



HAL
open science

Concours Castor, ressource pédagogique pour l'enseignement de l'informatique? Étude exploratoire auprès d'enseignants

Béatrice Drot-Delange, Françoise Tort

► To cite this version:

Béatrice Drot-Delange, Françoise Tort. Concours Castor, ressource pédagogique pour l'enseignement de l'informatique? Étude exploratoire auprès d'enseignants. Parriaux Gabriel; Pellet Jean-Philippe; Baron Georges-Louis; Bruillard Eric; Komis Vassilis. De 0 à 1 ou l'heure de l'informatique à l'école. Actes du colloque Didapro 7 – DidaSTIC, Peter Lang, pp.199-218, 2018, 978-3-0343-3324-5. hal-01753096

HAL Id: hal-01753096

<https://hal.science/hal-01753096>

Submitted on 29 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BÉATRICE DROT-DELANGE^a & FRANÇOISE TORT^b

a. Université Clermont Auvergne, ACTé
beatrice.drot-delage@uca.fr

b. ENS Paris-Saclay, STEF
francoise.tort@ens-paris-saclay.fr

Concours Castor, ressource pédagogique pour l'enseignement de l'informatique ? Étude exploratoire auprès d'enseignants¹

Résumé

La question que nous posons dans cette communication est celle de savoir dans quelle mesure les défis du concours Castor peuvent constituer une ressource pour les professeurs dans le cadre de l'enseignement de l'informatique. Pour apporter des éléments de réponse exploratoires à ce questionnement, nous croisons l'analyse faite par des enseignants d'informatique et sciences du numérique (ISN) de défis du concours Castor avec les commentaires des concepteurs concernant les savoirs informatiques mobilisés par ces défis.

Les résultats obtenus montrent que les seuls énoncés des défis ne permettent pas toujours aux enseignants, malgré une expérience importante dans l'enseignement de l'informatique, de cerner quels sont les champs de l'informatique mobilisés ou potentiellement mobilisables dans les défis. Plus précisément, nous identifions des défis « ambitieux », « opaques » ou « ambigus » selon l'écart entre les analyses des enseignants et les intentions des concepteurs. Ceci pose question d'une part sur la mise en valeur du potentiel pédagogique de chaque défi et d'autre part sur la rédaction des solutions par les concepteurs et leur valorisation. L'article se conclut par des pistes de recherche.

Mots clés : ressources pédagogiques, enseignement de l'informatique, discipline scolaire, domaines de l'informatique

1 Cette recherche bénéficie d'un financement dans le cadre du projet ANR ReVEA.

1 Introduction

En France, l'informatique est désormais présente dans les programmes de l'enseignement primaire et du collège depuis la rentrée 2016 et au lycée pour certaines filières depuis 2012. Cette introduction s'est faite dans un contexte où peu de ressources existent, où les enseignants sont peu formés à l'informatique et à son enseignement, où les accompagnateurs des enseignants du premier degré sont sensibles à l'importance de l'enseignement de l'informatique sous réserve qu'ils aient eu dans le passé une expérience de cet enseignement (Villemonteix, 2017).

Parallèlement, depuis 2011, le concours Castor² informatique France connaît un succès qui ne se dément pas, d'année en année. Les défis du concours mettent en jeu des concepts, méthodes et principes propres à la science informatique, tout en étant accessibles à des élèves qui n'ont pas de prérequis dans ce domaine (Tort & Dagiene, 2012). Le concours se déroule sur le temps scolaire et à l'initiative des enseignants qui inscrivent leurs classes. Toutefois les défis ne sont pas conçus pour mesurer des savoirs ou des compétences acquis et ne correspondent à aucune forme canonique scolaire. La manière de résoudre le problème n'est pas explicitée et doit être découverte et construite par le candidat (Tort, Drot-Delange & Mano, 2017). Les défis se présentent sous la forme de courts énoncés de problèmes à résoudre, comprenant souvent des images ou des diagrammes, et des artefacts interactifs plus ou moins complexes (Tort & Drot-Delange, 2015).

La question que nous posons dans cette communication est celle de savoir dans quelle mesure les défis du concours Castor peuvent constituer une ressource pédagogique pour les professeurs dans le cadre de leur enseignement de l'informatique.

2 Problématique

Les connaissances professionnelles des enseignants relèvent selon Shulman (2007) de trois types : des connaissances disciplinaires, des connaissances

2 <<http://castor-informatique.fr>>

pédagogiques et des connaissances curriculaires des contenus à enseigner. Dans cet article, nous nous intéresserons plus particulièrement à la question des connaissances disciplinaires. Shulman (2007) considère que ce type de connaissances renvoie « à la quantité et à l'organisation de la connaissance *per se* dans l'esprit du professeur » (p. 104). Une des difficultés de l'informatique est qu'il n'existe pas d'ontologie de cette science (Bruillard, 2016), malgré les nombreuses propositions de classification qui sont proposées régulièrement (Denning & Martell, 2015). De plus, l'informatique n'est présente dans l'enseignement obligatoire secondaire que depuis 2012, sous forme d'enseignements optionnels au lycée, et comme chapitre aux programmes d'autres disciplines (mathématiques et technologie) au collège. Les enseignants prenant en charge ces enseignements n'ont pas (ou rarement) de formation initiale dans ce domaine (Baron *et al.*, 2015). On comprend dès lors que l'élaboration de connaissances disciplinaires en informatique telles que les définit Shulman, est une tâche complexe pour les enseignants d'informatique.

