

Université de Montréal

La programmation informatique à l'école primaire

*Pratiques effectives de programmation et mobilisation d'habiletés de
résolution collaborative de problèmes (RCP)*

Par

Simon Parent

Département de psychopédagogie et d'andragogie, Faculté des sciences de l'éducation

Thèse présentée en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)

en sciences de l'éducation, option psychopédagogie

Janvier 2021

© Simon Parent, 2021

Université de Montréal

Département de psychopédagogie et d'andragogie, Faculté des sciences de l'éducation

Cette thèse intitulée

La programmation informatique à l'école primaire

*Pratiques effectives de programmation et mobilisation d'habiletés de
résolution collaborative de problèmes (RCP)*

Présentée par

Simon Parent

A été évaluée par un jury composé des personnes suivantes

Normand Roy

Président-rapporteur

Maurice Tardif

Directeur de recherche

Michel Lepage

Membre du jury

Margarida Romero

Examinatrice externe

Résumé

L'objectif de cette recherche doctorale est de décrire et comprendre les effets de la programmation informatique sur l'apprentissage et la mobilisation de compétences d'élèves du primaire. Notre intérêt pour ce phénomène se justifie sous deux considérations : d'abord, la programmation est une activité qui a connu une démocratisation au cours des dernières années, notamment dans le milieu scolaire. Puis, en ce qui concerne la résolution collaborative de problèmes (RCP), il s'agit de compétences appartenant à ce que certains appellent les compétences du 21^e siècle (voir notamment Chalkiadaki, 2018; van Laar et al., 2017), et que d'autres appellent les compétences du futur (Conseil des compétences futures, 2020; Gouvernement du Canada, 2020). Les recherches empiriques menées en contextes authentiques de classe sont très peu nombreuses dans la littérature (Lye et Koh, 2014), ce qui atteste pour nous la nécessité de porter un regard contemporain et imprégné des nouvelles tendances technopédagogiques sur ce phénomène d'envergure internationale.

Les principaux référents conceptuels sont associés au champ de l'apprentissage collaboratif à l'aide de l'ordinateur (en anglais le *Computer-supported collaborative learning, CSCL*), un champ dont nous citons plusieurs travaux réalisés dans les années 1990 (Koschmann et al., 1996; Scardamalia et Bereiter, 1994). Il nous a permis d'aborder les aspects techniques (programmation, ordinateur), sociaux (interactions, médiatisation) et pédagogiques (apprentissage, développement de compétences) du phénomène que nous souhaitons étudier.

À l'aide d'un devis qualitatif et interprétatif de type *étude de cas multiples*, nous avons mis en place un scénario pédagogique dans trois écoles primaires au Québec. Amenant les élèves à découvrir graduellement les concepts fondamentaux de la programmation informatique, le scénario *Deviens un maître NAO* mise sur la programmation visuelle et tangible pour mobiliser, voire développer, un ensemble d'habiletés et de compétences, tant transversales que disciplinaires. Une série de 20 niveaux sont proposés aux élèves, qui doivent réaliser des programmes ayant pour but d'animer un robot humanoïde nommé NAO. Le premier cas a été observé dans une école alternative de Montréal. Les élèves (n=29) ont été séparés en équipes multiniveau, ce qui a amené la collaboration d'élèves d'âges et de niveaux scolaires différents.

Les cas 2 à 5 ont été étudiés dans une école de Montréal et présentent des élèves de quatrième, cinquième et sixième année (n=70). Enfin, le dernier cas se situait dans une classe de cinquième année de la grande région de Québec (n=10). Plusieurs de ces élèves ont des difficultés d'apprentissage ou sont en situation de handicap.

Nous avons observé les pratiques effectives de programmation ainsi que la mobilisation des compétences de RCP à l'aide d'un appareillage méthodologique considérable : (a) une caméra filmant l'activité des élèves dans leur environnement immédiat, (b) la caméra intégrée de l'ordinateur qui filmait le visage des élèves et leurs interactions verbales, et (c) l'enregistrement de l'écran d'ordinateur que les élèves utilisaient pour faire la programmation visuelle et tangible du robot NAO. Nous avons ensuite utilisé ce corpus de données vidéos pour déterminer une typologie des pratiques effectives de programmation d'élèves du primaire, pour créer une grille d'observation de la RCP à partir d'observations empiriques et authentiques et pour comprendre le processus de RCP dans le cadre d'une activité de programmation au primaire.

Parmi les principaux résultats, nous proposons une typologie des pratiques de programmation visuelle fortement basée sur nos observations empiriques, ce qui l'ancre fortement dans la réalité du terrain de recherche. Puis, nous avons également été en mesure de concevoir une grille d'observation de la RCP en nous appuyant sur les travaux scientifiques antérieurs et, surtout, sur nos observations empiriques. Cette grille, présentant 3 dimensions et 15 indicateurs, permet d'observer les processus et habiletés mis en œuvre par les élèves de façon collaborative. Enfin, l'utilisation de cette grille a mené à l'analyse systématique et compréhensive de la façon dont les élèves parvenaient à compléter les activités de programmation de façon collaborative. Nous soulignons d'une part la prépondérance de certaines habiletés (échanges à propos du problème, attribution explicite ou implicite des rôles, vérification des solutions, etc.), et d'autre part la faible mobilisation d'habiletés comme l'adaptation des interventions, l'identification des forces et des faiblesses et la vérification des actions.

Mots-clés : programmation informatique, résolution collaborative de problèmes (RCP), programmation tangible, programmation visuelle, grille d'observation.

Abstract

The objective of this doctoral research is to describe and understand the effects of computer programming on the learning and mobilization of skills of elementary school students. Our interest in this phenomenon is justified by two considerations: first, programming is an activity that has been democratized in recent years, especially in the school environment. Second, with respect to collaborative problem solving (CPS), these are skills that belong to what some call 21st century skills (see, for example, Chalkiadaki, 2018; van Laar et al., 2017), and others call future skills (Future Skills Council, 2020; Government of Canada, 2020). Empirical research conducted in authentic classroom contexts is very scarce in the literature (Lye and Koh, 2014), which attests to the need for us to take a contemporary look at this international phenomenon, infused with new techno-pedagogical trends.

The main conceptual referents are associated with the field of Computer-supported collaborative learning (CSCL), a field from which we cite several studies conducted in the 1990s (Koschmann et al., 1996; Scardamalia and Bereiter, 1994). It allowed us to address the technical (programming, computer), social (interactions, mediatization) and pedagogical (learning, skill development) aspects of the phenomenon we wanted to study.

Using a qualitative and interpretative multiple case study design, we implemented a pedagogical scenario in three elementary schools in Quebec. By gradually introducing students to the fundamental concepts of computer programming, the Become a NAO Master scenario relies on visual and tangible programming to mobilize and even develop a set of skills and competencies, both transversal and disciplinary. A series of 20 levels are offered to students, who must create programs to animate a humanoid robot named NAO.

The first case was observed in an alternative school in Montreal. The students (n=29) were separated into multi-level teams, which led to the collaboration of students of different ages and grade levels. Cases 2-5 were studied in a Montreal school and featured students in grades 4, 5, and 6 (n=70). Finally, the last case was in a fifth grade class in the greater Quebec City area (n=10). Many of these students have learning difficulties or disabilities.

We observed the effective programming practices as well as the mobilization of CPR skills using a considerable methodological apparatus. Indeed, for each team in each class, we had: (a) a camera filming students' activity in their immediate environment, (b) the computer's built-in camera that filmed students' faces and verbal interactions, and (c) the recording of the computer screen that students used to do the visual and tangible programming of the NAO robot. We then used this video data set to determine a typology of elementary students' effective programming practices, to create a CPS observation grid based on empirical and authentic observations, and to understand, using the created grid, the process of CPS in an elementary programming activity.

Among the main results, we propose a typology of visual programming practices strongly based on our empirical observations, which anchors it strongly in the reality of the research field. Then, we were also able to design a CPS observation grid based on previous scientific works and, above all, on our empirical observations. This grid, presenting 3 dimensions and 15 indicators, allows us to observe the processes and skills implemented by the students in a collaborative manner. Finally, the use of this grid led to a systematic and comprehensive analysis of the way in which students managed to complete programming activities in a collaborative manner. On the one hand, we note the preponderance of certain skills (exchanges about the problem, explicit or implicit assignment of roles, verification of solutions, etc.), and on the other hand, the weak mobilization of skills such as adapting interventions, identifying strengths and weaknesses, and verifying actions.

Keywords : computer programming, collaborative problem solving (CPS), tangible programming, visual programming, observation grid.

Table des matières

LISTE DES TABLEAUX.....	XIII
LISTE DES FIGURES.....	XV
LISTE DES SIGLES ET ABRÉVIATIONS	XVII
REMERCIEMENTS.....	XXI
INTRODUCTION	1
CHAPITRE 1 - PROBLÉMATIQUE	5
1.1 L’omniprésence technologique	5
1.2 La programmation : un langage informatique.....	8
1.3 La programmation en contexte scolaire.....	9
1.3.1 Genèse du phénomène et implications pratiques.....	9
1.3.2 Une conjoncture internationale	11
1.3.3 La situation canadienne et québécoise	14
1.4 Les outils principaux : des plateformes en ligne aux dispositifs robotiques.....	17
1.5 Effets de l’utilisation de la programmation sur l’apprentissage.....	18
1.6 L’objectif général de recherche	23
1.7 La pertinence de cette recherche.....	23
CHAPITRE 2 - CADRE THÉORIQUE.....	27
2.1 La programmation : un outil de médiation humain-ordinateur	27
2.1.1 Perspectives pédagogiques de la programmation.....	30
2.1.2 L’apprentissage de la programmation au primaire	31
2.2 Les compétences transversales	36
2.2.1 La résolution de problèmes	41
2.2.2 La collaboration	45
2.2.3 La résolution collaborative de problèmes (RCP).....	49
2.3 Distinction entre développement et mobilisation de compétences	51
2.3.1 Enquête PISA 2015.....	51
2.3.2 De la nécessité d’étudier les aspects inhérents à la RCP	53
2.4 Objectifs spécifiques de recherche.....	54

CHAPITRE 3 - MÉTHODOLOGIE.....	57
3.1 Type de recherche	57
3.2 Sélection des cas.....	58
3.2.1 Échantillon	59
3.2.2 Processus de recrutement.....	59
3.3 Présentation des cas.....	60
3.3.1 Cas #1 : Classe A1	60
3.3.2 Cas #2 à 5 : Classes B1 à B4.....	60
3.3.3 Cas #6 : Classe C1.....	61
3.4 Processus de collecte de données	62
3.4.1 Le robot humanoïde NAO	63
3.4.2 Le logiciel Chorégraphe	64
3.4.3 Scénario pédagogique <i>Deviens un maître NAO</i>	65
3.5 Instrument de collecte de données.....	66
3.5.1 Appareillage d'observation vidéographiée.....	67
3.5.2 Typologie des pratiques effectives de programmation	68
3.5.3 Grille d'observation de la RCP	69
3.6 Traitement et analyse des données.....	70
3.6.1 Codage mixte.....	70
3.6.2 Codage fermé	71
3.7 Forces et limites de la recherche.....	72
3.8 Présentation des articles	73
CHAPITRE 4 - PRATIQUES EFFECTIVES DE PROGRAMMATION VISUELLE CHEZ DES ÉLÈVES DU PRIMAIRE : PROPOSITION D'UNE TYPOLOGIE COMPRÉHENSIVE ET ADAPTÉE	77
4.1 Contexte	78
4.2 Cadre théorique.....	80
4.2.1 Les pratiques de programmation : aperçu de la littérature	83
4.3 Méthodologie	84
4.3.1 Collecte de données	86
4.3.2 Modalités.....	87
4.3.3 Traitement et analyse des données.....	88
4.4 Résultats	88
4.4.1 Adaptation de la classification.....	88

