces processus d'enseignement seront* soutenus par des tâches *de maïeutique et d'exposé.*




*Esquisser les traces écrites :* (Étude limitée au volet déclaratif de la phase 1)

Les structures cognitives vues précédemment permettent d'organiser les nouvelles informations que l'élève doit apprendre, facilitant ainsi le rappel ultérieur.
En l'occurrence ici, les cartes de connaissances (cartes des formats et cartes heuristiques) sont des documents qui ont vocation à devenir des traces écrites, constitutives de la leçon.




Killian, S. (2019). How to Make a Concept Map Easily – with Examples
https://www.evidencebasedteaching.org.au/concept-mapping-complete-guide/