

Un référentiel de compétences en programmation pour construire des ressources et des évaluations

Sophie Chane-Lune

Laboratoire d'Informatique et de Mathématiques, Université de La Réunion, France
sophie.chane-lune@univ-reunion.fr

Résumé A partir du référentiel de compétences en programmation RCP, d'une part, et du catalogue de tâches aligné sur ce référentiel, d'autre part, nous avons pu identifier des difficultés de programmeurs débutants. Nous proposons ici d'étudier l'opérationnalité et la cohérence de ce référentiel et de son catalogue de tâches associé, en l'utilisant pour construire des ressources et des évaluations et pour catégoriser des tâches complexes. Les premiers résultats nous encouragent à poursuivre les expérimentations dans l'objectif de consolider le référentiel pour conclure à propos de son opérationnalité et de sa cohérence.

Introduction

L'introduction de l'informatique en tant que discipline scolaire en France [1] s'est effectuée avec la mise en place de programmes en Numérique et Sciences Informatiques (NSI) pour les classes de première et de terminale [7]. Le préambule commun à ces programmes d'informatique au lycée fait explicitement référence aux compétences en programmation, telles que *modéliser un problème* [...] ou encore *développer des capacités d'abstraction et de généralisation*. Cependant la suite des programmes n'y fait plus référence et est organisée par contenus, dans lesquels sont notées des capacités attendues. Il est par conséquent difficile pour l'enseignant, de mettre en œuvre une approche par compétences, car il doit distinguer pour chaque compétence, les différents contextes où la compétence peut intervenir.

C'est pour répondre à cette problématique que nous avons élaboré le **Référentiel de Compétences en Programmation (RCP)** [3], en croisant les compétences génériques de la pensée informatique au sens de Wing [10] et les domaines et notions, des programmes de la spécialité NSI.

Concernant la pensée informatique, nous avons fondé notre réflexion sur les définitions opérationnelles proposées par Selby et Woollard à savoir la capacité à penser en termes d'abstraction, de décomposition, à penser « algorithmiquement », et à penser en termes d'évaluation et de généralisation [9]. Nous avons choisi de nommer les **compétences génériques** suivantes [4] pour décrire la pensée informatique : *abstraire*, *décomposer*, *anticiper*, *évaluer* et *généraliser*, auquel nous avons proposé d'ajouter la compétence *modéliser*. Nous pensons en effet qu'il est utile, en particulier pour les programmeurs débutants, de distinguer la compétence *modéliser* de la compétence *abstraire* à laquelle elle pourrait

être rattachée, parce que nous avons pu observer que des élèves maîtrisent l'une et pas l'autre.

Notre méthode a consisté à réécrire l'ensemble des capacités attendues des programmes d'informatique au lycée relatives au domaine de la programmation, en utilisant ces compétences génériques. Lors de ce premier travail [3], seules les compétences abordées en classe de seconde en Sciences Numériques et Technologie (SNT) ont été présentées et analysées. L'utilisation du référentiel RCP et du catalogue de tâches élémentaires alignées sur ce référentiel nous a permis d'analyser quelques difficultés rencontrées par des programmeurs débutants.

Nous proposons maintenant d' étoffer le catalogue des tâches élémentaires pour le domaine des types construits en première : les tableaux, les tuples et les dictionnaires.

La figure 1 montre la partie du référentiel RCP concernant les tableaux, au programme de la classe de première en NSI.

Domaine	Évaluer	Modéliser	Anticiper	Décomposer	Généraliser	Abstraire
Construire un tableau et accéder à ses éléments	Évaluer le résultat de l'exécution d'un programme utilisant un tableau Quel est le type de l'expression suivante ? Quelle est la valeur de l'expression suivante ?	Modéliser une série d'informations par un tableau donné en extension À l'aide d'un tableau, choisir une ou des variables pour mémoriser des informations	Anticiper les traitements à programmer pour accéder aux éléments d'un tableau par leurs indices et les modifier Écrire une expression permettant d'accéder à l'information élémentaire... Écrire une instruction permettant de modifier un élément du tableau suivant...		Généraliser l'écriture d'un tableau en utilisant la compréhension Écrire une expression permettant de définir le tableau suivant par compréhension	Abstraire les données complexes en utilisant des tableaux Réécrire le programme suivant... en masquant les variables suivantes... et en les remplaçant par une ou des variables de type tableau
Parcourir un tableau pour rechercher ou calculer des informations	Évaluer le résultat de l'exécution d'un programme itérant sur un tableau Quelle est la valeur de la variable... à la fin de l'exécution du programme Python suivant ?	Modéliser un traitement de tableau en choisissant d'itérer sur les indices ou les éléments Choisir et nommer une variable pour réaliser le traitement suivant...	Anticiper un parcours linéaire de tableau Écrire un programme Python permettant de calculer / afficher ...		Généraliser un traitement en l'appliquant à tous les éléments d'un tableau Écrire une expression permettant de définir le tableau... en parcourant le tableau suivant ...	
Parcourir et modifier un tableau par des boucles imbriquées	Évaluer les valeurs successives des indices avec des boucles imbriquées Quelle est la valeur de la variable... à la fin de l'exécution du programme Python suivant ?	Modéliser une matrice ou une image par un tableau de tableaux Modéliser la situation suivante... à l'aide d'un tableau de tableaux	Anticiper un traitement de tableaux nécessitant plusieurs parcours imbriqués Écrire un programme Python permettant de... [nécessitant un parcours complexe]	Décomposer en identifiant des parcours séquentiels de tableaux Décomposer le problème suivant... [en identifiant un parcours séquentiel de tableaux]		Abstraire un parcours séquentiel en utilisant une fonction Réécrire le programme suivant... en encapsulant [le traitement...] dans une fonction à définir

FIGURE 1. Extrait du référentiel pour la classe de première

On appelle **compétence élémentaire**, une compétence à l'intersection d'une ligne et d'une colonne. Cela correspond à la compétence générique de la colonne, contextualisée par la ou les notions en jeu dans la ligne courante. A chaque compétence élémentaire, est associée, en dessous sur fond grisé, un **modèle de tâche élémentaire** permettant de tester en contexte cette compétence élémentaire.