4.4.2	Analyse des pratiques de programmation visuelle observées	89
4.4.3	Mobilisation des pratiques de programmation visuelle	93
4.5	Discussion	95
4.5.1	Les pratiques les plus utilisées.....	95
4.5.2	Typologie des pratiques de programmation	96
4.6	Discussion	99
4.6.1	Pertinence de la typologie proposée	101
4.7	Limites	101
4.8	Conclusion	102
CHAPITRE 5 - CONCEPTION D'UNE GRILLE D'OBSERVATION DE LA RÉOLUTION COLLABORATIVE DE PROBLÈMES (RCP)		104
5.1	Contexte	105
5.2	Cadre théorique.....	107
5.2.1	L'observation	109
5.2.2	L'observation de la RCP dans la littérature.....	110
5.3	Méthodologie	111
5.3.1	Phase descriptive.....	112
5.3.2	Phase exploratoire.....	112
5.3.3	Évaluation systématique.....	113
5.4	Résultats	114
5.4.1	Phase descriptive.....	114
5.4.2	Phase exploratoire.....	116
5.4.3	Évaluation systématique.....	117
5.5	Discussion	119
5.5.1	Justesse de la grille	123
5.5.2	Usages potentiels de la grille	123
5.6	Limites	124
5.7	Conclusion	125
CHAPITRE 6 - COMPRENDRE LE PROCESSUS DE RÉOLUTION COLLABORATIVE DE PROBLÈMES (RCP) LORS D'UNE ACTIVITÉ DE PROGRAMMATION INFORMATIQUE AVEC DES ÉLÈVES DU PRIMAIRE		127
6.1	Contexte	128
6.2	Cadre théorique.....	130

6.3	Méthodologie	133
6.3.1	Échantillon	133
6.3.2	Collecte des données	134
6.3.3	Instruments	135
6.3.4	Analyse des données	135
6.4	Résultats	136
6.4.1	Les indicateurs de fréquence	137
6.4.2	Les indicateurs de séance	138
6.5	Discussion	139
6.5.1	L'analyse des indicateurs de fréquence	139
6.5.2	L'analyse des indicateurs de séance	144
6.5.3	Facteurs pouvant influencer la mobilisation des compétences de RCP	145
6.6	Forces et limites	147
6.7	Conclusion	148
CHAPITRE 7 - CONCLUSION GÉNÉRALE		151
7.1	Principaux résultats	151
7.2	Pistes d'applications pratiques et de recherches futures	155
7.2.1	Dans les milieux de pratique	155
7.2.2	Dans le milieu de la recherche	156
7.2.3	Recommandation	157
RÉFÉRENCES BIBLIOGRAPHIQUES		159
ANNEXES		177
	Annexe 1 – Scénario pédagogique <i>Deviens un maître NAO</i>	179
	Annexe 2 – Modèle de la compétence de RCP (Kamga et al., 2017)	182
	Annexe 3 – Grille d'observation de la RCP (version synthétique)	183

Cette recherche a été approuvée par le Comité d'éthique de la recherche en éducation et en psychologie de l'Université de Montréal (CPER-17-123-P).

Liste des tableaux

Tableau 1. –	Types de tâches de programmation (Ruf, Berges et Hubweiser, 2015)	29
Tableau 2. –	Tableau synthèse des compétences du 21 ^e siècle	39
Tableau 3. –	Tableau synthétique intégrateur de la RP	44
Tableau 4. –	Matrice de la RCP (OCDE, 2017)	50
Tableau 5. –	Tableau synthèse des participants (par cas)	61
Tableau 6. –	Types de tâches de programmation (Ruf, Berges et Hubweiser, 2015)	69
Tableau 7. –	Types de tâches de programmation (Ruf, Berges et Hubweiser, 2015)	84
Tableau 8. –	Participants à l'étude	85
Tableau 9. –	Classification adaptée de Ruf, Berges et Hubweiser (2015)	89
Tableau 10. –	Comparaison des proportions de durée et de fréquence d'encodage	95
Tableau 11. –	Typologie des pratiques effectives de programmation	99
Tableau 12. –	Participants	113
Tableau 13. –	Grille d'observation de la RCP	122
Tableau 14. –	Échantillon de l'étude	133
Tableau 15. –	Fréquence relative des comportements, par groupe	137
Tableau 16. –	Indicateurs de séance pour les trois séances, par groupe	139

Liste des figures

Figure 1. –	Programmation sous la forme de blocs	8
Figure 2. –	Représentations pour une série de commandes sur studio.code.org	32
Figure 3. –	Compétences clés pour le 21 ^e siècle (Romero et Vallerand, 2016)	41
Figure 4. –	Modèle du processus de RP (Robertson, 2017).....	44
Figure 5. –	Modèle de la théorie de l'activité (Engeström, 1987).	46
Figure 6. –	Interface de la tâche <i>The Aquarium</i> (OCDE, 2017)	52
Figure 7. –	Interface du logiciel Choregraphe (Aldebaran Robotics, 2014).....	64
Figure 8. –	Démarche générale de l'activité de programmation.....	66
Figure 9. –	Types d'usages de la programmation (Romero, Lepage et Lille, 2017).	83
Figure 10. –	Interface de programmation visuelle du logiciel Choregraphe	87
Figure 11. –	Proportions de fréquence d'encodage des pratiques de programmation par niveau du scénario pédagogique	93
Figure 12. –	Proportions de fréquence d'encodage des pratiques d'assemblage fondamentales (4) par niveau	94
Figure 13. –	Représentation graphique du modèle de Strayer et Gauthier (1982).....	112
Figure 14. –	Modèle de l'activité d'Engeström (1987)	130
Figure 15. –	Modèle de l'activité en contexte de résolution collaborative de problèmes à l'aide de la programmation tangible (adapté du modèle de l'activité d'Engeström, 1987).	135
Figure 16. –	Fréquence des comportements pour les trois séances, par groupe	138
Figure 17. –	Diagramme synthèse de la recherche	154

Liste des sigles et abréviations

CEFRIO : Centre facilitant la recherche et l'innovation dans les organisations

CRCN : Cadre de référence de la compétence numérique

EDI: Ministère de l'Emploi, du Développement de la main-d'œuvre et de l'Inclusion des personnes handicapées

HITSA : The Information Technology Foundation for Education

ISDE: Ministère de l'Innovation, des Sciences et du Développement économique du Canada

MÉES : Ministère de l'Éducation et de l'Enseignement supérieur (depuis 2016)

MELS : Ministère de l'Éducation, du Loisir et du Sport (2005 à 2015)

MIT : Massachusetts Institute of Technology (États-Unis)

PA : Pensée algorithmique

PANÉES : Plan d'action numérique en éducation et en enseignement supérieur

PFÉQ : Programme de formation de l'école québécoise

PI : Pensée informatique

RCP : Résolution collaborative de problèmes

RÉPAQ : Réseau des écoles publiques alternatives du Québec

RP : Résolution de problèmes

RPComp : Résolution de problèmes complexes

TIC : Technologies de l'information et de la communication

À mes parents.

Remerciements

Je tiens à remercier sincèrement toutes les personnes qui ont, directement ou indirectement, contribué à la réalisation de cette recherche. Ce projet est le fruit de nombreuses collaborations que je souhaite souligner ici. Bien qu'il soit difficile de nommer tout le monde, je tiens tout de même à remercier particulièrement certaines personnes.

D'abord, j'aimerais remercier les membres du jury : Normand Roy et Michel Lepage pour leurs commentaires constructifs et généreux, qui m'ont permis de mener à terme cette recherche. Leur soutien à différents moments au cours de mes études supérieures a été très apprécié. Je remercie également Maurice Tardif d'avoir accepté de m'accompagner et de m'avoir soutenu pour la fin de mes études doctorales, de même que Margarida Romero, qui a accepté de se joindre au jury et offert des rétroactions très aidantes fondées sur sa grande expertise.

La conception des instruments et l'analyse des données ont été réalisées avec la collaboration de Sonia Lefebvre et Raoul Kamga Kouamkam, qui ont gentiment accepté de collaborer au processus initial de création de la grille d'observation, ainsi que Pierre-Luc et Simon, qui ont patiemment et minutieusement collaboré au traitement et à l'analyse d'un corpus de données imposant.

Merci à Nicolas et Bong-Sou, pour leur aide lors des nombreuses collectes de données, et surtout, pour leur complicité et leur amitié.

Un grand merci à mes collègues de l'ACSE pour leur bienveillance et leur soutien indéfectible tout au long de mon parcours.

Je tiens à remercier également toutes les enseignantes et tous les enseignants, de même que les directrices et directeurs d'établissement, qui ont collaboré de près ou de loin à ce projet. Sans elles et sans eux, ce projet n'aurait pas été possible.

Enfin, les mots me manquent pour exprimer la reconnaissance que j'ai envers mes parents, qui m'ont offert toutes les chances de réussir et m'ont amené à me dépasser.

Cette recherche a été rendue possible grâce au soutien financier du Conseil de recherches en sciences humaines (CRSH) et du Fonds de recherche du Québec – Société et culture (FRQSC).

Introduction

Avec l'évolution rapide et continue des outils numériques, de plus en plus de gens à travers le monde doivent faire face, de différentes façons, aux impacts d'une conjoncture que le Forum économique mondial (2016) qualifie de 4^e Révolution industrielle. C'est alors qu'une question point à l'horizon : comment préparer les citoyens de demain à faire face aux avancées technologiques et à leur impact sur l'éducation et le marché du travail. Souvent motivés par un certain utilitarisme économique, plusieurs gouvernements à travers le monde tentent d'anticiper les besoins futurs en adaptant le cursus aux différents impératifs techniques et technologiques du marché du travail de demain. Or, l'impératif économique n'est pas – et ne devrait pas être – la seule motivation. En effet, l'intérêt de sensibiliser et de former les élèves aux nouveaux outils numériques dépasse le simple fait de former de futurs travailleurs.

Il serait par ailleurs difficile de ne pas mentionner le contexte particulier dans lequel nous avons achevé ce manuscrit. En effet, la pandémie de coronavirus a eu des répercussions dévastatrices à divers égards, notamment la santé, l'économie et l'éducation. En ce qui concerne les élèves, elle a mis en évidence la nécessité que ces derniers maîtrisent une gamme d'outils numériques variés pouvant être utilisés dans différents contextes. Plusieurs pays du monde ont porté une attention particulière à l'utilisation du numérique en éducation, amenant ainsi de nombreux gouvernements à instaurer différentes politiques ou différents programmes visant à favoriser l'utilisation du numérique. Des outils numériques sont utilisés non seulement en tant qu'objets d'apprentissage, mais aussi en tant qu'outils d'apprentissage, en vue de développer d'autres compétences tant disciplinaires que transversales (comme la résolution de problèmes et la collaboration). Au Canada, avec de multiples systèmes d'éducation provinciaux, le gouvernement fédéral a un rôle plus limité. Cela n'a pas empêché le ministère fédéral de l'Emploi, du Développement de la main-d'œuvre et de l'Inclusion des personnes handicapées (EDI) de constituer un conseil des compétences futures, dont le rôle est d'offrir des indications quant aux tendances associées aux compétences et à la main-d'œuvre (Gouvernement du Canada, 2020). Dans son rapport, le Conseil recommande de « développer et élargir l'accès à la formation en

compétences numériques et technologies émergentes » (Conseil des compétences futures, 2020, p. 48).

Au Québec, l'intérêt politique envers l'utilisation des outils numériques en éducation s'est manifesté par une succession de mesures : d'abord le lancement du *Plan d'action numérique en éducation et en enseignement supérieur* (PANÉES), puis la parution du *Cadre de référence de la compétence numérique*. Ensemble ces deux mesures ont établi une grande orientation, qui allait ensuite se décliner en différentes retombées, notamment à l'égard de la programmation informatique. Par exemple, avec son guide *L'usage pédagogique de la programmation informatique*, le ministère de l'Éducation (2020) a offert des pistes aux enseignants souhaitant intégrer la programmation à leur enseignement. L'utilisation de la programmation en contexte scolaire est d'ailleurs devenue, au cours des dernières années, un phénomène d'envergure planétaire. En effet, les curriculums nationaux de plusieurs pays comportent maintenant des éléments relatifs à la programmation informatique, l'un des principaux outils évoqués lorsqu'il est question des compétences essentielles pour les futurs travailleurs.

Ainsi, comme la programmation est appelée à prendre un rôle d'une importance grandissante dans le contexte scolaire au cours des prochaines années, nous avons cru nécessaire d'étudier son utilisation auprès d'élèves. L'objectif de cette recherche est donc de décrire et comprendre les impacts de la programmation informatique sur la mobilisation des habiletés de résolution collaborative de problèmes (RCP) d'élèves du primaire. Afin de répondre à ce grand objectif, nous avons mis en place une série d'activités de programmation par blocs impliquant l'utilisation de robots humanoïdes nommés NAO. L'originalité de cette recherche repose principalement dans l'arrimage entre les compétences observées d'une part (la résolution de problèmes et la collaboration) et les outils utilisés d'autre part (robot humanoïde et logiciel de programmation par blocs). De surcroît, le fait de mener cette recherche avec des élèves du primaire est aussi digne de mention, dans la mesure où la programmation est traditionnellement¹ utilisée avec des élèves plus âgés (à partir de l'enseignement secondaire). Bien que l'utilisation de la programmation se soit démocratisée au cours des dernières années, les études empiriques

¹ Nous faisons ici référence à la littérature consultée, que nous abordons dans le premier et second chapitre de ce manuscrit.

portant sur son utilisation au primaire sont relativement peu nombreuses et l'utilisation de robots humanoïdes avec ces élèves est novatrice.