Les concepteurs des défis du Concours Castor sont des acteurs ayant une expertise forte dans le domaine de la science informatique. En France, il s'agit de membres de l'association France-IOI³ et de l'INRIA (Institut national de recherche en informatique et en automatique). Les organisateurs cherchent à couvrir l'ensemble des champs de la science informatique et, désormais, de la pensée computationnelle (Dagiene, Sentance & Stupuriene, 2017) et élaborent des classifications permettant aux concepteurs de situer chaque défi dans un domaine de l'informatique.

Dagiene, Sentance et Stupuriene proposent la classification suivante. Elle est construite selon deux dimensions, la première est celle des connaissances, la seconde celle des habiletés (*skill*). Les connaissances relèvent de 5 domaines : « algorithmes et programmation », « données, structures de données et représentations », « processus informatiques et matériels », « communication et réseau », « interactions, systèmes et société ». Ces domaines sont précisés par des mots clés permettant de mieux cerner leur contenu. Les habiletés sont construites sur les compétences considérées comme relevant de la pensée informatique, et identifiées notamment par Selby et Woollard (2013), à savoir l'abstraction, la pensée algorithmique, la décomposition, l'évaluation et la généralisation.

Une organisation possible des connaissances en informatique est donc proposée par les initiateurs du concours eux-mêmes. Nous posons la

3 Fondée en juin 2004 dans le but de développer la sélection et l'entraînement de l'équipe de France aux Olympiades Internationales d'Informatique. <<http://www.france-ioi.org>>

question de savoir si cette organisation des connaissances est perceptible par des enseignants qui ont d'une part acquis une certaine familiarité avec le concours, puisqu'ils inscrivent régulièrement leurs élèves, et d'autre part qui enseignent l'informatique en lycée.

Nous faisons l'hypothèse que les défis du concours Castor constituent des ressources pour les enseignants, à condition que ces défis mettent en œuvre des savoirs informatiques connus et reconnus par les enseignants. Toutefois, si l'identification des connaissances et compétences couvertes par une ressource, ici un exercice, nous semble une condition nécessaire à ce qu'un enseignant décide de l'exploiter dans son enseignement, elle n'est certainement pas suffisante. L'adéquation de l'exercice pour traiter du concept identifié est aussi un facteur de choix. Sur ce point, les défis du castor introduisent une forme que l'on peut qualifier d'innovante. Les études menées n'ont pas encore véritablement évalué leur pertinence sur la compréhension, et l'apprentissage par les élèves. Il s'agit donc de se demander dans quelle mesure les enseignants repèrent les notions et savoirs du domaine informatique présents dans ces défis et si cette perception correspond aux intentions des concepteurs. Autrement dit, les intentions des concepteurs en termes de savoirs informatiques mis en jeu dans les défis sont-elles suffisamment lisibles dans les énoncés du concours Castor pour être exploitées par les enseignants ?

Pour apporter des éléments de réponse exploratoires à notre questionnement, nous proposons de croiser l'analyse faite par des enseignants des défis du concours Castor avec celle des concepteurs, du point de vue des connaissances disciplinaires. Nous nous intéressons à la perception par les enseignants du contenu informatique des défis, que nous confrontons aux intentions des concepteurs.

3 Méthodologie

Nous avons demandé à des enseignants de faire une analyse d'un corpus de 19 défis retenus par nous. Nous présentons d'abord la manière dont les données ont été collectées puis la méthode d'analyse mise en œuvre.

3.1 Présentation des corpus de données

Plusieurs corpus de données ont été constitués et étudiés. Le premier corpus comporte les énoncés d'une sélection de défis des éditions passées du concours (corpus 1). Le second corpus correspond aux analyses de ces défis effectuées et rédigées par les enseignants (corpus 2). Le troisième corpus est constitué des textes des solutions des défis rédigées par les concepteurs et publiées après le concours (corpus 3).

Tableau 1 : Liste des défis retenus pour l'expérimentation⁴.

N ^{os}	Année du concours	Pays concepteur	Niveau de classe	Titre du défi
1	2012	Russie	4 ^e -3 ^e	Affichage digital
2	2011	Slovaquie	4 ^e -3 ^e	Les amis de Lucie
3	2012	France	4 ^e -3 ^e	Anonymisation
4	2013	France	6 ^e -5 ^e	Arroser la plante
5	2012	Tchécoslovaquie	6 ^e -5 ^e	Bébrocarina
6	2013	France	6 ^e -5 ^e	Chercher-remplacer
7	2013	France	6 ^e -5 ^e	Choix des invités
8	2012	France	6 ^e -5 ^e	Course de grenouilles
9	2012	France	6 ^e -5 ^e	Chez le dentiste
10	2012	France	6 ^e -5 ^e	Dessine un castor
11	2013	Slovaquie	6 ^e -5 ^e	Dessiner un dé
12	2012	Tchécoslovaquie	6 ^e -5 ^e	Labyrinthe
13	2013	Suède	4 ^e -3 ^e	Réseau pas cher
14	2013	France	6 ^e -5 ^e	Robot peintre
15	2013	Slovénie	2 ^e	Rumeurs
16	2012	France	2 ^e	Territoire de Castor
17	2012	Lettonie	1 ^e -Tale	Transport de troncs
18	2013	France	6 ^e -5 ^e	Trier par poids
19	2012	France	2 ^e	Trombone rouge

Corpus 1 : énoncés de défis sélectionnés

Les défis retenus proviennent des concours des années 2011 à 2013 (cf. Tableau 1). Ils ont été choisis parmi ceux qui semblaient poser par leur

4 Les énoncés, solutions et explications des défis sont consultables en ligne : <<http://castor-informatique.fr/questions/?menu=1>>.