Les modèles de tâches que nous avons identifiés sont à comparer aux types de tâches décrites dans le référentiel de Jolivet et Wang [5]. En particulier le type de tâche *analyser un programme* de Jolivet rassemble nos modèles de tâches construits pour les différentes compétences élémentaires de la compétence générique *évaluer*. Le type de tâche *concevoir un programme* de Jolivet correspond dans notre référentiel à plusieurs compétences distinctes et complémentaires :

modéliser, anticiper et décomposer. Il est notable de constater que nos deux approches concluent à la nécessité de diversifier les activités des élèves afin de leur permettre, en réalisant différents types de tâches, de développer différentes compétences.

Le référentiel PIAF [8] a, comme le nôtre, été conçu à partir des compétences de la pensée informatique, mais en le contextualisant pour la tranche d'âge 5/12 ans, ce qui explique qu'il n'aborde un programme que comme séquence d'action avec éventuellement des conditions et des boucles. Par ailleurs, le choix de séparer la compétence *évaluer* en deux parties *évaluer* et *prédire* et d'inclure la compétence *prédire* dans la compétence 1 : *Définir des abstractions / généraliser* nous semble confus. Il nous semble aussi indispensable pour un référentiel destiné au lycée de distinguer ce qui relève de l'abstraction (nommer des objets ou actions) et ce qui relève de la généralisation (définir des paramètres). Enfin les compétences élémentaires identifiées (par exemple C 6.3 *Corriger une séquence d'actions pour atteindre un objectif donné*) semblent être plus des « types de tâches » au sens de Jolivet [6] que des compétences au sens utilisé par Wing.

Notre problématique est maintenant d'étudier l'opérationnalité et la cohérence du référentiel de compétences et de son catalogue de tâches associées.

L'opérationnalité du référentiel peut être évaluée par la capacité des enseignants à l'utiliser pour concevoir des ressources alignées avec les programmes du lycée. On étudiera en particulier l'intérêt du référentiel pour les enseignants, pour construire des ressources, pour construire des évaluations et pour catégoriser des tâches complexes.

La cohérence du référentiel peut être estimée en recherchant d'éventuelles incohérences dans les résultats d'évaluation sur des tâches complexes et des tâches élémentaires. Par exemple, en supposant qu'une tâche complexe requiert trois compétences élémentaires, il y aurait incohérence si un nombre significatif d'élèves réussissent la tâche complexe sans réussir les tâches élémentaires associées aux compétences élémentaires requises.

Dans les sections suivantes, nous étudierons ces deux propriétés lors de la construction de ressources et d'évaluations et lors de la catégorisation de tâches complexes.

1 Un référentiel pour construire des ressources

Dans cette section, nous présentons l'utilisation du référentiel RCP et de son catalogue de tâches élémentaires associé, par une enseignante, pour construire des ressources. Nous décrivons ici la conception de deux ressources très différentes dans le cadre d'une situation d'apprentissage sur le thème des tableaux avec des élèves en première NSI. Dans un premier temps, nous présenterons les *questions flash*, de petites ressources proches des tâches élémentaires. Nous poursuivons avec des ressources plus conséquentes et destinées aux activités de *pair-programming*.

1.1 Questions flash

Les questions flash s'inscrivent dans une stratégie d'enseignement et permettent de mobiliser une connaissance, un savoir-faire, un traitement automatique ou réfléchi, dans l'objectif de tester, d'évaluer ou de consolider des notions ciblées.

Dans le référentiel ainsi construit (Figure 1), le contexte des tableaux est composé en trois notions, à savoir « *Construire un tableau et accéder à ses éléments* », puis « *Parcourir un tableau pour rechercher ou calculer des informations* » et enfin « *Parcourir et modifier un tableau par des boucles imbriquées* ». Pour réactiver et/ou évaluer de manière ponctuelle les connaissances des élèves sur ce sujet tout en poursuivant sa progression, l'enseignante a introduit des questions flash régulières au cours de ses séances. Ces questions flash ont été construites en sélectionnant parmi les modèles de tâches, celles mettant en jeux les compétences élémentaires visées sur les tableaux, et ont été proposées de façon régulière au groupe d'élèves.

Par exemple la question 1 ci-dessous est une instance de tâche élémentaire construite à partir du modèle de tâche pour la compétence élémentaire *Évaluer le résultat de l'exécution d'un programme utilisant un tableau*, alors que la question 2, est construite à partir du modèle de tâche pour la compétence élémentaire *Anticiper les traitements à programmer pour accéder aux éléments d'un tableau par leurs indices et les modifier*.

Question 1. Soit le tableau $T = [12, 4, 35, 7, 11]$.
Que renvoie $T[1]$? $T[5]$? $T[0]$? $T[-1]$?

Question 2. Soit le tableau $T = [12, 4, 35, 7, 11]$.
Écrire un code permettant d'accéder au dernier élément du tableau T ?

Concernant les compétences en programmation sur les parcours dans les tableaux, les deux questions flash suivantes ont conduit à identifier les difficultés d'une majorité d'élèves du groupe de première sur la compétence élémentaire *Anticiper un parcours linéaire de tableau*.

Question 3. Écrire un programme qui renvoie la somme des éléments du tableau $T=[15,20,7,12]$.