Ce manuscrit présente donc une recherche doctorale structurée autour de l'objectif général précité. Le premier chapitre porte sur la problématique à l'origine de notre intérêt de recherche. Nous y présentons l'ensemble des éléments contextuels et historiques permettant de comprendre l'origine de l'utilisation pédagogique de la programmation de même que son évolution au fil des années. S'inscrivant dans une conjoncture internationale, la démocratisation de la programmation informatique a amené le développement de nombreux outils accessibles, que ce soit sous la forme d'interfaces (plateformes, sites internet, applications, etc.) ou sous la forme de petits robots programmables. Le second chapitre présente le cadre théorique. Nous y abordons les principaux concepts et la théorie qui ont apporté une perspective scientifique au phénomène décrit dans la problématique, notamment les applications pédagogiques de la programmation, les compétences transversales, ainsi que la théorie de l'activité, qui offre une lentille des plus intéressantes pour observer le phénomène ciblé. Le troisième chapitre porte sur la méthodologie de la recherche : cette section présente comment nous avons été en mesure d'observer la programmation informatique en contexte scolaire par l'explicitation des activités qui ont été mises en place (le scénario pédagogique *Deviens un maître NAO*), par la présentation des 109 participants provenant de trois écoles primaires du Québec, puis par l'énumération des instruments d'observation participante et systématique utilisés et des méthodes d'analyses correspondantes. Le quatrième chapitre décrit brièvement les trois articles qui représentent le cœur de ce manuscrit. Chacun associé à un objectif spécifique (voir le Chapitre 4), ces trois articles doivent être considérés comme complémentaires afin de permettre l'atteinte de l'objectif général.

Le premier article vise à identifier les pratiques effectives de programmation d'élèves dans une activité de programmation en collaboration. Nous y proposons une typologie basée sur les travaux de Ruf, Berges et Hubweiser (2015), puis adaptée au contexte qui nous occupe. Le second article présente le processus ayant mené à la création d'une grille d'observation inédite de la résolution collaborative de problèmes. En nous inspirant du modèle proposé par Strayer et Gauthier (1982), nous présentons les trois phases de création de cette grille, de même que la

version la grille en soi, qui s'articule autour des trois dimensions de la RCP. Le troisième et dernier article, visant à comprendre comment le processus de RCP se déroule avec des élèves du primaire réalisant une activité de programmation, évoque les résultats découlant de l'utilisation de la grille d'observation de la RCP. Il est notamment question dans cet article de comprendre quelles composantes des trois dimensions de la RCP sont mises en œuvre par les élèves et de quelle façon cela influence la situation de résolution de problèmes en collaboration. Enfin, la conclusion de ce manuscrit présente une synthèse de ces trois articles, de même qu'une synthèse générale de la recherche, les implications pratiques, ainsi que des pistes de recherche futures.

Chapitre 1 - Problématique

Selon Boyer et Martineau (2018), la problématique est fondamentalement argumentative et est « l'aboutissement du processus de compréhension du monde par le chercheur » (p.90), où doit être défendue la pertinence de la recherche, tant pour le milieu scientifique (pertinence scientifique) que pour l'éducation (pertinence sociale). Nous présentons donc dans ce chapitre le processus ayant mené à la formulation de l'objectif général de recherche. D'abord, nous abordons le numérique et son omniprésence, puis nous proposons une description sommaire de la programmation nécessaire à la compréhension de la problématique. Il est ensuite question de l'utilisation de la programmation à l'école, qui est déclinée de façon à présenter l'origine de ce phénomène, l'état de la situation à l'échelle internationale, nationale et provinciale, ainsi que les outils pouvant être utilisés à l'école et les effets potentiels sur l'apprentissage. La présentation graduelle et logique de ces informations permet d'aboutir à la formulation de notre objectif général de recherche et à une argumentation quant à la pertinence de mener une telle recherche.

1.1 L'omniprésence technologique

Que ce soit à la maison ou à l'école, pour le plaisir ou le travail, les appareils électroniques permettent d'effectuer des tâches qui seraient d'une complexité considérable autrement. L'ordinateur en est d'ailleurs un exemple évocateur. À travers les pays de l'Organisation de coopération et de développement économiques (OCDE), 72 % des élèves mentionnent une utilisation de l'ordinateur à l'école et 93 % des élèves disent utiliser l'ordinateur à la maison (OCDE, 2015b). Comme ces résultats sont basés sur l'étude PISA 2012 et qu'il ne semble pas exister de données correspondantes plus récentes de l'OCDE, il est raisonnable de penser que ces taux sont fort probablement plus élevés aujourd'hui. Similairement, l'utilisation des téléphones cellulaires, au cours des dernières années, s'est développée en un phénomène majeur et ne connaissant pas de frontière. En plus de leur principale utilité, ces téléphones permettent de consulter de l'information, de créer des contenus numériques et d'accomplir des centaines d'autres tâches variées. Au Québec, le taux d'adoption de ces appareils confirme leur omniprésence : le Centre facilitant la recherche et l'innovation dans les organisations (CEFRIO)

constate qu'en 2019, 77 % des adultes québécois possédaient un téléphone intelligent (2020). Les données décrivent une réalité singulière chez les Québécois de 18 à 24 ans, pour qui le taux d'adoption de téléphones intelligents s'élève à un taux de 98%.

Bien que l'utilisation de ces appareils soit généralisée dans la vie de tous les jours, leur usage à l'école, en revanche, suscite des questionnements (Brigaudeau, 2018; Maisonneuve et Lemieux, 2018; Morasse, 2018; Pineau, 2018; Vigneault, 2018). Ces remises en question émanent notamment d'une décision du Gouvernement français d'interdire l'utilisation des téléphones intelligents dans l'enceinte des écoles et collèges (Blanquer, 2018). Cette décision fut justifiée par l'évocation des effets nuisibles de ces appareils tant sur la qualité d'écoute et de concentration que sur la qualité de la vie collective, de même que par leur propension à favoriser le cyberharcèlement et l'accès à du contenu inapproprié (Ministère de l'Éducation nationale, 2018). Bien que ces arguments s'appuient sur la littérature scientifique (voir notamment De-Sola Gutiérrez et al., 2016; Société canadienne de pédiatrie, 2018), les moyens législatifs mis en œuvre ont fait réagir plusieurs chercheurs de la francophonie qui insistent plutôt sur l'importance d'enseigner aux jeunes comment se servir de tels appareils, et ce, de façon responsable et critique (Allaire, 2017; Karsenti, 2018). Bref, le cas du téléphone intelligent est un exemple parmi d'autres que nous utilisons pour faire état de cette omniprésence numérique non seulement à la maison, mais aussi à l'école.

Les propos des chercheurs précités mettent en évidence le rôle central de l'école lorsqu'il est question d'éducation au numérique. Au Québec, dès le début des années 1980, le gouvernement voyait déjà l'intérêt, voire la nécessité, « d'enseigner l'ordinateur » aux élèves en raison de son importance dans la société (Pinet, 1982). Cette préoccupation est demeurée présente au sein du *Programme de formation de l'école québécoise* (PFÉQ), publié en 2006 et toujours en vigueur en 2020, où le ministère faisait état d'une démocratisation des technologies de l'information et de la communication (TIC) ainsi que d'un impératif du milieu du travail pour une certaine maîtrise de la technologie, « à la fois langage et outil » (Gouvernement du Québec, 2006). Cela dit, plus de 14 années après la parution du PFÉQ, les propos du ministre de l'Éducation de l'époque ne parviennent pas à dépeindre la réalité contemporaine. En 2018, le Ministère de l'Éducation et de l'Enseignement supérieur (MÉES) renouvelle cet intérêt qui n'a cessé d'évoluer en l'encrant

davantage dans la formation scolaire de tous les élèves et étudiants québécois. C'est avec le *Plan d'action numérique en éducation et en enseignement supérieur* (PANÉES) que le MÉES offre de nombreuses balises et mesures concrètes pour encadrer la présence du numérique, dans les salles de classe et dans la société :

« Les progrès fulgurants dans le domaine du numérique [...] confirment la nécessité d'agir afin de répondre aux enjeux technologiques, éthiques et sociaux liés aux innovations qui viendront transformer nos modes de vie, nos milieux de travail et notre quotidien. [...] Le [PANÉES] est guidé par la vision d'une intégration efficace et d'une exploitation optimale du numérique au service de la réussite » (Gouvernement du Québec, 2018, p. 9).

Au Québec, le PANÉES succède à une tradition presque quarantenaire d'éducation au numérique en contexte scolaire. Or, bien qu'il ne s'agisse pas d'un phénomène récent, l'évolution constante des outils numériques oblige continuellement à repenser les façons de faire (Sebire et al., 2018). L'État québécois s'appuie d'ailleurs sur la nécessité de répondre à l'évolution rapide du numérique pour justifier l'application de son plan d'action. Cela amène donc à questionner les façons de mettre en œuvre les pratiques pédagogiques centrées sur l'utilisation d'appareils numériques. Incidemment, ce questionnement met en lumière les différents dispositifs numériques accessibles dans les établissements d'enseignement.

Nous avons choisi de diriger notre intérêt de recherche sur la programmation, qui est l'une des façons d'enseigner à l'aide du numérique. Dans la pratique, au Québec, la programmation est déjà implantée dans plusieurs classes sans toutefois qu'il y ait de ligne directrice claire à cet égard dans le PFÉQ quant aux contenus, méthodes et outils utilisés (Gouvernement du Québec, 2018). Il revient donc aux enseignants de choisir les outils utilisés pour éduquer au numérique, étant donné que cet apprentissage est facultatif et non prescrit (Heitink et al., 2016; Stockless et al., 2018). C'est d'ailleurs pourquoi les attitudes et les croyances des enseignants envers le numérique seraient des facteurs majeurs influençant son intégration dans l'enseignement (Farjon et al., 2018; Julio et al., 2018). Il est toutefois à noter que le gouvernement du Québec a récemment démontré à quelques reprises son intention de favoriser l'implantation de la programmation informatique dans les pratiques pédagogiques des enseignants, notamment avec le *Cadre de référence de la compétence numérique* (Ministère de l'Éducation et de l'Enseignement supérieur, 2019) et avec le guide *L'usage pédagogique de la programmation informatique*

(Ministère de l'Éducation, 2020a), que nous évoquerons dans une section subséquente de ce chapitre où nous traçons un portrait plus détaillé de la programmation dans les écoles du Québec.

1.2 La programmation : un langage informatique

La programmation est définie comme étant l'action d'écrire, à l'aide d'un langage informatique, une série d'actions qui sont interprétées puis exécutées par un ordinateur (Blackwell, 2002). Il existe une multitude de langages informatiques, tels que *Python* et *C++*. Bien que tous ces langages aient leurs particularités, ils ont néanmoins la même utilité : commander une série d'actions. Les langages de programmation peuvent avoir différentes formes. La forme textuelle est la plus commune. Or, elle peut aussi prendre la forme de boîtes ou d'éléments visuels (figure 1).



Figure 1. – Programmation sous la forme de blocs²

Ces boîtes, qui sont des représentations visuelles contenant un code écrit, peuvent être assemblées par juxtaposition ou à l'aide de liens (fils, flèches, etc.). La présence de symboles ou d'images sur les boîtes peut aider le programmeur novice à comprendre rapidement l'utilité et la fonction de la boîte (du code). Quel que soit le type de langage, le processus de programmation est une occasion propice à la mobilisation de plusieurs processus cognitifs. Romero (2016b) suggère d'ailleurs que le fait de comprendre ce processus permet non seulement de changer le

² Photo : Cimon Parent.

rapport qu'entretiennent les personnes avec le numérique, mais aussi de les rendre plus critiques quant à l'utilisation qu'ils en font. En effet, en comprenant mieux comment fonctionnent les différents outils et appareils numériques, il devient plus aisé de les utiliser de façon judicieuse et appropriée.

Par ailleurs, l'utilisation de la programmation avec les enfants est souvent conjuguée à l'utilisation d'appareils robotiques (Komis et Misirli, 2011; Sullivan, 2008). La programmation devient ainsi une façon d'animer un robot et de lui faire exécuter diverses tâches. Cette union entre la programmation et la robotique occupe une place importante dans cette recherche.