5 Numérotation introduite pour la présente recherche, afin de faire plus facilement référence aux défis dans la suite du texte.

contenu et leur forme des questions de translittération, notamment du fait des interfaces et images utilisées. En effet, cette recherche a été initiée dans le projet ANR Translit, qui avait pour objet la translittération, et se poursuit au sein du projet ANR REVEA concernant les ressources vivantes en éducation et en formation. Notons que la sélection effectuée reste représentative des défis proposés ces années-là.

Corpus 2 : analyses des défis par les enseignants

Nous avons demandé à des enseignants impliqués dans ce projet de recherche (*cf.* Tableau 2) de procéder à une analyse des défis retenus. Ces professeurs ont pour point commun d'enseigner l'informatique dans le cadre de l'enseignement de spécialité ISN (Informatique et Sciences du Numérique) en classe de terminale, mais aussi comme enseignement optionnel en classe de seconde ou de première (Informatique et Création Numérique).

Leur analyse a été recueillie au moyen d'un document fournissant une grille qu'ils devaient compléter. La grille comportait le nom du défi, l'année du concours et le niveau de classe minimal auquel le défi a été proposé. Les enseignants ont complété quatre rubriques : (1) lister les notions ou les concepts informatiques mobilisés par le défi (ils pouvaient préciser s'il n'y avait pas de notions, selon eux, ou alors s'il s'agissait plutôt de notions mathématiques) ; (2) décrire la « meilleure » stratégie de résolution possible, et mentionner s'il existait selon eux d'autres stratégies de résolution ; (3) préciser si les élèves étaient susceptibles d'avoir déjà rencontré ces notions dans leur scolarité et, si oui, dans quelle matière ou dans quel cours ; (4) préciser l'intérêt de ces défis du point de vue de l'apprentissage de l'informatique, et la difficulté estimée selon eux.

Tableau 2 : Enseignants participant à la recherche.

Enseignant ⁶	Discipline	Ancienneté dans l'enseignement	Enseigne ISN depuis	Statut
Christophe	Mathématiques ⁷	33 ans	2012	Agrégé
François	Mathématiques	18 ans	2012	Agrégé
Isabelle	Mathématiques	11 ans	2012	Agrégée

6 Les prénoms ont été modifiés, le genre conservé.

7 En France, les enseignants en charge des enseignements d'informatique appartiennent à une autre discipline, majoritairement les mathématiques.

Corpus 3 : textes des solutions des défis

Les auteurs rédigent des solutions aux défis (cf. Figure 1). Ces solutions sont publiées dès que la session du concours est terminée et restent accessibles en permanence. Outre la description de la solution elle-même, elle comporte une section intitulée « C'est de l'informatique ! » qui explique en quoi le problème posé, la situation décrite ou la solution apportée relèvent de la science informatique. La longueur de ces textes est très variable d'un défi à l'autre.

Course de grenouille

Les grenouilles programmables de Castor et Raton sont placées dans un labyrinthe. Le but est de programmer les grenouilles pour qu'elles échangent leurs positions en sautant le même nombre de fois et sans entrer en collision. Les grenouilles peuvent être programmées par une séquence de lettres dirigeant leurs sauts : **N** pour Nord, **S** pour Sud, **O** pour Ouest, et **E** pour Est.

Ordres de Castor :

Ordres de Raton :

Objectif :

Trouvez deux séries de commandes de même longueur (une série pour chaque grenouille) qui permettent aux grenouilles d'échanger leur position sans se rentrer dedans. N'hésitez pas à faire des essais, vous ne pouvez pas perdre de points.

Solution

Voici un exemple de séquences d'ordres que l'on peut donner à la grenouille de Castor et à celle de Raton, respectivement :

```
OOOSSESSEEOEOE
OONNEEOONNEEE
```

C'est de l'informatique !

Chacune des grenouilles se comporte comme un petit robot que l'on peut **programmer**. Ici, un programme est simplement une séquence de directions, chaque direction étant l'une des lettres N, S, E ou O.

Ce qui est intéressant dans ce sujet, c'est que le temps entre en compte. En effet, il y a deux grenouilles, donc il faut **synchroniser** leurs actions pour éviter une collision. De manière générale, les programmes qui ont besoin de prendre en compte des interactions avec l'environnement, et donc qui dépendent du temps sont souvent bien plus difficiles à écrire que les autres.

Figure 1 : Le défi « Course de grenouille » et sa solution.

3.2 Méthode d'analyse des corpus

Les thèmes abordés dans les corpus 2 et 3 ont fait l'objet d'une analyse thématique de contenu à l'aide du logiciel NVivo 10. Les thèmes intéressants de notre étude sont les notions et les savoirs informatiques cités par les enseignants (corpus 2) et les concepteurs des défis (corpus 3).

Nous avons procédé d'abord de manière ascendante en respectant le plus possible les expressions et termes retenus par les enseignants et concepteurs. L'unité de codage est le groupe de mots. Un premier regroupement a ainsi été effectué et a fait émerger les catégories présentées dans le tableau 3.

Tableau 3 : Premières catégories identifiées dans les discours.