Question 4. Écrire un programme qui renvoie la moyenne des éléments du tableau $T=[15,20,7,12]$.

L'enseignante a trouvé simple la création de questions flash sur un thème bien précis du programme de première NSI en allant directement chercher dans le catalogue de tâches élémentaires celles aux bonnes intersections, ou en écrivant de nouvelles tâches analogues. L'apport semble être à double sens d'ailleurs puisque les nouvelles questions ainsi créées pourront venir alimenter le catalogue.

1.2 Pair-programming

Le pair-programming est une modalité d'enseignement de la programmation par la pratique où deux élèves doivent collaborer sur une même activité tout en ayant des rôles différents : le pilote programme pendant que le co-pilote analyse, commente, teste le travail de son binôme. Nous avons initié une première expérimentation concluante sur le pair programming [2], marquée par une participation active des élèves, des résultats encourageants ainsi qu'une identification des compétences *évaluer* et *anticiper* suivant le rôle endossé. En effet, le pilote développe des compétences d'anticipation alors que le co-pilote développe celles d'évaluation. L'échange des rôles institutionnalisé par le pair programming, amène les élèves à développer ces deux compétences génériques successivement.

Pour pallier aux difficultés rencontrées par les élèves sur le parcours dans un tableau (cf. questions 3 et 4), l'enseignante a proposé une séance de remédiation axée sur des compétences élémentaires spécifiques, grâce à des activités pratiques sur machine et en pair-programming. Pour cette séance, il a fallu prévoir un temps d'adaptation à l'outil par les élèves car l'outil utilisé était différent de celui utilisé lors de la découverte de cette méthode de programmation et nous prenons en compte dans cette analyse tous les éléments qui pourraient influencer cette expérimentation. Une fois l'outil pris en main et les notions de parcours remémorées, l'enseignante proposait aux binômes de retravailler cette notion de parcours à partir de questions construites sur le modèle de tâches associées à la compétence élémentaire *Anticiper un parcours linéaire d'un tableau* du domaine « *Parcourir un tableau pour rechercher ou calculer des informations* ». Voici un exemple de question conçue pour cette séance de remédiation :

Question 5. On considère dans cette partie le tableau `notes`, suivant, des élèves de première lors d'un devoir en NSI :

```
notes = [15, 13.5, 9.5, 12, 11, 18, 15, 12, 7, 10, 13, 12, 8.5,
         11, 8.5, 16, 12]
```

a. Pilote : Écrire la fonction `occurrences(v, T)` qui renvoie le nombre d'occurrences de la valeur `v` dans le tableau `T`.

Co-pilote : Combien d'élèves de cette classe ont eu 12 au devoir ? Testez la fonction `occurrences` avec le tableau `notes`.

Changement des rôles pilote et co-pilote

b. Pilote : Écrire la fonction `nb_plus_de_dix` qui prend en paramètre un tableau `T` et qui renvoie le nombre des valeurs de `T` supérieures ou égales à 10.

Co-pilote : Combien d'élèves de cette classe ont eu plus de 10 au devoir ? Testez la fonction `nb_plus_de_dix` avec le tableau `notes`.

Changement des rôles pilote et co-pilote

A la différence des questions 3 et 4, d'autres compétences élémentaires interviennent dans la question 5, autres que celles sur le parcours dans un tableau, telles que les notions d'accumulateur, de variables intermédiaires, de fonction à

un ou plusieurs paramètres qui ont été identifiées comme pré-requis à ces activités.

Cette séance de pair-programming, construite à partir du référentiel RCP, a permis aux élèves, par le changement de rôle régulier, de s'entraîner, de comprendre mais surtout de consolider la notion de parcours tout en évaluant ou écrivant le code demandé, permettant ainsi d'atteindre les objectifs de cette séance de travail en binôme.

1.3 Synthèse

A travers ces deux méthodes d'enseignement (les questions flash et le pair-programming), l'enseignante a pu s'appropriier le référentiel RCP pour concevoir des ressources variées, alignées sur le programme du lycée, offrant ainsi des expériences d'apprentissage diverses pour les élèves. Au regard de cette première expérimentation, lors de la construction de ressources, le référentiel et son catalogue se sont avérés opérationnels.

Pour confirmer l'opérationnalité du référentiel RCP, une perspective serait de proposer l'usage de ce référentiel et de son catalogue de tâches associé, à une cohorte d'enseignants et d'étudier ainsi l'appropriation de ce référentiel par des enseignants non impliqués dans la construction du référentiel.

2 Un référentiel pour construire une évaluation

Dans cette section nous présentons un autre usage du référentiel par l'enseignante pour la conception d'évaluations. Nous allons exposer la stratégie élaborée par l'enseignante visant à construire une évaluation. Nous distinguons quatre cas au regard des objectifs fixés par l'enseignante : (a) évaluer une compétence élémentaire, (b) évaluer des compétences dans un contexte ciblé, (c) évaluer une compétence générique et (d) évaluer des compétences dans plusieurs contextes identifiés.

- a. Si l'objectif consiste à vérifier ou à évaluer une compétence élémentaire en programmation telle que *Anticiper un parcours linéaire d'un tableau*, nous constatons que l'enseignante construit l'évaluation en choisissant, dans le catalogue des tâches, celles associées à la compétence élémentaire en privilégiant de diversifier les situations, par exemple, sur le choix du type de données (des tableaux d'entiers mais aussi des tableaux de chaînes de caractères) et sur le nombre de tableaux. Des tâches similaires aux questions 3 et 4 sont de bons candidats.

L'évaluation pourra ainsi se faire en mode débranché ou pas. De manière générale, on privilégie l'évaluation en mode débranché pour la compétence générique *évaluer* : en effet la tâche de l'élève étant d'évaluer mentalement, l'usage d'un interprète du langage biaiserait le résultat. Pour la compétence générique *anticiper*, on peut utiliser le mode branché, si on veut permettre à l'élève d'utiliser le retour instrumental de l'interprète Python, pour valider sa production.