1.3 La programmation en contexte scolaire

À la suite de plusieurs auteurs, nous avançons que la programmation peut être apprise puis utilisée à l'école, cette dernière étant un lieu propice à l'enseignement balisé et orienté sur le développement d'habiletés et de savoir-faire (Depover et al., 2007; Komis et Misirli, 2011; Romero, 2016b). Cette section fait état, d'abord, de la genèse du phénomène de programmation à l'école, puis de la situation internationale et canadienne à cet égard.

1.3.1 Genèse du phénomène et implications pratiques

Les premières publications scientifiques s'intéressant à l'utilisation de la programmation en contexte scolaire avec des élèves du primaire dans la littérature que nous avons consultée remontent à la fin des années 1960. Dans un rapport soumis par Feurzeig, Papert, Bloom, Grant et Solomon (1969) à la *National Science Foundation* (États-Unis), les auteurs présentent les résultats d'un projet impliquant l'usage de la programmation en tant que cadre conceptuel et opérationnel pour l'enseignement des mathématiques au primaire. Menée auprès d'élèves de 7 à 13 ans (2^e année et plus), cette étude concluait qu'il était effectivement possible d'enseigner la programmation à des élèves de cet âge et qu'il était également possible de l'utiliser en tant qu'outil pour l'apprentissage de différents concepts mathématiques, tout en évoquant de possibles « effets secondaires » (*side effects*) sur d'autres habiletés « extramathématiques » (*extra-mathematical*), comme la lecture (Feurzeig et al., 1969). Les réflexions ayant mené à la réalisation d'une telle étude reposent sur les potentialités de la programmation en tant qu'outil

dont la complexité et la formalité internes permettent aux élèves de développer certaines habiletés ou compétences :

« Students are accustomed to using language and logic in the context of a sympathetic listener who makes reasonable interpretations of their statements, and is tolerant of the gaps in their arguments. [...] One would like to find areas of mathematical work in which students would impose the need for precise articulation on themselves. We believe that such areas can be created by appropriate instruction in the use of computer and programming languages » (Feurzeig et al., 1969, p. 5 et ss.).

La rigueur et la systématisation inhérentes à la programmation sont donc les éléments-clés à l'origine de son potentiel pédagogique. Or, la quasi-totalité des langages de programmation n'était pas accessible ou pertinente pour des élèves du primaire (Stoutemyer, 1979)³.

La nécessité de rendre la programmation accessible aux enfants a soulevé plusieurs questions, que nous réunissons sous la question globale suivante : comment amener des élèves de huit ans à faire de la programmation à l'aide d'un langage qui, bien que simplifié, puisse permettre de construire des programmes dont la complexité interne n'est pas diminuée? Parmi les principaux langages accessibles à de jeunes élèves on évoque, dans la littérature, le APL (*A Programming Language*), développé par Kenneth Iverson de la compagnie américaine IBM. En plus d'être accessible à des élèves du primaire, il est associé à l'approche « boîte de verre » de l'informatique (en opposition à la boîte noire) puisqu'il permet à l'élève de développer sa compréhension du fonctionnement interne de l'ordinateur, ce dernier devenant « transparent » à ses yeux (Peelle, 1974). Il en va de même pour les langages FOCAL ou BASIC, qui ont aussi été utilisés auprès d'élèves du primaire, mais qui seraient plus appropriés pour les adolescents et les adultes, en comparaison au langage LOGO (Holzman et Glaser, 1977). Enfin, ce langage LOGO, créé par Feurzeig et Papert en 1968, a été utilisé dans de nombreuses études portant sur l'utilisation de la programmation au primaire (voir notamment Clements et Gullo, 1984; Gorman et Bourne, 1983; Khayrallah et Van den Meiraker, 1987; Miller et al., 1988). Dès le début des années 1980, des chercheurs et étudiants du Massachusetts Institute of Technology (MIT) mettaient sur pied le *Projet LOGO* : un environnement permettant aux enfants d'interagir avec les ordinateurs à l'aide

³ Stoutemyer (1979) va même jusqu'à dire qu'**aucun** langage de programmation communément enseigné à l'époque n'était utilisable pour l'apprentissage des mathématiques, et ce, du primaire au collège.

d'un dispositif adapté à leur âge. En utilisant le langage informatique simple LOGO, les enfants étaient en mesure de programmer des séquences animant un dispositif robotique ayant la forme d'une tortue. On voulait offrir aux élèves un outil qui leur permette non seulement d'apprendre à utiliser un ordinateur, mais surtout de faire en sorte qu'ils apprennent les rudiments de la programmation, tout en aspirant à affecter positivement leur façon d'apprendre dans toutes les matières (Papert, 1981). Il n'était pas question d'utiliser l'ordinateur comme une fin en soi, mais plutôt de l'utiliser comme un outil qui puisse maintenir les enfants au centre de leurs propres apprentissages (Bossuet, 1982). Le projet LOGO a d'ailleurs été mis à l'essai au Québec, où on a vanté son caractère transdisciplinaire, son haut potentiel pour la résolution de problèmes en équipe ainsi que sa capacité à faire mobiliser aux élèves leur métacognition (*Les logiciels éducatifs*, 1984).

Selon Papert, LOGO n'est pas qu'un outil et un langage, mais également une philosophie, cette dernière exigeant beaucoup plus d'efforts à adopter puisqu'un changement des façons de faire demande beaucoup plus de temps que la simple implantation d'un nouvel outil (Papert, 1980). Les propos de Papert ont été critiqués par certains auteurs dont Baron et Drot-Delange (2016), qui attribuent un caractère « parfois proche de celui de l'évangéliste » (p.43) à son style d'écriture. Ces derniers évoquent d'ailleurs les propos de Crahay (1987), qui parlait d'une argumentation « triomphaliste » (p.44) et d'une « rhétorique militante » (p.45) présentes dans les travaux de Papert.

Malgré l'absence d'intégration massive dans les écoles, le projet LOGO a tout de même été le point de départ de ce qui allait devenir une tendance pédagogique. Cette dernière s'est ensuite transformée avec l'apparition de robots programmables plus évolués et pouvant être construits par l'apprenant : la programmation est alors devenue une façon d'animer un dispositif, rendant son utilisation encore plus concrète et surtout plus motivante.

1.3.2 Une conjoncture internationale

Afin de répondre aux impératifs technologiques contemporains, notamment en ce qui concerne le marché du travail, plusieurs pays à travers le monde ont choisi de mettre en place des initiatives visant à favoriser l'apprentissage de la programmation chez les élèves (Schleicher, 2019). Alors

que certains gouvernements ont choisi d'introduire ce nouveau contenu au programme de formation national, d'autres ont plutôt choisi de financer des initiatives particulières.

En Finlande, dans le programme d'enseignement du niveau *comprehensive* (équivalent au primaire et secondaire du Québec), on compte sept compétences générales parmi lesquelles figure la compétence relative au numérique. Elle se décline en plusieurs catégories, au sein desquelles on retrouve la pensée computationnelle et la programmation (Toikkanen, 2015). L'enseignement de la programmation est ainsi obligatoire tant au niveau primaire que secondaire. Les raisons qui motivent cette intégration sont le développement des compétences de pensée logique et de programmation, tout en favorisant à long terme l'employabilité dans le secteur des technologies numériques (Balanskat et Engelhardt, 2014).

Il en va de même au Royaume-Uni, où le parcours typique d'un élève, au niveau primaire et secondaire, est séparé en quatre étapes principales appelées *Key Stages*. Dans le curriculum national, l'apprentissage de la programmation est considéré, tout au long de ces quatre *Key Stages*, comme un outil de développement de la créativité et de la pensée computationnelle (Department for Education, 2013). Dès le premier *Key Stage*, c'est-à-dire à l'âge de cinq à sept ans, les élèves sont initiés à la programmation : il s'agit d'un enseignement formel ayant pour but d'amener les élèves à comprendre ce qu'est un algorithme et son utilisation dans un programme. Ils sont aussi appelés à créer et déboguer des programmes simples, puis à utiliser un raisonnement logique pour anticiper le fonctionnement d'un programme (Department for Education, 2013). Le second *Key Stage* concerne les élèves âgés de 7 à 11 ans, ce qui représente approximativement le secteur primaire dans le système scolaire québécois. Les attentes pour ce niveau sont de l'ordre de la conception, de la sélection et de l'utilisation du raisonnement logique et des algorithmes.

Il s'agit là de deux pays qui se démarquent par leur intégration systématique et nationale de la programmation à l'enseignement primaire et secondaire. Cela dit, d'autres pays ont opté pour des mesures de moins grande envergure, comme le financement d'initiatives favorisant l'intégration de la programmation dans la formation. C'est notamment le cas de l'Estonie, qui s'est doté en 2014 d'un plan stratégique à long terme (2014-2020) intitulé *Estonian Lifelong*

Learning Strategy. L'un des principaux objectifs de ce plan était de faire en sorte que la technologie soit utilisée de façon plus efficace pour l'apprentissage et l'enseignement (Ministry of Education and Research, 2014). C'est pendant cette période qu'un programme nommé *ProgeTiiger* (Prog-Tigre), financé par le ministère estonien de l'Éducation et de la Recherche, fut implanté afin de développer des ressources pédagogiques pour l'utilisation d'outils de programmation variés aux niveaux préscolaire, primaire et secondaire (Information Technology Foundation for Education, 2020). Une multitude de guides et de canevas d'activités ont ainsi été proposés aux enseignants en vue de faciliter l'enseignement de la programmation avec une trentaine d'applications et de dispositifs différents comme *Bee-Bot*, *Dash and Dot*, *MakeyMakey* et *Scratch* (HITSA, 2017). En rétrospective, il est aujourd'hui possible de constater que l'Estonie se place au-dessus de la moyenne quant à la compétence numérique de sa population, en plus d'avoir fait d'importants progrès concernant les ressources d'apprentissages numériques disponibles (OCDE, 2020a).

Plus récemment, l'Australie a aussi emboîté le pas avec son programme *National Innovation and Science Agenda* (Australian Government, 2020), une mesure de plus d'un milliard de dollars australiens dont 64 millions (environ 60 millions en dollars canadiens) ont été alloués à des « early learning and school STEM initiatives ». Parmi ces initiatives, on retrouve par exemple les *Australian Digital Technologies Challenges*, des activités destinées aux élèves de la 3^e à la 8^e année visant à développer des habiletés avec différentes interfaces de programmation, comme *Blockly* et *Scratch*, et différents langages (Python, Javascript, etc.).

Bien que n'étant pas issus d'une recension systématique des initiatives internationales, ces exemples permettent néanmoins d'illustrer de quelle façon et dans quelle mesure la programmation a été, à travers le monde, intégrée à l'apprentissage scolaire d'élèves du primaire. Or, dans plusieurs autres pays, les initiatives pour l'apprentissage de la programmation sont souvent isolées et ne bénéficient pas d'une reconnaissance des instances gouvernementales. Considérant le contexte de cette étude, il convient de tracer un portrait de la situation au Canada et plus spécifiquement au Québec.

1.3.3 La situation canadienne et québécoise

À l’instar de l’Estonie, le gouvernement canadien a opté pour le financement de différentes initiatives à travers le pays. Son programme *CodeCan*, du ministère de l’Innovation, des Sciences et du Développement économique du Canada (ISDE), « a pour objectif d'aider les jeunes Canadiens, notamment ceux appartenant à des groupes traditionnellement sous-représentés, à acquérir les compétences dont ils auront besoin pour se préparer à des études approfondies, notamment des compétences numériques avancées et des cours en STIM débouchant sur les emplois de demain » (ISDE, 2019). Au cours de ses deux premières années, le programme a rejoint 1,3 million d’élèves et 61 000 enseignants en finançant des dizaines d’initiatives. Parmi ces dernières, on retrouve par exemple *Déjouez les codes!*, qui propose des cours et ateliers sur la programmation, et *Black Boys Code*, qui permet à de jeunes garçons du primaire et du secondaire de développer des compétences en programmation. Par ailleurs, déjà en 2016, des milliers d’écoles canadiennes de niveau primaire, secondaire, collégial et même universitaire ont participé au mouvement international *#heureducode*, qui vise à initier les élèves et les étudiants à la programmation (Code.org, 2016).

Publié récemment, le *Cadre de référence pancanadien pour l’enseignement de l’informatique* (Canada Learning Code, 2020) place la programmation au centre des contenus abordés, sans toutefois s’y limiter. Selon ce document, à la fin du secondaire, les élèves canadiens devraient notamment pouvoir « concevoir un algorithme à l’aide d’une combinaison d’éléments informatiques, notamment les fonctions, les objets, les expressions conditionnelles et les tableaux » (p.22). Or, comme l’éducation relève des provinces et territoires au Canada, il existe de nombreuses disparités. Selon ce cadre de référence, 7 provinces et territoires sur 13 ont intégré les compétences et habiletés en informatique au programme du primaire et du secondaire, ce qui n’implique pas forcément la programmation.