Notions et savoirs	Nombre de références encodées
Algorithme et programmation	45
Codage de l'information	6
Graphes et algorithmes	25
Intelligence artificielle	1
Système, contrainte, plan	17
Tableau, Base de données	16
Utilisation d'outil	1

Tableau 4 : Correspondances entre nos catégories initiales et la classification de Dagiene et al. (2017).

Catégories initiales	Classification de Dagiene et al. (2017)
Algorithme et programmation	Algorithme et programmation
Codage de l'information	Communication et réseaux Données, structures et représentations
Graphes	Données, structures et représentations
Intelligence artificielle	Interactions, systèmes et société
Système, contrainte, plan	Algorithmes et programmation Processus informatiques et matériels
Tableau, Base de données	Données, structures et représentations
Utilisation d'outil	Processus informatiques et matériels

Cette classification a ensuite été reprise et comparée à celle proposée par Dagiene et al. (2017). Il s'est avéré que la catégorisation proposée était

compatible avec celle de ces auteurs (*cf.* Tableau 4). Les différences entre les deux peuvent s'expliquer par le fait que nos propres catégories ont été élaborées à partir des discours des enseignants et des concepteurs analysant et expliquant des cas d'études très précis. Autrement dit, les connaissances repérées et mobilisées par eux sont contextualisées et la formulation est liée à cette contextualisation. Ce n'est pas le cas de la classification proposée par Dagiene et *al.* qui a pour objet de couvrir les champs de l'informatique de manière exhaustive.

4 Résultats

Pour mettre en évidence les domaines informatiques repérés par les enseignants dans les défis sélectionnés, nous comparons le codage du corpus 2 (analyse des défis par les enseignants) à celui du corpus 3 (rubrique « C'est de l'informatique »). Le codage a été effectué selon les catégories de contenus informatiques proposées par Dagiene et *al.* (*cf.* Tableau 4).

Les résultats obtenus sont présentés ci-après. La section 4.1. donne une vue globale de la perception de l'ensemble de la sélection des défis par les enseignants et selon les intentions des concepteurs. La section 4.2 présente le résultat d'une analyse fine par enseignant et par défi et aboutit à une typologie des défis. Enfin, la section 4.3 compare les mots utilisés par les enseignants et ceux utilisés par les concepteurs.

4.1 Vue globale des domaines repérés par les enseignants et visés par les concepteurs

La figure 2 montre une vue globale de la perception de l'ensemble des défis par les trois enseignants et les intentions des concepteurs en matière de notions informatiques présentes dans les défis. Un défi n'est comptabilisé qu'une seule fois dans un même domaine, même si le codage du discours de plusieurs professeurs le rattache à ce domaine. Un défi peut appartenir à plusieurs domaines.

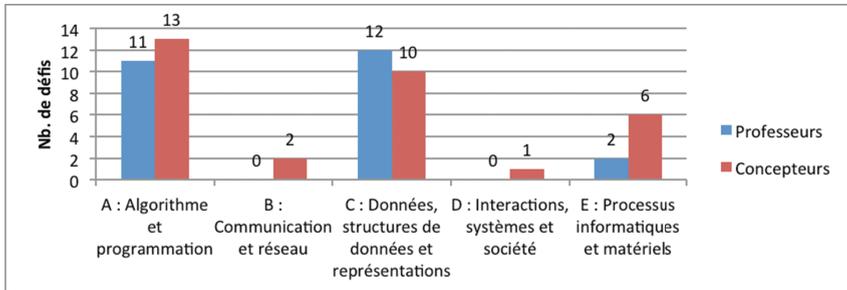


Figure 2 : Domaine des notions citées dans les corpus 2 et 3 pour les 19 défis étudiés.

Globalement, les défis proposés dans la sélection relèvent principalement des domaines « Algorithmes et programmation » (domaine A) et « Données, structures de données et représentations » (domaine C). Ce sont, en effet, les domaines les plus couverts par les défis du concours, et ce même dans les autres pays (Vanicek, 2014). L'étude des résultats du codage tant de l'analyse produite par les enseignants que de la rubrique « C'est de l'informatique » des concepteurs montre une attribution aux domaines de l'informatique relativement similaires.

Cependant, deux domaines sont absents de l'analyse faite par les enseignants : « Communication et réseau » (domaine B) et « Interactions, systèmes et société » (domaine D) alors qu'ils sont présents chez les concepteurs. Enfin, un domaine est sous-représenté chez les enseignants, « Interactions, systèmes et sociétés » (domaine E), par rapport aux intentions des concepteurs.

Finalement, les concepteurs tendent à citer davantage de notions informatiques que les enseignants, rattachant ainsi les défis à davantage de domaines informatiques. Ceci peut s'expliquer par le fait que les enseignants interrogés se sont focalisés sur les notions et savoirs directement mobilisés dans le défi, alors que les concepteurs explorent les applications et références possibles bien au-delà du contexte du défi.

4.2 *Écarts entre perceptions des enseignants et intentions des concepteurs*

Observons maintenant défi par défi l'analyse faite par les enseignants par rapport aux concepteurs. On constate des différences dans l'appréciation des notions informatiques de chaque défi (*cf.* Tableau 5).

Les résultats montrent plusieurs cas de figure, selon les accords/désaccords entre enseignants et avec les concepteurs. Nous en proposons la typologie suivante. Les libellés proposés pour chaque catégorie qualifient la capacité du défi à « se laisser interpréter » en termes de notions informatiques.