- b. Si l'objectif consiste maintenant à vérifier ou à évaluer des compétences dans un contexte ciblé, comme « *Parcourir un tableau pour rechercher ou calculer des informations* », l'enseignante sélectionne des tâches associées aux compétences présentes dans ce contexte, afin de recouvrir l'ensemble des compétences élémentaires intervenant dans ce contexte.

Dans cette situation, on peut aussi utiliser un test en mode débranché pour tester l'ensemble des compétences. Ce test peut prendre la forme d'un QCM si on prend le soin de proposer des distracteurs pertinents pour la compétence *évaluer*.

- c. Si l'objectif consiste à vérifier ou évaluer une compétence générique, par exemple *modéliser*. L'enseignante propose de choisir des tâches associées aux compétences élémentaires dans lesquelles la compétence générique intervient, à savoir dans notre cas : *modéliser une série d'informations par un tableau donné en extension*, *modéliser un traitement de tableau en choisissant d'itérer sur les indices ou les éléments* et *modéliser une matrice ou une image par un tableau de tableaux*. Nous constatons qu'évaluer une compétence générique unique peut parfois être complexe, car les tâches sélectionnées, pour cette compétence générique, peuvent dissimuler d'autres compétences génériques. Cet usage peut être envisagé pour une évaluation diagnostique lorsque l'enseignant suspecte des difficultés spécifiques pour un élève par rapport à une compétence générique particulière (par exemple difficulté à généraliser). Comme dans le premier cas, si la compétence générique évaluée est *anticiper*, il est peut être préférable d'opter pour un test en mode branché, bien que cela ne soit pas impératif.

- d. Si l'objectif consiste à vérifier ou évaluer des notions dans plusieurs contextes ciblés, comme par exemple des tableaux, l'enseignante propose de sélectionner des tâches associées aux compétences élémentaires présentes dans ces différents contextes pour chacune des six compétences génériques. Cette situation correspond à la création d'une évaluation sommative portant sur plusieurs notions. Dans cette situation, afin de construire une évaluation comportant un nombre raisonnable de questions, il est nécessaire d'échantillonner en équilibrant à la fois le nombre de questions par notion et le nombre de questions par compétence générique, tout en donnant une couverture suffisante des différentes notions et compétences. Nous avons jusqu'à présent réalisé cette opération manuellement : un prolongement intéressant serait de proposer un algorithme pour ce travail.

Dans ce dernier cas, on préconise un test en mode débranché.

Synthèse

Les situations a) b) c) et d) correspondent au choix de s'intéresser à une ou plusieurs notions (lignes dans le référentiel) et à une ou plusieurs compétences génériques (colonnes dans le référentiel) pour construire une évaluation. Les situations a) et b) ont déjà été expérimentées avec une classe de première lors de séances d'apprentissage sur les tableaux. Les situations c) et d) seront

expérimentées avec la même classe en fin de progression sur ce thème. Notre conclusion provisoire est que le référentiel a pu guider notre travail d'élaboration d'évaluations, en assurant un bon alignement entre compétences travaillées et compétences évaluées.

Comme dans la précédente partie, nous proposons d'étendre l'usage de ce référentiel et de son catalogue de tâches associé, à une cohorte d'enseignants et d'étudier ainsi l'appropriation de ce référentiel par ces enseignants pour la construction d'évaluations. Cela nous permettra d'identifier les similitudes ou les différences avec l'enseignante dans la manière de concevoir des évaluations.

3 Un référentiel pour catégoriser les tâches complexes

Dans cette section, nous proposons une méthode pour catégoriser des tâches complexes. Nous illustrons la méthode sur deux tâches complexes choisies dans des manuels scolaires de NSI.

On identifie trois niveaux de catégorisation des tâches complexes. Pour le premier niveau, les utilisateurs du référentiel RCP commencent par faire une analyse des mots qui composent la consigne. Lorsque la consigne est explicite et emploie des mots similaires ou proches aux compétences génériques ou décrivant les notions dans laquelle elle intervient, une simple analyse lexicale suffit pour cibler les compétences élémentaires mises en jeu et permettant ainsi la catégorisation de la tâche complexe. Dans le deuxième niveau, lorsque l'analyse lexicale est insuffisante, on propose de catégoriser non pas à partir de l'énoncé mais à partir d'une solution attendue par l'enseignant, en examinant les compétences élémentaires sollicitées. Et enfin, lorsque l'on n'arrive toujours pas à catégoriser une tâche complexe, on propose d'itérer les deux précédentes étapes à plusieurs enseignants qui vont proposer leurs solutions possibles. On demande finalement aux enseignants de débattre de leurs catégorisations respectives en vue d'aboutir à un consensus.

Reprenons en détail cette stratégie sur les deux exemples sélectionnés :

Question 6. On souhaite gérer un emploi du temps hebdomadaire qui contient les horaires (début et fin) et matières de chaque créneau.

1. Proposer deux représentations différentes de l'emploi du temps avec des types construits.
2. Pour chaque représentation, écrire une fonction qui ajoute un créneau à l'emploi du temps et retourne **True** s'il est libre, ou sinon retourne **False** sans le modifier.