Par exemple, en Ontario, le nouveau programme-cadre de mathématiques pour le primaire⁴ inclut l’apprentissage de la programmation dès la première année pour « renforcer les habiletés de résolution de problèmes et développer l’aisance en technologie » (Gouvernement de

⁴ Programme-cadre pour les élèves de la 1^{re} à la 8^e année en Ontario.

l'Ontario, 2020). En Colombie-Britannique, le gouvernement annonçait, au mois de juin 2016, un investissement de six millions de dollars pour supporter la mise en place d'un nouveau programme de formation accordant une grande importance à la programmation et à l'ordinateur en classe (Gouvernement de la Colombie-Britannique, 2016). On explique cette décision par leur potentiel quant au développement de la pensée critique et des compétences en résolution de problème (Smart, 2016).

Puis, au Québec, le ministère de l'Éducation et de l'Enseignement supérieur a lancé, en 2019, le *Cadre de référence de la compétence numérique* (CRCN) (Ministère de l'Éducation et de l'Enseignement supérieur, 2019), une mesure phare de son *Plan d'action numérique* (Gouvernement du Québec, 2018), lancé en 2018 et visant à valoriser l'utilisation du numérique pour l'apprentissage ainsi que le développement de la compétence numérique des jeunes et des adultes. Parmi les dimensions de la compétence numérique, la seconde est intitulée *Développer et mobiliser ses habiletés technologiques* et est principalement axée sur la programmation, l'intelligence artificielle et tout phénomène émergent lié au numérique. En effet, cette dimension met l'accent sur le développement de la pensée informatique et la programmation y est mentionnée comme l'une des principales façons d'y parvenir (Ministère de l'Éducation et de l'Enseignement supérieur, 2019, p. 14). C'est d'ailleurs ce que propose la mesure 2 du *Plan d'action numérique* : « l'utilisation de la programmation informatique à des fins pédagogiques et didactiques pour soutenir les élèves dans la réalisation des apprentissages et le développement des compétences prévues au PFÉQ » (Gouvernement du Québec, 2018, p. 27). Puisqu'il s'agit encore, au moment d'écrire cette thèse, d'une mesure récente et considérant le contexte de pandémie de COVID-19 ayant provoqué une réorganisation des priorités dans le milieu de l'éducation, nous croyons que les véritables concrétisations de cette mesure apparaîtront graduellement au cours des prochaines années (activités en classes, programmes dans les écoles, etc.). En revanche, antérieurement au CRCN, certains projets ont été lancés pour intégrer la programmation et la robotique dans les écoles primaires. On retrouve notamment le projet *Code Mtl*, piloté par la Fondation de la Commission scolaire de Montréal (CSDM), qui permet à plus de 6 400 élèves de 5 à 12 ans provenant de 95 écoles d'être initiés à la programmation à l'école (Fondation de la Commission scolaire de Montréal, 2020). De plus, en janvier 2018, le MÉES a

annoncé le lancement du projet pilote *Robots 360* à la CSDM. Ce projet « vise à explorer des solutions et à soutenir le milieu scolaire dans l'utilisation de la programmation informatique en classe comme outil pédagogique pour favoriser le développement de la pensée informatique et l'atteinte de la réussite éducative des élèves » (Ministère de l'Éducation et de l'Enseignement supérieur, 2018).

Le gouvernement du Québec a réitéré son intérêt pour la programmation, d'abord exprimé dans le PANÉES et le CRCN, en publiant un guide intitulé *L'usage pédagogique de la programmation informatique* (Ministère de l'Éducation, 2020a). Ce dernier présente la programmation de façon générale et simplifiée, puis offre des pistes d'utilisation aux enseignants qui souhaitent introduire la programmation dans leur enseignement. Le ministère de l'Éducation a également publié une version mise à jour du *Référentiel des compétences professionnelles* de la profession enseignante, où les dimensions de la compétence numérique présentées dans le CRCN sont partie intégrante de la douzième compétence « Mobiliser le numérique » (Ministère de l'Éducation, 2020b). Cela permettra aux enseignants de développer, dès la formation initiale universitaire, des connaissances et des méthodes relatives à l'utilisation du numérique – dont la programmation – dans leur pratique.

Par ailleurs, ce tour d'horizon international met en relief la place de la programmation dans les programmes de formation. S'agit-il d'un contenu à enseigner dans le cadre d'une matière (par ex. : l'informatique) ou d'une compétence particulière, comme c'est le cas en Finlande ou au Royaume-Uni? Ou s'agit-il plutôt d'un contenu transversal qui n'est pas défini en termes d'attentes observables ou de critères d'évaluation, comme c'est le cas en Estonie et au Québec? Dans ce second cas de figure, où il s'agit de la responsabilité de tout le monde et de personne à la fois, de nombreuses questions peuvent être soulevées quant à la véritable utilisation de la programmation dans les salles de classe.

Malgré cela, on constate néanmoins que l'apprentissage de l'informatique, et plus précisément de la programmation, est au centre d'orientations pédagogiques et politiques à l'échelle internationale. Bien que la perception de l'importance de la programmation n'est pas forcément synonyme de sa nécessité, elle démontre néanmoins qu'il s'agit d'un phénomène mondial, que

le Québec et le Canada sont dans cette conjoncture, et qu'il y aurait avantage à être conscient des aspects inhérents à l'utilisation de la programmation en contexte scolaire. Ce constat suscite une réflexion quant à l'usage pédagogique de la programmation dans le but d'améliorer les apprentissages. Pour mieux comprendre ce qui peut être fait, il faut d'abord savoir ce qui a été mis à l'essai en contexte scolaire.

1.4 Les outils principaux : des plateformes en ligne aux dispositifs robotiques

Pour comprendre le rôle de la programmation en contexte scolaire, il est nécessaire de connaître les principaux outils disponibles et leur fonctionnement. L'historique des outils majeurs en la matière débute avec Resnick et d'autres chercheurs du MIT qui ont mis au point, entre 2003 et 2007, une interface appelée *Scratch*, permettant aux enfants de se familiariser avec la programmation par blocs ainsi qu'avec les concepts associés à la programmation en général (Wilson et al., 2013). L'avancement de la technologie et l'accessibilité à des ordinateurs ont certainement contribué au développement et à l'essor de cet outil et des autres qui ont suivi. Resnick et son équipe mentionnent un apport de l'application Web sur le plan de l'apprentissage de concepts informatiques, de la pensée créative, du raisonnement systématique et du travail collaboratif (Resnick et al., 2009). Son utilisation est aussi liée à une augmentation de la motivation et de l'intérêt des élèves au primaire (Wilson et Moffat, 2010). Les plateformes Web sont l'exemple le plus facilement transférable lorsqu'il est question de l'utilisation de la programmation à l'école : étant très accessibles, adaptées et n'engendrant que peu (voire pas) de dépenses, elles permettent d'aborder la programmation en classe sans que l'enseignant soit un programmeur aguerri.

Les grandes compagnies technologiques comme Apple, Google et Microsoft participent à l'essor de la programmation accessible en proposant des programmes ou des outils (souvent gratuits) ayant pour but d'accompagner les élèves, les enseignants et les citoyens en général dans leur découverte de la programmation. Par exemple, Apple propose son application *Swift Playground*, qui permet aux enfants d'apprendre à programmer de façon ludique, ainsi qu'une série de livres numériques présentant des activités pédagogiques aux enseignants. Google, avec son

programme *CS First*, a rendu disponible une série d'activités pédagogiques, accessibles aux élèves et aux enseignants, pouvant être effectuées à l'aide de *Scratch*. Microsoft offre quant à elle une plateforme nommée *Makecode*, dans laquelle les élèves peuvent programmer à l'aide de blocs ou de texte.

En revanche, l'utilisation d'une plateforme de programmation n'est pas synonyme de mobilisation, voire de développement, de compétences (Papert, 1980, 1994). Le souci du maintien d'un enseignement pertinent demeure. On décèle ici l'intérêt de bien comprendre les diverses façons d'utiliser la programmation en classe. Il en va de même pour les petits robots, accessibles au grand public, qui offrent une autre occasion d'utiliser la programmation de façon ludique. Ces robots, attrayants et relativement abordables, sont adoptés dans plusieurs écoles primaires par des enseignants qui désirent des solutions clé en main. La robotique s'est d'ailleurs fait connaître avec la compagnie LEGO, créatrice des petits blocs emboîtables multicolores. Lors des activités de robotique, ce sont les élèves qui programment le robot, et parfois même qui le construisent, afin de lui faire accomplir certaines tâches. Mosley et Kline (2006) mentionnent l'apport des *LEGO Robotics* en matière d'engagement social et d'apprentissage de l'informatique. De même, Pullan (2013) soutient que la robotique contribue à une meilleure compréhension de la programmation, ce qui appuie notre choix d'opter pour une vision intégratrice de ces deux outils dans le cadre de cette recherche. Resnick (2007) a quant à lui vérifié l'utilisation d'un robot miniature, le *Cricket*, pouvant être programmé par les enfants. En plus de développer la pensée créative, nécessaire à la résolution de problèmes du quotidien, cet outil s'intégrait bien dans la « l'approche de l'apprentissage à l'éducation préscolaire »⁵ : imaginer, créer, jouer, partager et réfléchir (Resnick, 2007, p. 2). La tangibilité des robots programmables est donc un facteur majeur à considérer en songeant à la programmation dite pédagogique (Mubin et al., 2013).

1.5 Effets de l'utilisation de la programmation sur l'apprentissage

Afin d'appuyer ou de nuancer les constats soulevés dans les précédentes sections, nous avons procédé à une recension approfondie de la littérature scientifique. Dans la présente section sont mis de l'avant les travaux ayant contribué à une meilleure compréhension des effets de la

⁵ Traduction libre (« *the kindergarten approach to learning* »).

programmation et de la robotique pédagogique, et ce, de l'université – où l'utilisation de ces outils pourrait sembler la plus opportune – jusqu'au niveau préscolaire, où leur utilisation est beaucoup moins répandue.

D'abord, en contexte universitaire, l'utilisation de la programmation à des fins pédagogiques a été étudiée par Hromkovic, Kohn, Komm et Serafini (2016a) auprès d'étudiants en biologie. Le langage LOGO (Papert, 1980) leur a été enseigné, ce qui a permis d'observer l'apprentissage de concepts fondamentaux en programmation chez des apprenants n'ayant aucune connaissance antérieure en la matière. L'utilisation subséquente du langage Python visait à préparer les étudiants aux tâches concrètes associées spécifiquement au domaine de la biologie. Les conclusions des chercheurs indiquent que l'apprentissage des concepts fondamentaux de programmation avec LOGO a été d'une grande valeur pour les étudiants, puisqu'ils ont par la suite démontré une compétence globale intéressante : « they were able to apply many methods and paradigms to more complex algorithms » (Hromkovič et al., 2016a, p. 164). Ils insistent d'ailleurs sur l'importance d'un apprentissage de la programmation non seulement pour la connaissance d'un langage informatique (objet), mais surtout dans une perspective de développement de compétences générales (outil).

Une autre étude, menée par Özüorçun et Bicen (2017) auprès de 100 étudiants à la faculté d'ingénierie de la *European University of Lefke*, à Chypre, a mis l'accent sur la compétence de résolution de problèmes. Le dispositif d'étude était constitué d'un processus en quatre étapes : l'analyse d'une situation-problème, la conception d'un algorithme pour la résoudre, l'élaboration d'un pseudo-code et d'un ordinogramme⁶, pour enfin terminer avec l'application et la mise à l'essai du programme avec un ordinateur. Lors du processus de RP à l'aide de la programmation tangible⁷, les futurs ingénieurs ont démontré de façon significative un développement de compétences en créativité, en collaboration et en pensée analytique (Özüorçun et Bicen, 2017). Nous pouvons faire deux constats à la lumière des conclusions de ces études : (1) la programmation peut avoir des effets positifs sur le développement de compétences transversales

⁶ L'ordinogramme est un organigramme présentant la structure d'un programme informatique.

⁷ La programmation tangible implique l'utilisation concomitante d'un dispositif robotique (dans ce cas-ci, il s'agit des *Lego Mindstorm EV3*). Ce concept est décliné dans la section 2.1.2.2. de ce manuscrit.

et complexes, et (2) l'activité de programmation ne se limite jamais simplement à l'acte d'écrire des lignes de texte. Bien au contraire, l'accent est davantage mis sur le processus de conception et de structuration du programme, en amont, sans nécessiter l'utilisation de l'ordinateur. Cette pratique a d'ailleurs été appliquée pour le développement de la pensée algorithmique (PA) à l'université par Futschek et Moschitz (2010), qui suggèrent de « jouer les algorithmes », c'est-à-dire de les rendre concrets, un peu à la façon d'une pièce de théâtre où chaque étudiant se voit attribuer un rôle (une variable) dans l'algorithme. Il s'agit d'une méthode qui pourrait certainement être appliquée à tous les niveaux d'enseignement.