La première catégorie est celle où les domaines mentionnés par les enseignants et les concepteurs sont identiques. C'est le cas de 5 défis (3, 12, 13, 14 et 18). Nous qualifions ces défis de « transparents », car les enseignants accordent les mêmes domaines de l'informatique à ces défis que les concepteurs.

La deuxième est celle où tous les enseignants sont d'accord entre eux, mais où les concepteurs citent davantage de domaines que les enseignants. C'est le cas de 5 défis (4, 7, 9, 11 et 17). Nous qualifions ces défis d'« ambitieux », car les concepteurs y voient davantage de potentialité que les enseignants.

La troisième est celle où les domaines mentionnés par les enseignants peuvent différer entre eux et/ou avec ceux des concepteurs. C'est le cas de 4 défis (5, 8, 15 et 16). Pour le défi 8, tous les enseignants mentionnent le domaine A, ainsi que les concepteurs, mais l'un des enseignants cite 2 autres domaines, dont un seul est présent chez les concepteurs. Nous qualifions ces défis d'« opaques », car ne permettant pas aux enseignants de clairement cerner les domaines informatiques en jeu.

Enfin, 5 défis ne sont pas classés dans des domaines de l'informatique par au moins un des enseignants (1, 2, 6, 10 et 19). Nous considérons alors que les intentions des concepteurs sont « ambiguës » pour les enseignants. Nous reprenons plus en détail les propositions faites par les enseignants en nous demandant quelles sont alors les notions mobilisées par le défi, puisqu'au moins un des enseignants considère qu'il ne s'agit pas d'informatique.

Isabelle, pour le défi 1 (Affichage digital), précise qu'elle « ne voit pas » quelles sont les notions mobilisées et qu'elle considère davantage ce défi comme un jeu. Pourtant, Christophe indique comme notion la « négation, le OU inclusif ». François mentionne « l'enchaînement d'instructions ».

Tableau 5 : Domaine(s) relevé(s) dans les commentaires des enseignants (corpus 2) et des concepteurs (corpus 3) pour chacun des défis (corpus 1).

	Christophe	François	Isabelle	Concepteurs	Catégorie
1 : Affichage digital	A	A	**	ACE	Ambigu
2 : Les amis de Lucie	AC	**	C	C	Ambigu
3 : Anonymisation	C	C	C	C	Transparent
4 : Arroser laPlante	A	A	A	ADE	Ambitieux
5 : Bébrocarina	C	C	A	AB	Opaque
6 : Chercher_Remplacer	C	**	AC	A	Ambigu
7 : Choix des Invités	C	C	C	AC	Ambitieux
8 : Course de Grenouille	ACE	A	A	AE	Opaque
9 : Chez le dentiste	C	C	C	AC	Ambitieux
10 : Dessine Castor	**	A	E	ABCE	Ambigu
11 : Dessiner un dé	A	*	A	AE	Ambitieux
12 : Labyrinthe	A	A	A	A	Transparent
13 : Réseau pas cher	*	*	C	C	Transparent
14 : Robot Peintre	A	A	A	A	Transparent
15 : Rumeurs	C	*	*	E	Opaque
16 : Territoire de Castor	C	A	A	C	Opaque
17 : Transport_Trongs	C	C	C	AC	Ambitieux
18 : Trier par Poids	A	A	A	A	Transparent
19 : Trombone rouge	C	A	**	C	Ambigu

Légende

* les données ne sont pas disponibles, l'enseignant n'ayant pas rempli la grille pour ce défi.

** l'enseignant explique qu'il ne s'agit pas d'informatique.

A : Algorithmique et programmation, B : Communication et réseau,

C : Données, structures de données et représentation, D : Interactions, systèmes et société,

E : Processus informatiques et matériels.

François précise pour le défi 2 (Les amis de Lucie) qu'il s'agit selon lui « essentiellement d'un travail sur la consigne », que c'est un exercice « qui ne teste pas grand-chose ». Dans le même temps, Isabelle mentionne la notion de voisin et les graphes. Christophe évoque également les graphes mais aussi l'organisation des données, le test et le parcours d'un tableau.

Pour le défi 6 (Chercher Remplacer), il s'agit selon François d'un raisonnement logique et il n'indique pas de connaissances informatiques à mobiliser. Christian évoque le tri de l'information et l'utilisation d'un outil, tandis qu'Isabelle mentionne les chaînes de caractères et les correspondances de chaînes.

Christophe indique pour le défi 10 (Dessine un castor) qu'il ne sait pas quelles sont les notions mobilisées et qu'il ne sait pas définir les stratégies à mettre en œuvre pour le résoudre. Isabelle pour ce même défi évoque un « phénomène de rétroaction, la réponse d'un système ». Elle précise aussi qu'« en informatique, je ne vois pas trop ». François considère qu'il s'agit de mettre en place une démarche algorithmique.

Le défi 19 (Trombone rouge) relève, selon Isabelle, de la logique et elle ne voit pas quelles sont les notions informatiques mobilisées. Il est à noter que Christophe et François ne lui attribuent pas le même domaine de connaissances. Selon Christophe, le défi s'inscrit plutôt dans le domaine « Données, structures de données et représentation ». Il précise que les notions sont celles de l'organisation des données, de la lecture de graphe et du tri de l'information. Selon François, le défi s'inscrit dans le domaine « Algorithmique et programmation ». Il s'agit en effet de « mettre en place une démarche algorithmique » pour résoudre ce défi.