Question 7. Écrire un programme qui construit la matrice illustrée ci-dessous :

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{pmatrix}$$

La tâche complexe question 6 (manuel NSI 1ère, Hachette, p. 78) comprend deux tâches successives. Dans la première tâche, l'expression « *Proposer deux représentations différentes* » permet d'identifier la compétence élémentaire *modéliser*. La recherche de solutions va permettre de trouver les deux représentations : un tableau de tuples ou un tableau d'enregistrements à champs nommés. Cette première tâche sollicite donc les trois compétences « *Modéliser par des tuples* », « *Modéliser par des enregistrements à champs nommés* » et « *Modéliser par des tableaux* ». Dans la seconde tâche, l'expression « *écrire une fonction qui* » permet d'identifier la compétence *anticiper*. La recherche de solutions amène à décliner les notions en jeu : fonction, boucle bornée, tuples, enregistrements et tableaux, ce qui permet d'énumérer les compétences sollicitées, au croisement de ces notions et de la compétence générique *anticiper*.

Dans la tâche complexe question 7 (manuel NSI 1ère, Hachette, p. 97), l'analyse lexicale de la consigne permet de cibler le domaine « *construire un tableau* ». L'expression « *écrire un programme qui construit* » laisse présager la compétence *anticiper*. Cependant la solution proposée par les enseignants, ne mobilise que la compétence « *modéliser une série d'informations par un tableau donné en extension* ». La solution donnée dans le manuel utilise une compréhension imbriquée, ce qui amènerait à la catégoriser dans « *généraliser l'écriture d'un tableau en utilisant la compréhension* ». L'imprécision de la consigne ne permet pas de décider de la ou des compétences devant être impliquées dans la réalisation de la tâche.

Analyse de la cohérence du référentiel

C'est suite aux premières catégorisations d'activités complexes effectuées que l'on s'est posé la question de la cohérence du référentiel. En trouvant des incohérences entre les résultats d'évaluation sur des tâches complexes et sur des tâches élémentaires, on a pu s'apercevoir que des tâches que l'on avait considérées comme élémentaires ne pouvaient pas être considérées comme telles. On avait en particulier considéré comme élémentaire la tâche « *anticiper l'écriture d'une répétition* ». Suite à l'incohérence entre la réussite d'une tâche complexe comportant entre autres une boucle bornée et l'échec d'une tâche élémentaire ne comportant qu'une boucle non bornée, il a été nécessaire de scinder cette compétence en deux : « *anticiper l'écriture d'une boucle bornée* » et « *anticiper l'écriture d'une boucle non bornée* ». On s'interroge actuellement sur le caractère élémentaire des compétences de la ligne « *Construire un tableau et accéder à ses éléments* ». Les résultats des évaluations à venir pourraient nous inciter à scinder chacune de ces compétences.

La consolidation du référentiel en vue de valider son opérationnalité et sa cohérence est donc encore un travail en cours. Nous avons cependant décidé de déjà rendre public le référentiel, car ce sont précisément les expérimentations à plus grande échelle qui nous permettront de conclure.

Le référentiel RCP est disponible en ligne à l'adresse :

<https://iremi974.gitlab.io/rcp/>

Conclusions et perspectives

Le référentiel RCP, dont les origines et les principes de construction avaient déjà été présentés précédemment, a fait l'objet dans ce travail d'une première expérimentation destinée à évaluer son opérationnalité. La construction de plusieurs sortes de ressources d'enseignement, l'élaboration d'évaluations et l'usage du référentiel pour catégoriser des tâches complexes semblent indiquer qu'il peut constituer un cadre favorisant le travail de l'enseignant désirant adopter une approche par compétences. La validation de cette opérationnalité ne peut cependant reposer sur l'expérimentation avec une seule enseignante, de surcroît quand cette enseignante a participé à la conception du référentiel. Les prolongements immédiats de ce travail vont donc consister à proposer au niveau académique une formation d'un groupe d'enseignants NSI à l'approche par compétences avec ce référentiel, puis à les impliquer dans une démarche collaborative de création de ressources et d'évaluations. Le corpus de ressources ainsi recueilli devrait permettre de répondre à la question de la possibilité d'appropriation de RCP par des enseignants hors du groupe de recherche qui l'a conçu, ainsi qu'à la question de son opérationnalité.

La question de la cohérence et de la nature élémentaire des compétences proposées nécessite aussi de mener des expérimentations de plus grande ampleur. L'évaluation de plusieurs classes d'élèves ayant passé les mêmes évaluations constituées exclusivement de tâches élémentaires, comparées à leurs résultats sur des tâches complexes devrait nous permettre de continuer à consolider le référentiel en distinguant si besoin des compétences que l'on considère comme élémentaires.

D'un point de vue plus théorique, l'articulation entre modèles de tâches et compétences, nous semble une réflexion fondamentale à engager pour mieux articuler notre travail avec d'autres travaux en cours dont ceux de Jolivet, et pourrait nous amener à étudier les usages et la portée de ces travaux y compris hors du champ dans lequel ils ont pris leur source. Nous pensons en particulier aux EIAH alors que nos préoccupations initiales ne portaient que sur l'accompagnement du travail de l'enseignant.

Remerciements

Nous remercions les relecteurs anonymes pour leurs remarques très pertinentes qui nous ont permis d'améliorer la rédaction de cette communication.