Pour ce qui est de l'utilisation de la programmation à l'enseignement secondaire, le nombre d'articles scientifiques est plutôt limité. En effet, il est important de considérer le caractère interordre de plusieurs études portant sur la programmation en contexte pédagogique : plusieurs d'entre elles s'intéressent aux niveaux *K-12*, un terme fréquent dans la littérature anglophone pour désigner les niveaux du préscolaire à la fin du secondaire (préscolaire à la 12^e année). Plusieurs de ces articles abordent le développement de la pensée informatique (PI) aux niveaux *K-12* (Atmatzidou et Demetriadis, 2016; Barr et Stephenson, 2011; Caspersen et Nowack, 2013; Grover et Pea, 2013; Lu et Fletcher, 2009; Sengupta et al., 2013; Settle et al., 2012; Tatar et al., 2017; Touretzky et al., 2013), mais aucun ne traite spécifiquement de la programmation. La littérature limitée portant sur la programmation a d'ailleurs été soulevée dans la revue de littérature de Lye et Koh (2014), qui n'ont réussi à identifier que neuf articles revus par les pairs sur la programmation aux niveaux *K-12*, et la plupart portaient sur des activités parascolaires pour lesquelles les élèves étaient soit volontaires, soit sélectionnés spécifiquement. Cela implique donc, selon eux, un besoin indispensable de recherches dans ce champ spécifique (Lye et Koh, 2014).

En ce qui concerne le niveau secondaire exclusivement, une revue de la littérature orientée sur le *Pedagogical Content Knowledge* (PCK), de Saeli, Perrenet, Jochems et Zwaneveld (2011), a soulevé que l'une des principales raisons d'enseigner la programmation au secondaire était le développement de compétences en RP chez les élèves, et ce, via l'enseignement de connaissances et de stratégies inhérentes à l'activité de programmation. Les travaux de Bugmann et Karsenti (2018a) tendent à confirmer ces résultats, et ce, dans le cadre d'une étude

exploratoire menée auprès de 79 élèves en adaptation scolaire, âgés de 12 à 18 ans, qui ont été appelés à programmer un robot humanoïde en collaboration avec leurs pairs. Par ailleurs, lorsqu'il est question de l'intégration de la programmation à l'école secondaire, cette dernière semble pouvoir être justifiée par le développement potentiel de la PI, qui peut engendrer à son tour de « nombreux impacts éducatifs positifs » (Israel et al., 2015). À ce sujet, Hromkovič et al. (2016b) considèrent la programmation comme étant une occasion d'enseigner la pensée algorithmique et l'informatique aux élèves, non pas dans l'ambition d'une maîtrise d'un langage informatique, mais plutôt dans le but de contribuer à la compréhension des fondements de l'informatique et de la programmation.

Enfin, pour ce qui est de l'enseignement préscolaire et primaire, une étude de cas réalisée auprès de 10 élèves du préscolaire âgés de 5 à 6 ans indique un développement de concepts mathématiques, de même qu'un développement d'habiletés sociales et de résolution de problème (Fessakis et al., 2013). Le dispositif mis en place consistait en une interface de programmation, basée sur le langage LOGO, où les élèves devaient faire progresser une coccinelle (librement ou dans un labyrinthe) à l'aide de commandes visuelles (p. ex., flèches vers l'avant, vers la droite, etc.). Les auteurs mentionnent que les élèves ont participé activement aux activités proposées et qu'ils ont pu développer leurs habiletés de communication et de collaboration dans un contexte d'interaction authentique. Dans une étude expérimentale menée auprès de 65 élèves de 4^e et 5^e année à Madrid, Moreno-León et Robles (2015) ont utilisé la programmation comme outil d'apprentissage en classe d'anglais. Ils ont alors pu observer une amélioration des résultats scolaires, de même qu'une amélioration perçue, chez les élèves, de leur compétence en collaboration et de leur métacognition.

Selon Serafini (2011), la programmation sur ordinateur est l'outil principal pour développer la pensée computationnelle⁸, rendant ainsi son apprentissage dès le primaire une nécessité. Or, Lu

⁸ Wing Wing, J. M. (2010). *Computational Thinking: What and Why?* <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why> définit la pensée computationnelle comme « le processus de pensée qui consiste à formuler les problèmes et leurs solutions de manière à ce que les solutions soient représentées sous une forme qui puisse être efficacement réalisée par un agent de traitement de l'information » (traduction libre : « *Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent* »).

et Fletcher (2009) soulignent que la programmation ne devrait pas devenir le synonyme de la pensée computationnelle. En effet, bien que la programmation sur ordinateur soit un outil efficace pour l'apprentissage de compétences en résolution de problème (Barr et Stephenson, 2011; Béziat, 2012; Komis et Misirli, 2013; Lai et Yang, 2011), il existe des méthodes permettant de développer les compétences visées sans toutefois faire appel au langage informatique textuel, ni même à l'ordinateur (Académie des sciences, 2013; Futschek et Moschitz, 2011). Ces méthodes font allusion à l'enseignement de la programmation déconnectée (*unplugged*), c'est-à-dire sans l'utilisation de quelconque appareil technologique. Par exemple, il est possible d'initier des élèves à la notion d'algorithme en leur demandant de penser à une série d'étapes permettant à un personnage de sortir d'un labyrinthe dessiné au tableau. Il est suggéré que la programmation déconnectée (*unplugged*), à l'aide de blocs de bois par exemple, serait efficace (Conde et al., 2017), voire plus efficace qu'une programmation sur écran en ce qui concerne l'intérêt des élèves et la collaboration (Horn et al., 2009). Il est donc souhaitable de considérer la programmation comme étant non seulement le vecteur d'un apprentissage de compétences technologiques, mais surtout de compétences d'ordre général, aussi appelées compétences transversales⁹. Ces dernières ont un rôle majeur dans le développement et la scolarité des enfants sans nécessairement être rattachées à l'informatique. La technologie est donc un outil facilitant l'apprentissage et non pas forcément l'objet de l'apprentissage. Ce principe est à la base de deux constats fondamentaux dont nous tenons compte : 1) le terme « programmation » a un sens large, qui inclut tant son utilisation sur ordinateur que de façon déconnectée, et 2) la programmation et la robotique sont à la fois des objets d'apprentissage (apprendre à utiliser les outils) et des outils d'apprentissage (utiliser les outils pour apprendre).

En somme, la programmation jumelée à l'utilisation d'un robot peut être utilisée pour l'enseignement en contexte pédagogique. Que ce soit au primaire ou à l'université, la programmation peut être utilisée pour contribuer au développement de compétences transversales, qui contribuent à leur tour aux apprentissages des apprenants à travers les différentes matières du programme. Comme mentionné précédemment, les élèves d'aujourd'hui apprennent et se développent dans un contexte bien différent de celui des précédentes

⁹ Nous explicitons le concept de compétence transversale à la section 2.2. de ce manuscrit.

générations. Il s'avère que la programmation — à la fois langage et outil de la pensée — peut avoir un certain potentiel technopédagogique. Nous retenons le rôle qu'occupe la technologie non seulement dans les programmes de formation, mais fondamentalement, dans la façon même d'apprendre des élèves. Ce constat explique sans doute de la façon la plus explicite l'intérêt de cette recherche, c'est-à-dire la nécessité de porter un regard contemporain et imprégné des nouvelles tendances technopédagogiques sur ce phénomène d'envergure internationale. En effet, nous nous retrouvons devant un intérêt croissant pour un domaine relativement nouveau qui évolue rapidement. Les enseignants semblent trouver tant bien que mal des façons d'utiliser ces outils, sans nécessairement avoir de balises solides pour en encadrer l'utilisation et pour favoriser leur potentiel pédagogique.

1.6 L'objectif général de recherche

Dans ce chapitre, il a été exprimé à maintes reprises que la programmation permettrait de mobiliser et potentiellement développer des compétences en résolution de problème, en raisonnement logique, en pensée critique et en collaboration. L'objectif général de cette recherche s'inscrit dans cette réflexion : nous souhaitons décrire et comprendre les impacts de la programmation informatique sur la mobilisation de compétences d'élèves du primaire. Plus précisément, nous nous concentrons sur deux de ces compétences, soit la résolution de problèmes et la collaboration. Ce choix s'explique tant par le devoir de concision qu'exige l'entreprise d'un projet de recherche doctoral que par la relation, significative et largement documentée, unissant la résolution de problèmes à la collaboration (Hesse et al., 2015; Kamga et al., 2017; OCDE, 2017), de même que la récurrence de la compétence en RP et de la collaboration dans la littérature scientifique relative à la programmation en contexte scolaire (voir *Chapitre 2*).

1.7 La pertinence de cette recherche

Nous établissons la pertinence de notre projet de recherche selon deux axes : un axe scientifique et un axe social. Plusieurs auteurs ont étudié l'utilisation de différents dispositifs avec des élèves du primaire. Par exemple, certains ont documenté les représentations d'élèves de maternelle d'opérations élémentaires de programmation tangible pour faire déplacer le robot BeeBot (Komis et Misirli, 2011), et d'autres ont étudié empiriquement le lien entre la pensée informatique

d'élèves et la programmation tangible (Chen et al., 2017; Noh et Lee, 2020). Plus récemment, des auteurs ont observé les habiletés associées à la pensée informatique lors d'activités de programmation en dyades au primaire (Wei et al., 2021). Or, dans leur revue de la littérature sur la pensée informatique, Lye et Koh (2014) affirment : « with the paucity of research in naturalistic classroom settings, there is, clearly, a need to conduct studies in this area. This will better help to inform educators and researchers on how to design and implement the grade-appropriate programming into K-12 curriculum » (p. 57). Ce constat met d'ailleurs en lumière la tendance à observer ou évaluer la résolution collaborative de problème à l'aide d'agents virtuels (OCDE, 2017) : en interagissant avec un agent virtuel dans un scénario fermé, l'élève doit résoudre un problème. Il s'agit du dispositif utilisé dans le cadre de l'étude *Programme for International Student Assessment's* (PISA) de 2015. En plus de soulever de nombreuses questions quant à l'authenticité de l'activité¹⁰, cette méthode évacue le contexte et l'environnement (la classe) comme variable dans le processus de résolution collaborative de problèmes.

En s'intéressant aux élèves du niveau primaire, notre recherche pourra contribuer aux réflexions déjà initiées dans les travaux des auteurs cités. Cela dit, notre choix de nous concentrer sur le primaire va au-delà de la littérature : en pouvant contribuer au développement de compétences transversales, la programmation pourrait offrir aux apprenants des habiletés qu'ils pourront réinvestir et continuer de développer tout au long de leurs études ultérieures.

Enfin, au regard de la pertinence sociale, nous anticipons des retombées relatives aux pratiques pédagogiques. Nous souhaitons présenter aux praticiens (enseignants et autres intervenants) les effets empiriques de la programmation en contexte scolaire, à l'heure où : a) des gouvernements la rendent obligatoire (Government of the United Kingdom, 2017; Ministry of Education and Research, 2014; Toikkanen, 2015), b) les enseignants l'utilisent de plus en plus (Karsenti et Bugmann, 2017), et c) il semble important de considérer son incidence sur la mobilisation et le développement des « compétences du 21^e siècle », qui jouissent d'un grand intérêt dans le milieu de l'éducation (Griffin et Care, 2015; Romero et Vallerand, 2016). Fessakis, Gouli et Mavroudi ont d'ailleurs souligné le besoin que nous ciblons :

¹⁰ Nous approfondissons cet aspect à la section 2.3.1. de ce manuscrit.

« The availability of software programming environments is not enough for the utilization of the learning potential of programming and its smooth integration in school practice though. Experimentally validated teaching/learning approaches, documented best practices, learning resources, curriculum standards, professional development and support for teachers are also needed. Such efforts should address the widest age spectrum possible, thus providing developmentally appropriate learning opportunities in a continuous and consistent way » (2013, p. 96).

Qui plus est, le lancement du *Plan d'action numérique en éducation et en enseignement supérieur* (PANÉES) du MÉES a induit une nouvelle réalité à laquelle le système scolaire québécois a dû faire face dès 2018, et qui demeure une préoccupation. En effet, avec les nombreuses mesures et la liste de matériel proposé pour l'achat dans les établissements d'enseignement, les enseignants ont dû choisir puis introduire un ou plusieurs appareils technologiques dans leurs pratiques pédagogiques. Nous sommes d'avis que notre recherche pourra contribuer à la discussion entourant la mise en oeuvre du PANÉES et des mesures en découlant au Québec, notamment à l'égard des choix associés à la programmation (par ex. : dispositifs robotiques).