Nous avons pu constater les différences de représentation entre enseignants, mais aussi entre enseignants et concepteurs sur les contenus informatiques mobilisés ou mobilisables dans ces défis. Nous nous intéressons maintenant aux termes employés par les uns et les autres dans leurs discours.

4.3 Comparaison des termes employés par les enseignants et les concepteurs

Comparons les mots employés par les concepteurs dans la rubrique « C'est de l'informatique » (corpus 3) à ceux utilisés par les enseignants dans leurs commentaires (corpus 2).

Tout d'abord, nous nous intéressons aux mots du domaine informatique les plus fréquemment cités par les concepteurs (le premier quartile). La figure 3 les mentionne. Nous recherchons ensuite si ces mots sont présents dans le discours des enseignants.

On constate que certains mots employés par les concepteurs ne sont pas présents dans les écrits des enseignants. C'est le cas par exemple du mot « langage ». D'autres mots sont proportionnellement bien plus présents chez les concepteurs. C'est le cas du mot « programme » qui est cité trois fois plus souvent par les concepteurs que par les enseignants. On peut avancer l'hypothèse que la notion de programme et celle de langage sont associées dans l'esprit des enseignants à des langages de programmation usités et connus (Scratch, Python, Java, C, etc.). Or ces langages sont absents des défis du Castor. Lorsque les concepteurs utilisent le mot langage, c'est à propos d'instructions proposées par eux (par exemple les lettres N S E O pour désigner des ordres de déplacements directionnels).

Procédons de manière identique pour les enseignants (*cf.* Figure 4). Les mots « jeu » et « logique » apparaissent dans les discours des enseignants, alors qu'ils ne sont pas présents chez les concepteurs. Sans que le terme de « jeux » soit une appréciation péjorative du défi en tant que tel, cela pourrait être la reconnaissance d'une forme de ressource, certes non canonique, qui peut susciter l'intérêt des élèves. Le mot « logique » est employé par les enseignants pour souligner que ce ne sont pas d'abord des notions informatiques qui sont mises en œuvre mais une capacité à faire preuve de logique. On constate d'ailleurs que ce terme n'est pas employé par les concepteurs.

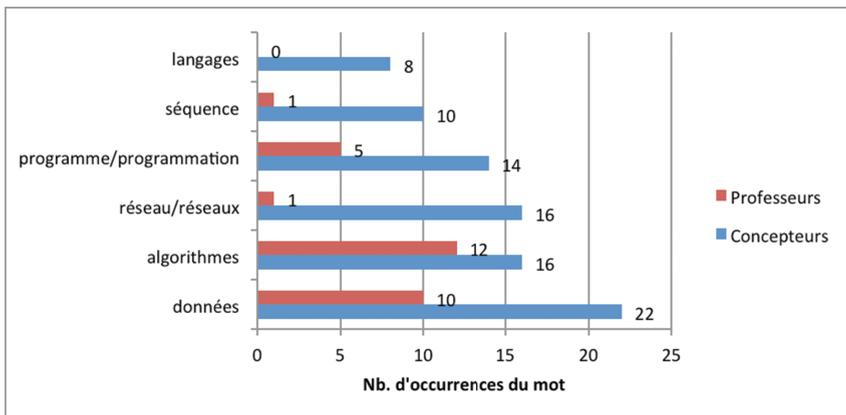


Figure 3 : Les mots du domaine informatique les plus fréquemment mentionnés (1^{er} quartile) par les concepteurs et leur présence dans les commentaires des enseignants.

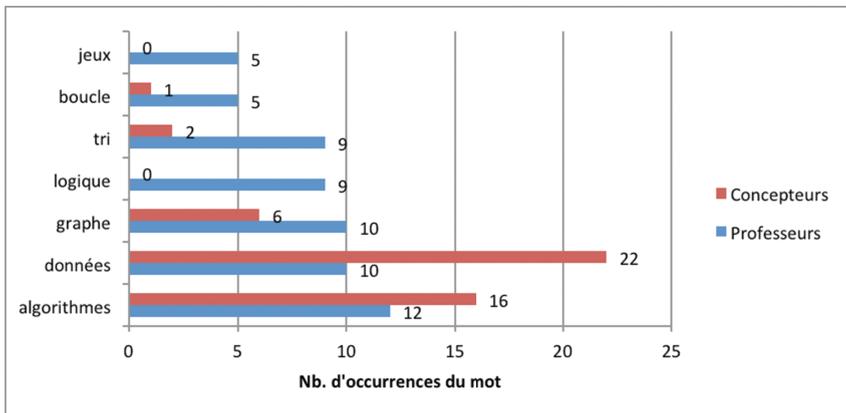


Figure 4 : Les mots du domaine informatique les plus fréquemment mentionnés (1^{er} quartile) par les enseignants et leur présence dans la rubrique « C'est de l'informatique ».

5 Discussion

Les professeurs impliqués dans cette étude enseignent l'informatique en lycée depuis plusieurs années, ils ne sont donc pas représentatifs de l'ensemble des enseignants qui sont amenés, ou qui vont l'être, à enseigner l'informatique. Nous considérons leur expertise comme un atout dans cette recherche, notamment dans leur compréhension fine des défis. L'écart entre leurs connaissances disciplinaires et celles des concepteurs est sans aucun doute moins grand que celui qu'auraient des enseignants dépourvus de cette expérience. Comparer leurs discours à celui des concepteurs a donc un sens.