Références

1. Baron, G.L. : La constitution de l'informatique comme discipline scolaire, le cas des lycées. Theses, Université René Descartes - Paris V (Oct 1987), <https://tel.archives-ouvertes.fr/edutice-00000369>
2. Chane-Lune, S., Declercq, C., Hoarau, S. : Expérimentation du pair-programming en spécialité NSI en classe de première pour l'acquisition de compétences en programmation. In : Broisin, J., Declercq, C., Fluckiger, C., Parmentier, Y., Peter, Y., Secq, Y. (eds.) Atelier " Apprendre la Pensée Informatique de la Maternelle à l'Université ", dans le cadre de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH). pp. 25–32. Brest, France (2023), <https://hal.science/hal-04144211>
3. Chane-Lune, S., Declercq, C., Hoarau, S. : Un référentiel de compétences en programmation pour identifier les difficultés des débutants et différencier les activités. In : Mens, K., Goletti, O. (eds.) Colloque Didapro 10 sur la Didactique de l'informatique et des STIC. pp. 53–63. Louvain-La-Neuve, Belgium (2024), <https://hal.science/hal-04482126>
4. Declercq, C. : Didactique de l'informatique : une formation nécessaire. Sticf **28**(3) (Apr 2022), <http://sticf.org/num/vol2021/28.3.8.declercq/28.3.8.declercq.htm>
5. Jolivet, S., Dechaux, E., Gobard, A.C., Wang, P. : Description et analyse de trois EIAH d'apprentissage de Python. In : Broisin, J., Declercq, C., Fluckiger, C., Parmentier, Y., Peter, Y., Secq, Y. (eds.) Atelier " Apprendre la Pensée Informatique de la Maternelle à l'Université ", dans le cadre de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH). pp. 9–16. Brest, France (2023), <https://hal.science/hal-04144212>
6. Jolivet, S., Dechaux, E., Gobard, A.C., Wang, P. : Vers l'analyse de ressources d'apprentissage de la programmation à l'aide d'un référentiel de types de tâches. In : Mens, K., Goletti, O. (eds.) Colloque Didapro 10 sur la Didactique de l'informatique et des STIC. pp. 87–98. Louvain-La-Neuve, Belgium (2024), <https://hal.science/hal-04482123>
7. Ministère de l'Éducation nationale de la Jeunesse et des Sports : Programmes et ressources en numérique et sciences informatiques - voie G (2019), <https://eduscol.education.fr/2068/programmes-et-ressources-en-numerique-et-sciences-informatiques-voie-g>
8. Parmentier, Y., Reuter, R., Higuette, S., Kataja, L., Kreis, Y., Dufflot-Kremer, M., Laduron, C., Meyers, C., Busana, G., Weinberger, A., Denis, B. : PIAF : Developing Computational and Algorithmic Thinking in Fundamental Education. In : AACE 2020 - EdMedia + Innovate Learning. Proceedings of EdMedia + Innovate Learning 2020 Online, vol. 1, pp. 315–322. Association for the Advancement of Computing in Education (AACE), Waynesville, NC, Amsterdam / Virtual, Netherlands (Jun 2020), <https://hal.science/hal-02888504>
9. Selby, C., Woollard, J. : Computational thinking : the developing definition. In : SIGCSE (2014)
10. Wing, J.M. : Computational Thinking. Communications of the ACM **49** (2006)