En guise de conclusion à ce chapitre, nous réitérons que ce projet de recherche repose sur une conception bipartite de la programmation, qui est à la fois un objet et un outil d'apprentissage. Nous proposons une vision intégratrice dans laquelle l'usage de dispositifs robotiques est associé à la programmation dans le but de favoriser la mobilisation, voire le développement, de compétences telles que la résolution de problème, le raisonnement logique, la collaboration et la pensée critique. Nous cherchons à comprendre l'incidence de l'utilisation de la programmation sur deux de ces compétences transversales majeures, soit la résolution de problèmes et la collaboration. Le prochain chapitre décline le cadre théorique de cette recherche, qui aboutira à la formulation des objectifs spécifiques de recherche.

Chapitre 2 - Cadre théorique

Le cadre théorique, aussi appelé cadre de référence, est une composante fondamentale de tout projet de recherche scientifique. En effet, non seulement permet-il de tracer un portrait théorique et scientifique sous-tendant les objectifs de recherche, mais il oriente également, en amont du projet de recherche, « les décisions concernant la nature des données à recueillir ainsi que l'analyse et l'interprétation que l'on peut en faire » (Gohier, 2018, p. 109). C'est donc dans cette perspective que nous présentons ici, dans un premier temps, le concept de programmation, permettant de saisir comment il s'articule aux autres concepts. Dans un second temps, nous déclinons la notion de compétence, puis nous détaillons dans un troisième temps le concept de compétence transversale, et plus particulièrement les compétences en résolution de problèmes et en collaboration. Enfin, nous énonçons les objectifs spécifiques de cette recherche.

2.1 La programmation : un outil de médiation humain-ordinateur

Depuis sa création et au gré des innovations technologiques, l'ordinateur a vu ses capacités décupler. Cela dit, malgré les nombreuses prouesses qu'on lui attribue aujourd'hui, son fonctionnement dépend de fondements remontant aux premières itérations de cette machine complexe. En effet, le fonctionnement de l'ordinateur repose essentiellement sur le traitement et la mémorisation d'informations (Bossuet, 1982; Guttag, 2017). N'ayant pas la capacité de réfléchir, l'ordinateur doit être contrôlé par des indications claires, précises et ne laissant pas de place à l'interprétation (Hardouin-Mercier, 1974). Ces indications claires sont le coeur de la programmation. Cette dernière est donc un procédé permettant de transmettre des consignes claires à un agent de traitement de l'information : « the purpose of writing a program is to communicate complete, detailed and unambiguous instructions for achieving a specific goal » (Turski, 1978, p. 6). Le programme constitue l'ensemble des consignes qui sont soumises à l'ordinateur en vue de lui faire exécuter une tâche en particulier.

Avec la popularité grandissante du phénomène de la programmation au cours des dernières années, le terme « coder » s'est insinué et est maintenant très répandu lorsqu'il est question de

langage informatique. Or, il est important de différencier l'action de communiquer avec l'ordinateur (la programmation) de l'écriture elle-même (le code). En effet, l'activité de programmation se décline en deux volets : l'acte de programmer et l'acte de coder. L'acte de programmer consiste à élaborer une structure complexe et organisée de consignes, c'est-à-dire un algorithme (Henri, 2014; Rice et Desautels, 1969) qui permet de mettre en œuvre les différentes fonctionnalités de la machine (Hardouin-Mercier, 1974; Van Roy et Haridi, 2004). L'acte de coder fait plutôt référence à la traduction de ces séries de consignes en un « langage parfaitement formalisé » qui pourra être compris par l'ordinateur (Baron et Bruillard, 2001; Lopez, 1986). Bien que souvent associée au simple fait d'écrire des lignes de codes, on nous rappelle qu'il est important de considérer la programmation comme un « exercice beaucoup plus large qui englobe des activités de conception, d'écriture, de test et de maintenance; [et qui] s'apparente autant, voire davantage, à la résolution de problèmes qu'à l'écriture d'un code » (Henri, 2014, p. 11). Ce point de vue est partagé par Mannila et ses collègues (2014), qui rappellent que le codage n'est qu'une partie du processus de programmation.

Le programme est une suite de consignes qui sont définies, puis envoyées à l'ordinateur (Lu et Fletcher, 2009). La meilleure façon de concevoir la programmation est donc de la considérer comme étant la composante d'un système langagier. Il s'agit de donner une marche à suivre à un exécutant — l'ordinateur — afin de lui faire exécuter une tâche (Duchâteau, 1989). Ce processus de programmation peut être représenté à l'aide d'outils comme l'organigramme – on retrouve dans la littérature le terme spécialisé « ordinogramme » (Hardouin-Mercier, 1974; Henri, 2014) – qui met en évidence les liens logiques et conditionnels entre les différentes parties d'un algorithme, d'un programme. Les ordinogrammes peuvent ensuite être traduits de façon textuelle, par le codage, en un langage spécifique de programmation. En s'inspirant des travaux de Bruner (1967) sur les modes de représentation de la connaissance – soit *enactive*, *ikonik* et *symbolic* – Ruf, Berges et Hubweiser (2015) identifient trois formes de représentation en programmation : le texte, le code et le diagramme¹¹.

¹¹ Le terme *diagramme* est synonyme des termes *organigramme* et *ordinogramme*, tels que présentés précédemment.

Il existe d'ailleurs plusieurs langages de programmation : ils ont tous la même utilité, mais ils ne sont pas tous construits de la même façon, au même titre que les langues parlées à travers le monde. Un même algorithme pourra donc être écrit à l'aide de différents codes, menant toutefois à un même résultat. Or, comme nous l'avons mentionné, l'ordinateur n'effectue fondamentalement que deux opérations : il fait des calculs et se souvient des résultats (Guttag, 2017). Le langage informatique se distingue donc du langage auquel l'humain est habitué : il ne laisse pas de place à l'interprétation et possède très peu de mots (Calmet et al., 2016). La moindre erreur de la part du programmeur, comme l'oubli d'un signe dans une opération, entraînera inévitablement l'échec du programme. C'est d'ailleurs là que réside tout le potentiel cognitif de l'interaction avec l'ordinateur via la programmation, qui s'avère être un processus d'une grande complexité. Dans leur analyse d'outils pédagogiques, Ruf, Berges et Hubweiser (2015) ont identifié 11 types de tâches ou d'opérations associées à la programmation.

Type de tâche	Formes de représentation de la connaissance
Écrire le code	Texte → Code Diagramme → Code
Écrire le code en utilisant un autre code	Texte et code → Code
Ajuster, étendre ou compléter le code	Texte et code → Code
Optimiser le code	Texte et code → Code
Déboguer le code	Code (et texte) → Code (et texte)
Définir les bonnes conditions pour le code	Texte et code → Texte
Tester le code	Code → Code
Transformer le code	Code → Code
Tracer/expliquer le code	Code → Texte
Spécifier un problème pour le code	Code → Texte
Dessiner un diagramme pour le code	Code → Diagramme

Tableau 1. – Types de tâches de programmation (Ruf, Berges et Hubweiser, 2015)

Leurs travaux nous permettent de comprendre comment, en contexte pédagogique, peut être utilisée la programmation, un processus qui peut être formalisé puis enseigné et appris à l'école.

2.1.1 Perspectives pédagogiques de la programmation

La mission de l'école québécoise se décline en trois volets : instruire, socialiser et qualifier (Gouvernement du Québec, 2006). Alors que le premier volet (instruire) semble particulièrement cohérent avec l'apprentissage de la programmation, l'association avec les deux autres volets peut paraître moins explicite. Or, l'une des principales composantes du volet « socialiser » consiste à faire des apprenants de futurs citoyens responsables, ce qui nous paraît tout à fait en ligne avec l'aspiration à enseigner la programmation, dans la mesure où cet enseignement pourrait contribuer à la compréhension du fonctionnement des outils technologiques, de même qu'à leur utilisation éthique et responsable. Idem pour le volet « qualifier », puisque l'apprentissage de la programmation et la connaissance du fonctionnement des appareils technologiques offriront aux apprenants un bagage de connaissances pertinentes avec le marché du travail, qui a vu se déployer ce que plusieurs appellent déjà la quatrième Révolution industrielle, où la technologie (intelligence artificielle, robotique, etc.) est omniprésente et centrale (Gaspar et al., 2018; Johannessen, 2018; Lee et al., 2018). Le Forum économique mondial, dans un rapport prospectif, abonde en ce sens : « the individuals who will succeed in the economy of the future will be those who can complement the work done by mechanical or algorithmic technologies, and 'work with the machines' » (2018, p. 3).

Considérant cette omniprésence technologique et les incidences de la quatrième Révolution industrielle, il semble indispensable d'amener les apprenants à réfléchir aux répercussions potentielles et variées du numérique, en plus de les accompagner dans la mobilisation, voire le développement, de différentes habiletés et compétences technologiques. La programmation peut être une façon d'y parvenir (Romero, 2016b). Dans cette réflexion approfondie sur les technologies en éducation, il est important de considérer leur double finalité en contexte pédagogique. Il s'agit là d'un aspect que Bossuet (1983) aborde à un niveau global en parlant des vertus transversales (ou multidisciplinaires) de l'informatique :

« Par la pratique de l'informatique, l'élève acquiert des méthodes et des techniques transposables dans d'autres disciplines, en particulier en mathématiques. Par l'utilisation d'un système informatique, l'élève acquiert ou vérifie des modèles de pensée dans un contexte de résolution de problèmes ou de communication engendrée par la présence d'un système informatique motivant » (Bossuet, 1983, p. 39-40).

Cette réalité a aussi été soulevée par Tondeur, Van Braak et Valcke (2007), qui distinguent trois types d'utilisation pédagogique de l'ordinateur. Le premier type, *basic computer skills*, se rattache à l'ordinateur en tant qu'objet d'apprentissage, alors que les deux autres types, *computers as an information tool* et *computers as a learning tool*, renvoient plutôt à son statut d'outil d'apprentissage (Tondeur et al., 2007). Dans une démarche plus près de notre objectif de recherche, les travaux de Romero, Lepage et Lille (2017) soulignent cette double finalité en ciblant exclusivement la programmation. En effet, les auteurs proposent un modèle de progression en cinq niveaux, débutant par l'apprentissage des fondements de la programmation, évoluant ensuite vers un « apprentissage par le biais de la programmation ». Ces travaux récents sont en adéquation avec ceux de Kelleher et Pausch (2005), qui ont souligné la double finalité dans une recension d'environnements de programmation destinés à une utilisation pédagogique.

Cette coexistence du statut d'objet et d'outil d'apprentissage de la programmation est en cohésion avec le double statut aussi accordé à l'ordinateur (Bossuet, 1983). Évidemment, cette double finalité peut être observée à tous les niveaux d'enseignement, selon les différentes caractéristiques et circonstances inhérentes à chacun de ces niveaux. Ci-après, nous évoquons les principaux impacts de l'utilisation de la programmation à l'école ainsi que les outils principaux recensés dans la littérature.

2.1.2 L'apprentissage de la programmation au primaire

La littérature portant sur l'apprentissage de la programmation nous informe de certains impacts observés auprès d'élèves du primaire. D'abord, nous évoquons les travaux de Clements et Gullo (1984), qui ont étudié la programmation en langage LOGO. Dans leur étude menée auprès de 18 élèves, ils ont été en mesure d'observer un développement de la compétence en RP chez les participants à l'aide d'une démarche quasi expérimentale (groupe contrôle et groupe expérimental, prétest et post-test). Komis et Misirli (2011) ont aussi pu observer des effets cognitifs relatifs aux stratégies de RP auprès d'un échantillon de 92 élèves. Ces conclusions entrent toutefois en contradiction avec celles de Kalelioglu (2015), dans son étude menée auprès de 32 élèves, où il souligne l'absence d'une évolution significative de la compétence en RP, de

même qu'avec celles de Lin et al. (2005), qui relatent la simple mobilisation de compétences en RP dans le cadre de leur expérimentation.

Comme nous avons pu le constater dans le premier chapitre de cette thèse, certains chercheurs ont reconnu les possibilités s'offrant aux enseignants souhaitant intégrer la programmation en contexte scolaire (Bossuet, 1982; Papert, 1994). Cela dit, l'âge des apprenants est un aspect non négligeable lors de l'utilisation de l'informatique à l'école, surtout lorsqu'il est question de programmation. On peut compter parmi ses applications pédagogiques deux méthodes qui permettent d'adapter la tâche de programmation à des enfants d'âge primaire tout en conservant son potentiel de développement de compétence : la programmation visuelle et la programmation tangible.