Considérer les défis du concours en tant que ressources mobilisables dans un enseignement de l'informatique nous a fait nous interroger sur la structure des connaissances sous-jacentes selon les enseignants et selon les concepteurs. Dans le cas de l'informatique, cette structure ne va pas nécessairement de soi et est encore en débat dans de nombreux pays lors de l'élaboration des curricula.

Les résultats obtenus montrent que les enseignants ne perçoivent pas toujours, à partir de l'énoncé du défi, l'intérêt de celui-ci pour l'apprentissage des notions informatiques.

Certains domaines semblent toutefois plus facilement repérables que d'autres, c'est notamment le cas du domaine A (« Algorithme et programmation »), voire du domaine C (« Données, structures de données et représentations »). Outre le fait que, de manière générale, les défis mettent plus souvent ces domaines en jeu, on peut aussi supposer que les enseignants interrogés en ont une meilleure connaissance du fait de leur formation et des programmes d'ISN qu'ils sont amenés à enseigner.

On constate aussi des écarts dans les termes employés tant par les concepteurs que par les enseignants. Les premiers emploient plus souvent les termes de réseaux, de programme, de séquences, de langage alors que les seconds emploient davantage les termes de logique, de tri, de boucle, de jeu.

Nous avons pu constater que certains défis étaient « transparents » pour les enseignants, au sens où l'analyse de leurs écrits aboutissait à la même attribution de domaines informatiques que celle des concepteurs. À l'inverse, certains défis restent « ambigus », au sens où au moins un des enseignants ne perçoit pas l'intention des concepteurs en termes de notions informatiques. Les intentions des concepteurs en matière d'intérêt du défi vis-à-vis de l'informatique sont parfois ambitieuses, et ne sont pas toutes perceptibles par les enseignants.

On peut alors craindre que les élèves non plus ne perçoivent pas en quoi ces défis relèveraient de la science informatique, avec un risque de résultats contreproductifs par rapport aux objectifs du concours Castor, à savoir la promotion de l'informatique auprès des élèves. Ce risque a déjà été constaté par Taub, Armoni et Ben-Ari (2012) dans le cas de l'informatique sans ordinateur. Les auteurs étudient si les objectifs assignés à cet enseignement sont remplis auprès d'élèves de collèves (grade 7), à savoir s'ils ont une représentation plus claire de l'informatique et s'ils ont envie d'en continuer l'étude plus tard. Les résultats obtenus montrent que les élèves sont moins attirés par l'informatique et la considèrent comme moins intéressante que ce qu'ils déclaraient avant de faire les activités sans ordinateur. L'hypothèse faite par les auteurs est que les élèves ne comprennent pas de quelle manière les activités proposées sont représentatives de l'informatique étudiée dans l'enseignement secondaire ou d'une éventuelle future carrière. Les élèves ne feraient pas le lien entre les activités proposées et les concepts informatiques, faute d'explicitation. Pourtant, là aussi, les documents à destination des enseignants les précisent.

Comment rendre plus « transparentes » les intentions des concepteurs du concours Castor et faciliter l'exploitation pédagogique et disciplinaire des défis ? Une solution mise en œuvre par les organisateurs du concours en Grande-Bretagne consiste à utiliser explicitement la classification proposée dans Dagiene *et al.* (2017). Ainsi, dans le livret présentant l'ensemble des défis du concours et leurs solutions⁸, il est fait mention pour chaque défi de deux catégories : la compétence relevant de la pensée informatique sollicitée et le domaine informatique couvert. Il s'agit en quelque sorte de fournir des métadonnées pour chaque défi, pour faciliter le travail de l'enseignant lorsque celui-ci souhaite s'approprier ces défis comme ressources pour son enseignement.

Cette question de la documentation de ressources dites « clés en main » en informatique n'est pas nouvelle (Drot-Delange, 2013). Elle se pose avec d'autant plus d'acuité que les enseignants n'ont pas toujours une connaissance des concepts. Cette question a déjà été soulevée par les initiateurs du mouvement *Computer Unplugged*. La première édition de leur manuel (Fellows, Bell & Witten, 1996) livrait des activités, sans instruction particulière pour les enseignants. Mais il s'est vite avéré qu'un minimum d'explications était nécessaire pour des enseignants qui n'avaient pas toujours la culture informatique et mathématique nécessaire à la prise en main et à l'exploitation de ces activités (Bell, Rosamond & Casey, 2012). Une version écrite par des enseignants pour les enseignants a donc été éditée (Fellows, Bell & Witten, 2002) comprenant des activités testées en classe. C'est la même logique qui prévaut dans Calmet, Hirtzig et Wilgenbus (2016) où les défis du concours Castor sont recensés par cycle et intégrés dans les progressions proposées par l'ouvrage.

6 Conclusion et perspectives

Nous souhaitons savoir dans quelle mesure les défis du concours Castor pouvaient constituer des ressources à destination des enseignants dans le cadre d'un enseignement de l'informatique. Pour cela, nous avons postulé

8 2016 *Competition complete answer book* : <<http://www.bebbras.uk/uploads/2/1/8/6/21861082/uk-bebbras-2016-answers.pdf>>. Consulté le 4 septembre 2017.

que la structure de connaissances informatiques sous-jacentes aux défis devait correspondre à celles des enseignants.

Nous avons montré que les énoncés des exercices seuls ne permettaient pas toujours aux enseignants, possédant une expérience importante dans l'enseignement de l'informatique, de cerner quels étaient les champs de l'informatique mobilisés ou potentiellement mobilisables dans les défis. Ceci en rend donc complexe l'utilisation dans une classe à des fins pédagogiques pour l'enseignant.