Domaine	Évaluer	Modéliser	Anticiper	Décomposer	Généraliser	Abstraire
Programmer avec des expressions	Évaluer la valeur et le type (entier, chaîne de caractères ou booléen) d'une expression comportant variables, constantes et opérateurs Quelle est le type et la valeur de l'expression suivante ?	Modéliser les informations disponibles à calculer par des variables de type élémentaire (entier, chaîne de caractères et booléen) Choisir une ou des variables pour représenter ...	Anticiper l'écriture d'une expression comportant variables, constantes et opérateurs Écrire une expression dépendant de ... permettant de calculer ...	Décomposer une expression complexe en identifiant une ou des variables intermédiaires Réécrire l'expression suivante en nommant ... la partie entre parenthèses	Généraliser une expression en remplaçant une constante par une variable Réécrire l'expression suivante en remplaçant la valeur ... par une notation plus générale	
Programmer avec des affectations et des séquences	Évaluer les valeurs des variables, à l'exécution d'un programme comportant une séquence d'affectations Quelle est la valeur de la variable ... à la fin de l'exécution du programme Python suivant ?	Modéliser les informations à calculer par une ou des variables intermédiaires Choisir une ou des variables pour mémoriser les résultats intermédiaires utiles pour calculer ...	Anticiper l'écriture d'un traitement séquentiel pour obtenir un résultat spécifique, la méthode étant donnée Écrire un programme Python permettant de calculer ... / Écrire un programme Python permettant d'afficher ...	Décomposer un traitement complexe en identifiant une ou des variables intermédiaires Décomposer le problème suivant en utilisant des variables intermédiaires		
Programmer avec des boucles bornées	Évaluer les valeurs des variables, à l'exécution d'un programme comportant une boucle bornée Quels sont les messages affichés à l'exécution du programme Python suivant ? / Quelle est la valeur de la variable ... à la fin de l'exécution du programme Python suivant ?	Modéliser par une ou des variables les informations à modifier à chaque répétition Choisir une ou des variables à modifier directement pour calculer ...	Anticiper l'écriture d'une boucle bornée pour obtenir un résultat spécifique, la méthode étant donnée Écrire un programme Python permettant de calculer ... / Écrire un programme Python permettant d'afficher ...	Décomposer un traitement complexe en identifiant une boucle bornée Décomposer le problème suivant en identifiant un traitement pouvant être répété plusieurs fois	Généraliser un traitement séquentiel en le remplaçant par un traitement répété Modifier le programme Python suivant en utilisant une répétition, tout en conservant les mêmes affichages à l'exécution / le même résultat	
Programmer avec des boucles non bornées	Évaluer les valeurs des variables, à l'exécution d'un programme comportant une boucle non bornée Quels sont les messages affichés à l'exécution du programme Python suivant ? / Quelle est la valeur de la variable ... à la fin de l'exécution du programme Python suivant ?	Modéliser par une ou des variables les informations à modifier à chaque répétition Choisir une ou des variables à modifier directement pour calculer ...	Anticiper l'écriture d'un traitement répété pour obtenir un résultat spécifique, la méthode étant donnée Écrire un programme Python permettant de calculer ... / Écrire un programme Python permettant d'afficher ...	Décomposer un traitement complexe en identifiant une boucle non bornée Décomposer le problème suivant en identifiant un traitement pouvant être répété		
Programmer avec des constructions alternatives ou conditionnelles	Évaluer les valeurs des variables, à l'exécution d'un programme comportant une ou des constructions alternatives ou conditionnelles Quels sont les messages affichés à l'exécution du programme Python suivant ? / Quelle est la valeur de la variable ... à la fin de l'exécution du programme Python suivant ?	Modéliser par une variable booléenne la condition d'une alternative ou d'une conditionnelle Choisir et nommer une variable pour mémoriser le résultat de la condition ... dans le problème suivant ...	Anticiper l'écriture d'un traitement alternatif ou conditionnel pour obtenir un résultat spécifique, la méthode étant donnée Écrire un programme Python permettant de calculer ... / Écrire un programme Python permettant d'afficher ...	Décomposer un traitement complexe en plusieurs cas en utilisant des constructions alternatives ou conditionnelles Décomposer le problème suivant en distinguant les cas admettant un traitement particulier		
Programmer avec des fonctions	Évaluer le résultat de l'appel d'une fonction avec des paramètres donnés Quelle est la valeur à l'évaluation de l'expression ... avec la définition de fonction Python suivante ?	Modéliser un traitement par une fonction en spécifiant ses paramètres Spécifier le nom, les paramètres d'entrée et le type de résultat à renvoyer pour une fonction destinée à calculer ...	Anticiper l'écriture du corps d'une fonction Écrire une fonction Python ... qui étant donnée ... renvoie ...	Décomposer un traitement complexe en utilisant la composition de fonctions Décomposer le problème suivant en identifiant deux parties pouvant chacune être programmées par une fonction	Généraliser un traitement, par la définition d'une fonction avec paramètres ou par l'appel d'un paramètre à une fonction Dans le programme Python suivant, remplacer les restrictions écrites plusieurs fois, en dérivant et utilisant une fonction	Abstraire un traitement par une définition de fonction avec paramètres et commentaire Décrire le traitement effectué par la fonction suivante en faisant abstraction de la méthode utilisée
Programmer avec des fonctions récursives	Évaluer le résultat de l'appel d'une fonction récursive Quel est le cas de base de cette fonction ? Quelle est la valeur à l'évaluation de l'expression ... avec la définition de fonction récursive suivante ? Tracer les appels à la fonction suivante ... à l'évaluation de ... Justifier que la fonction ... est récursive.	Modéliser un traitement par une fonction récursive en spécifiant ses paramètres Spécifier le nom, les paramètres d'entrée et le type de résultat à renvoyer pour une fonction destinée à calculer ... Proposer une trace pour résoudre la résolution récursive du problème suivant	Anticiper l'écriture d'une fonction récursive Écrire une fonction récursive ... qui étant donnée ... renvoie ... Écrire une fonction récursive à partir de la trace suivante		Généraliser une fonction récursive avec une fonction en paramètre Généraliser les fonctions suivantes en une seule, pour pouvoir choisir à l'appel de la fonction ...	
Programmer avec des tuples	Évaluer une expression permettant d'accéder aux informations d'un tuple Quel est le type de l'expression suivante ? Quelle est la valeur de l'expression suivante ?	Modéliser une information complexe par un tuple d'informations élémentaires À l'aide de tuple, choisir une ou des variables pour mémoriser des informations	Anticiper l'accès aux informations élémentaires d'un tuple Écrire une expression permettant d'accéder à l'information élémentaire ...			Abstraire des données complexes en utilisant des tuples Réécrire le programme suivant ... en remplaçant les variables suivantes ... et en les remplaçant par une ou des variables de type tuple
Programmer avec des enregistrements à champs nommés	Évaluer une expression permettant d'accéder aux informations d'un enregistrement à champs nommés Quel est le type de l'expression suivante ? Quelle est la valeur de l'expression suivante ?	Modéliser une information complexe par un enregistrement à champs nommés À l'aide d'un enregistrement à champs nommés, choisir une ou des variables pour mémoriser des informations	Anticiper l'accès aux informations élémentaires d'un enregistrement à champs nommés Écrire une expression permettant d'accéder à la valeur ...			Abstraire des données complexes en utilisant des enregistrements à champs nommés Réécrire le programme suivant ... en remplaçant les variables suivantes ... et en les remplaçant par une ou des variables de type enregistrement à champs nommés