L'existence de défis « ambitieux », « opaques » ou « ambigus » pose question d'une part sur la mise en valeur du potentiel pédagogique de chaque défi et d'autre part sur la rédaction des solutions et des rubriques « C'est de l'informatique ». La documentation associée aux défis et la communication des métadonnées décrivant un défi permettraient probablement aux enseignants de mieux cerner les intentions des concepteurs. Le chantier de la construction d'une classification est en cours dans les équipes de recherche. Il conviendrait que celles-ci n'associent pas seulement des concepteurs mais aussi des enseignants pour s'assurer de l'utilité de cette classification.

Les perspectives de cette recherche sont de trois ordres. Nous avons proposé dans cette communication une typologie des défis basée sur la complétude des domaines de l'informatique repérés par les enseignants. On peut se demander si ces défis présentent structurellement des caractéristiques qui les distinguent les uns des autres, ou si ce sont les connaissances des enseignants qui font la différence. Ensuite, nous souhaitons focaliser le questionnement sur l'activité de l'élève lors de la résolution de défis. Il s'agirait alors d'identifier les stratégies de résolution mises en œuvre et l'analyse qu'en font les enseignants. Enfin, dans une démarche complémentaire à celle présentée ici, on évaluera la pertinence des défis proposés par rapport aux domaines qu'ils sont censés couvrir.

Références

- Baron, G.-L., Drot-Delange, B., Grandbastien, M., & Tort, F. (2015). L'enseignement de l'informatique dans l'enseignement secondaire en France : un retour de balancier ? Dans G.-L. Baron, E. Bruillard,

- B. Drot-Delange (eds.). *Informatique en éducation : perspectives curriculaires et didactiques* (p. 83–101). Clermont-Ferrand : Presses Universitaires Blaise Pascal.
- Bell, T., Rosamond, F. et Casey, N. (2012). Computer Science Unplugged and Related Projects in Math and Computer Science Popularization. Dans H. L. Bodlaender, R. Downey, F. V. Fomin et D. Marx (dir.), *The Multivariate Algorithmic Revolution and Beyond* (vol. 7370, pp. 398–456). Berlin, Heidelberg : Springer Berlin Heidelberg.
- Bruillard, É. (2016). Quelle informatique à repenser et à construire pour les élèves de l'école primaire ? Dans F. Villemonteix, G.-L. Baron et J. Béziat (dir.), *L'école primaire et les technologies informatisées. Des enseignants face aux TICE*. (pp. 29–37). Villeneuve d'Asq : Presses Universitaires du Septentrion.
- Calmet, C., Hirtzig, M. et Wilgenbus, D. (2016). *1, 2, 3... Codez ! Le Pommier*.
- Dagiene, V., Sentance, S. et Stupuriene, G. (2017). Developing a Two-Dimensional Categorization System for Educational Tasks in Informatics. *Informatica*, 28(1), 23–44.
- Denning, P. J. et Martell, C. H. (2015). *Great Principles of Computing*. The MIT Press.
- Drot-Delange, B. (2013). Enseigner l'informatique débranchée : analyse didactique d'activités (pp. 1–13). Présenté à Actualité de la Recherche en Education et en Formation, Montpellier : France.
- Fellows, M. R., Bell, T. et Witten, I. (1996). *Computer Science Unplugged... offline activities and games for all ages : Original Activities Book*. Computer Science Unplugged.
- Fellows, M. R., Bell, T. et Witten, I. (2002). *Computer science unplugged*.
- Selby, C. & Woollard, J. (2013). Computational thinking : the developing definition, available via internet : <<http://eprints.soton.ac.uk/356481>>. Consulté le 30 avril 2016.
- Shulman, L.-S. (2007). Ceux qui comprennent. *Éducation et didactique*, 1(1), 97–114.
- Taub, R., Armoni, M. et Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education (TOCE)*, 12(2), 8.

- Tort, F. et Dagiene, V. (2012). Concours Castor : découvrir l'informatique autrement/*E-dossier de l'audiovisuel : L'éducation aux cultures de l'information*/.
- Tort, F. et Drot-Delange, B. (2015). Visual Literacy in Introductory Informatics Problems. Dans A. Brodnik et J. Vahrenhold (dir.), *Informatics in Schools. Curricula, Competences, and Competitions* (vol. 9378, pp. 175–182). Springer International Publishing.
- Tort, F., Drot-Delange, B. et Mano, M. (2017). Filles et informatique : qu'en est-il du concours Castor ? Dans J. Henry, A. Nguyen et É. Vandeput (dir.), *L'informatique et le numérique dans la classe. Qui, quoi, comment ?* (pp. 69–84). Namur : Presses Universitaires de Namur.
- Vanicek J., (2014) Bebras Informatics Contests ; Criteria for Good Tasks Revised. Dans Y. Gülbahar and E. Karatas (Eds) : *ISSEP 2014, LNCS 8730* (pp. 17–28). Springer.
- Villemonteix, F. (2017). L'enseignement de l'informatique à l'école primaire vu par les acteurs de l'accompagnement des pratiques pédagogiques. Dans J. Henry, A. Nguyen et É. Vandeput (dir.), *L'informatique et le numérique dans la classe. Qui, quoi, comment ?* (pp. 139–152). Namur : Presses Universitaires de Namur.