Construire un tableau et accéder à ses éléments	Évaluer le résultat de l'exécution d'un programme utilisant un tableau Quel est le type de l'expression suivante ? Quelle est la valeur de l'expression suivante ?	Modéliser une série d'informations par un tableau donné en extension À l'aide d'un tableau, choisir une ou des variables pour mémoriser des informations	Anticiper les traitements à programmer pour accéder aux éléments d'un tableau par leurs indices et les modifier Écrire une expression permettant d'accéder à l'information élémentaire ... Écrire une instruction permettant de modifier un élément du tableau suivant.		Généraliser l'écriture d'un tableau en utilisant la compréhension Écrire une expression permettant de définir le tableau suivant par compréhension	Abstraire les données complexes en utilisant des tableaux Réécrire le programme suivant ... en remplaçant les variables suivantes ... et en les remplaçant par une ou des variables de type tableau
Parcourir un tableau pour rechercher ou calculer des informations	Évaluer le résultat de l'exécution d'un programme lissant sur un tableau Quelle est la valeur de la variable ... à la fin de l'exécution du programme Python suivant ?	Modéliser un traitement de tableau en choisissant d'itérer sur les indices ou les éléments Choisir et nommer une variable pour calculer le traitement suivant ...	Anticiper un parcours linéaire de tableau Écrire un programme Python permettant de calculer / afficher ...		Généraliser un traitement en itérant sur tous les éléments d'un tableau Écrire une expression permettant de définir le tableau ... en parcourant le tableau suivant ...	
Parcourir et modifier un tableau par des boucles imbriquées	Évaluer les valeurs successives des indices avec des boucles imbriquées Quelle est la valeur de la variable ... à la fin de l'exécution du programme Python suivant ?	Modéliser une matrice ou une image par un tableau de tableaux imbriqués Modéliser la situation suivante ... à l'aide d'un tableau de tableaux	Anticiper un traitement de tableau nécessitant plusieurs parcours imbriqués Écrire un programme Python permettant de ... (représentant un parcours complexe)	Décomposer en identifiant des parcours séquentiels de tableaux imbriqués Décomposer le problème suivant ... (en identifiant un parcours séquentiel de tableaux)		Abstraire un parcours séquentiel en utilisant une fonction Réécrire le programme suivant ... en encapsulant (le traitement ...) dans une fonction à définir
Construire un dictionnaire et accéder à ses informations	Évaluer le résultat de l'exécution d'un programme utilisant un dictionnaire Quelle est la valeur à l'évaluation de l'expression suivante ?	Modéliser l'association d'informations par un dictionnaire Choisir une ou des variables pour mémoriser les informations suivantes ... (à représenter par un dictionnaire)	Anticiper le traitement permettant d'ajouter ou modifier une association à un dictionnaire Écrire une expression permettant d'accéder à ... / Écrire une instruction permettant de modifier ... dans le dictionnaire suivant ...		Généraliser l'écriture d'un dictionnaire en utilisant la compréhension Écrire une expression permettant de définir le dictionnaire suivant par compréhension	Abstraire les données complexes en utilisant des dictionnaires Réécrire le programme suivant en remplaçant ... par l'utilisation d'un dictionnaire
Itérer sur les éléments d'un dictionnaire	Évaluer le résultat d'un programme lissant sur un dictionnaire Quelle est la valeur de la variable ... à la fin de l'exécution du programme Python suivant ?	Modéliser un traitement de dictionnaire en choisissant d'itérer sur les clés, les valeurs ou les paires Choisir les indices pour parcourir le dictionnaire ... pour ...	Anticiper un traitement nécessitant d'itérer sur un dictionnaire Écrire un programme Python permettant de ...			
Spécifier et implémenter des structures de données		Modéliser les données et les traitements en spécifiant une structure de données par son interface Spécifier la structure de données ... par une interface possible	Anticiper l'implémentation d'une structure de données Implémenter la structure de données ...			Abstraire une structure de données en spécifiant une interface permettant plusieurs implémentations Spécifier la structure de données ... par son interface, en faisant abstraction de l'implémentation
Programmer avec des objets et des classes	Évaluer la création d'une instance, l'invocation de méthodes et l'accès aux attributs d'un objet Quelle est la valeur de l'objet ... à la fin de l'exécution du programme Python suivant ?	Modéliser les données et les traitements en spécifiant une classe, ses attributs et des méthodes Spécifier une classe, des attributs et des méthodes pour traiter le problème suivant ...	Anticiper des traitements en implémentant les méthodes d'une classe Implémenter les méthodes de la classe ... spécifier ainsi ...	Décomposer les données et les traitements en les regroupant dans des classes Proposer une décomposition en classes pour regrouper les objets à programmer pour le problème suivant ...	Généraliser des traitements et des données en écrivant la définition d'une classe Modifier le programme suivant pour rassembler dans une classe toutes les fonctions qui manipulent les mêmes objets	Abstraire un ensemble de données et de traitements en le remplaçant dans une classe en tant qu'attributs et méthodes Encapsuler tous les traitements manipulant ... dans une classe à extraire du programme suivant
Programmer avec des structures linéaires : listes, piles, files	Évaluer le résultat de l'appel de fonctions ou méthodes sur des structures de données linéaires Quelle est la valeur de la variable ... à la fin de l'exécution du programme Python suivant ?	Modéliser une situation en choisissant la structure de données linéaire adaptée Choisir la structure de données linéaire adaptée au problème suivant ...	Anticiper l'écriture des fonctions ou méthodes des structures de données linéaires Implémenter les fonctions ... pour la structure de donnée ...			Abstraire une structure de données linéaire en masquant son implémentation Encapsuler tous les traitements manipulant ... dans une classe à extraire du programme suivant
Programmer avec des arbres	Évaluer la taille ou la hauteur d'un arbre et des requêtes simples (large, deux-à-deux, gauche) Quelle est la valeur de l'expression suivante ? de la hauteur de l'arbre suivant ?	Modéliser une information hiérarchique par un arbre Représenter les informations suivantes par un arbre ...	Anticiper l'implémentation d'un arbre et son parcours en largeur ou en profondeur (infixe, préfixe ou postfixe) Écrire un programme Python permettant de représenter ... et de parcourir ...			Abstraire la structure de données arbres par une classe Encapsuler tous les traitements manipulant un arbre ... dans une classe à extraire du programme suivant
Programmer avec des graphes	Évaluer les distances, le rayon, le diamètre et le centre d'un graphe Quelle est la valeur de la distance / rayon ... du graphe suivant ?	Modéliser des informations relationnelles par un graphe avec une matrice d'adjacence ou une liste de successeurs Représenter les informations suivantes par un graphe ...	Anticiper l'implémentation d'un graphe et l'écriture d'algorithme de parcours ou de recherche Écrire un programme Python permettant de représenter ... et de parcourir ...			Abstraire la structure de données graphe par une classe Encapsuler tous les traitements manipulant un graphe ... dans une classe à extraire du programme suivant

FIGURE 2. Référentiel RCP