2.1.2.1 La programmation visuelle

L'utilisation de la programmation visuelle permet principalement de pallier l'opacité du langage informatique écrit (figure 2). Avec la programmation visuelle, les lignes de codes sont remplacées par des boîtes unies à l'aide de liens (Green et Petre, 1996).

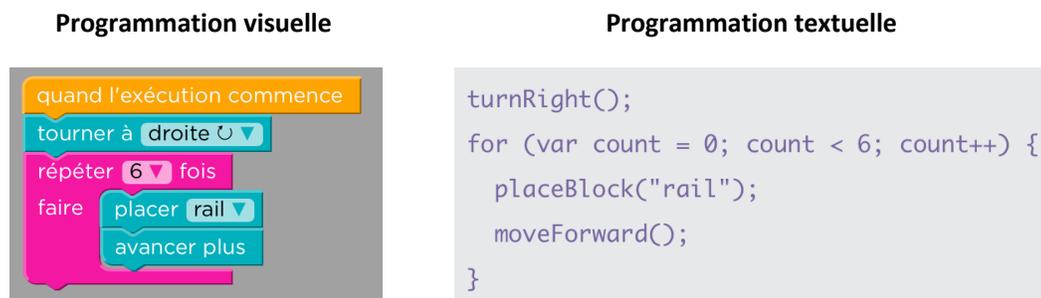


Figure 2. – Représentations pour une série de commandes sur studio.code.org

Les termes employés en programmation sont en anglais, ce qui peut constituer un obstacle non négligeable pour des élèves francophones du primaire. De fait, la programmation visuelle peut être adaptée en fonction de la langue de l'utilisateur, ce qui rend la programmation accessible à un ensemble très vaste et varié d'apprenants. Il est également à noter que la programmation visuelle permettrait de développer la compétence en résolution de problème chez les enfants (Lai et Yang, 2011). En effet, ce mode de programmation permet de rendre plus accessibles de

nombreux problèmes associés à la structure et à la logique des codes : à l'aide d'une méthode d'essai-erreur, les apprenants peuvent tenter de réorganiser leur programme afin qu'il exécute la tâche attendue. Chacun de leurs essais est accompagné d'une rétroaction immédiate (accomplissement [ou non] de la tâche anticipée) permettant de confirmer la validité ou l'invalidité du programme. Bien que ce processus itératif soit tout à fait présent dans la programmation conventionnelle, il n'est pas accessible aux élèves du primaire. Qui plus est, dans une étude menée auprès de 107 élèves du primaire, Sáez-López et al. (2016) ont évalué l'impact de la programmation visuelle avec la plateforme *Scratch*. Ils en sont venus à la conclusion que la programmation visuelle peut contribuer à l'apprentissage de concepts de programmation, au développement de la logique et des pratiques informatiques, tout en sollicitant la motivation, le plaisir et l'enthousiasme des élèves. L'aspect motivationnel est d'ailleurs mentionné de façon récurrente dans la littérature. C'est notamment le cas dans l'étude de Serafini (2011), où il est mentionné que la motivation des élèves était si présente qu'ils ne voulaient plus s'arrêter à la fin des périodes d'activité, de même que dans l'étude de Bugmann et Karsenti (2018a), où la motivation suscitée par la programmation avait des effets positifs sur la perception qu'avaient des élèves de l'école, et ce, en milieux socioéconomiques défavorisés.

Quant aux autres effets de la programmation visuelle, il y a non seulement sa capacité à faciliter le passage vers un langage textuel plus complexe (Armoni et al., 2015) et la généralisabilité des tâches qu'elle permet (Alexandrova et al., 2015), mais aussi le développement potentiel de la pensée critique, de la compétence en RP et de la collaboration (Pinto-Llorente et al., 2017). Une méta-analyse de Xu, Ritzhaupt, Tian et Umapathy (2019) démontre d'ailleurs que la programmation visuelle, aussi appelée programmation par blocs (*block-based programming*), aurait chez les programmeurs novices un effet cognitif¹² plus important que la programmation textuelle. Néanmoins, certains auteurs avancent que la programmation visuelle ne permettrait pas l'accès à un niveau de complexité comparable aux méthodes de programmation conventionnelle (Hendrix et Weeks, 2018).

¹² L'effet cognitif est décrit par les auteurs comme référant à la résolution de problème et à différentes opérations des six niveaux de la taxonomie de Bloom (connaissance, compréhension, application, analyse, synthèse et évaluation).

2.1.2.2 La programmation tangible

La programmation tangible consiste à utiliser un dispositif physique que l'enfant peut manipuler afin de rendre les actions programmées plus visibles (Henry et al., 2018; Komis et Misirli, 2013). Cette pratique est associée au maintien de la motivation et au développement d'une certaine confiance envers la programmation qui, dans sa forme originale textuelle, peut s'avérer difficile d'approche et peu motivante pour des élèves du primaire (Pullan, 2013). La robotique pédagogique est un très bon exemple de programmation tangible. Elle est d'ailleurs très propice à la collaboration en classe : que ce soit entre l'élève et un pair ou encore entre l'élève et son enseignant, la collaboration peut survenir lors des manipulations du robot et pendant le processus de programmation, où les élèves auront à faire face à des bogues (*bugs*), c'est-à-dire des problèmes empêchant le bon fonctionnement du programme. Ces bogues sont rarement évidents : ils induisent donc des situations de résolution de problème en collaboration où le but commun est de faire fonctionner le programme correctement (Mauch, 2001; Nugent et al., 2009; Petre et Price, 2004). La programmation semble donc être un outil utile pour mobiliser certaines compétences. À cet effet, les mots de Seymour Papert sont évocateurs :

« Quand on apprend à programmer un ordinateur, on n'y arrive presque jamais du premier coup. Apprendre à passer maître en l'art de programmer, c'est devenir hautement habile à déceler où se nichent les "bugs" et à y remédier » (Papert, 1981).

Pour parvenir à résoudre des bogues, les élèves doivent mettre en œuvre différentes compétences comme l'observation, l'estimation et la manipulation (Sullivan, 2008), ou encore l'interaction sociale, c'est-à-dire l'établissement de liens avec les pairs (Mitnik et al., 2008). Ces propos sont appuyés par ceux d'Israel et ses collègues (2015), qui évoquent aussi un potentiel intéressant chez les élèves en situation de handicap, notamment au regard de la motricité fine et de la résolution de problèmes complexes. Or, au-delà de son potentiel de différenciation pédagogique, la programmation à l'aide de dispositifs robotiques (réels ou virtuels) permettrait le développement de la pensée spatiale et de la syntonie du corps¹³ (Repenning et al., 2014). De l'anglais *body sytonicity*, ce terme renvoie à la faculté d'une personne de se mettre à la place d'un objet, comme lorsque des enfants programment un robot et se projettent sur lui afin de

¹³ Traduction libre (« *body sytonicity* »)(Repenning et al., 2014).

planifier les étapes subséquentes de programmation (Papert, 1980) : « if I'm this turtle and I turned 90 degrees left, what would I see? » (Repenning et al., 2014, p. 70). L'utilisation de la programmation au sein d'univers virtuels en 3D a aussi été étudiée par Bugmann, Karsenti et Parent (2018), où la conception d'un dispositif intitulé « Deviens un maître MineScratch » a permis de combiner l'univers en 3D du jeu *Minecraft* à la plateforme de programmation *Scratch*, engendrant ainsi de nombreux impacts positifs, notamment en lien avec les stratégies de RP, la collaboration et la créativité des élèves.

Nous retenons donc que la programmation visuelle (par blocs) ainsi que la programmation tangible (dispositif robotique) permettent de simplifier l'acte de programmer, rendant ainsi la programmation accessible à des élèves de niveau primaire. Leur utilisation concomitante permet de créer des situations propices à la mobilisation de différentes compétences comme la résolution de problème et la collaboration. L'interaction entre l'enfant, l'ordinateur et ses pairs se déroule dans un environnement¹⁴ au sein duquel l'élève, en collaboration ou non, utilise un langage (codes) pour faire réaliser des actions à l'ordinateur ou à un robot. C'est l'ensemble de ce processus qui permet de développer des compétences, notamment en contexte de résolution de problème.

En somme, la programmation peut être employée selon deux utilités : d'une part en tant qu'outil d'apprentissage et d'autre part en tant qu'objet d'apprentissage. Nous avons pu constater que ces deux utilités amènent des impacts variés, avec une forte occurrence de la compétence en RP, de la collaboration et de la motivation. Cela semble réaffirmer la pertinence de notre objectif de recherche, qui cible la résolution collaborative de problèmes (RCP)¹⁵. Par ailleurs, au primaire, des adaptations pédagogiques permettent de rendre la programmation accessible à des élèves de 6 à 12 ans, notamment à l'aide de la programmation visuelle (blocs colorés) et de la programmation tangible (animation de dispositifs robotiques ou virtuels, en 3D). Cela dit, les répercussions potentiellement positives de ces pratiques sont contrebalancées par des limites, notamment en

¹⁴ Cet environnement a été identifié par plusieurs auteurs en tant que « micromonde », un concept qui renvoie à un espace abstrait au sein duquel l'individu, seul ou avec un pair, interagit avec l'ordinateur (voir Depover, Karsenti et Komis, 2007; Briant et Bronner, 2015). Dans ce cas-ci, l'interaction se fait à l'aide de la programmation.

¹⁵ La RCP sera abordée à la section 2.2.1.3. de ce manuscrit.

ce qui concerne les niveaux de complexité atteignables. À tout prendre, ces deux applications pédagogiques de la programmation sont des outils relativement bénéfiques en salle de classe, alors que les limites nous apparaissent être moins importantes (quantitativement et qualitativement) que les impacts positifs potentiels. Dans la littérature évoquée, il semble y avoir un certain consensus autour de la mobilisation de la RP à l'aide de la programmation, sans toutefois que l'on explique comment s'effectue cette mobilisation, c'est-à-dire à quels moments dans le processus de programmation et de quelle façon. Nous croyons qu'il s'agit d'un élément de pertinence scientifique de notre recherche, qui contribue à nourrir la littérature à l'égard du lien entre la programmation et la compétence en résolution collaborative de problèmes.

2.2 Les compétences transversales

Depuis la réforme du système éducatif québécois en 1998, l'approche par compétence guide et encadre l'enseignement dans les établissements d'enseignement. Avec l'approche par compétence, c'est l'élève qui est appelé à construire ses connaissances avec l'aide de l'enseignant ou de ses pairs, alors que l'enseignant agit à titre d'accompagnateur (Boutin, 2004). Au cœur de cette approche se trouve la compétence. Cette dernière a été définie par le ministère de l'Éducation, du Loisir et du Sport (MELS) comme « un savoir-agir fondé sur la mobilisation et l'utilisation efficaces d'un ensemble de ressources » (2006, p. 4). Précisons que le savoir-agir peut être mobilisé dans des situations complexes et variées (Gouvernement du Québec, 2006) alors que le savoir-faire renvoie plutôt à l'exécution d'une tâche particulière dans un contexte particulier : le savoir-agir peut donc être constitué de plusieurs savoir-faire (Le Boterf, 2002). Ce savoir-agir est une démarche qui ne peut être automatisée et qui se déroule dans une situation relativement complexe (Lasnier, 2014).

Selon Lasnier, la compétence possède une structure particulière : elle se subdivise en capacités, qui sont séparées en habiletés. Chaque habileté contient à son tour des connaissances. Dans le Programme de formation de l'école québécoise (PFÉQ), les concepts de capacité et d'habileté sont représentés par ce qu'on appelle les « composantes de la compétence ».

Par exemple, pour la compétence « Lire des textes variés » en français, il y a la composante « utiliser le contenu des textes à diverses fins », une habileté composant la compétence. Cet

exemple tiré du PFÉQ relève du premier type de compétence que l'on peut y trouver, c'est-à-dire la compétence disciplinaire, associée spécifiquement à une discipline (ou matière) comme le français et les mathématiques. Le second type est la compétence transversale. La transversalité est définie par Fourez (2005) comme étant « liée à la possibilité de transférer une démarche, une notion, une compétence, etc., d'un contexte à un autre » (p. 402). Considérant le caractère général et transdisciplinaire des compétences mobilisées lors d'activités de programmation, nous nous intéressons à ces compétences dites transversales.

Les compétences transversales sont développées non seulement pour être mobilisées en contexte scolaire, mais aussi (et surtout) dans la vie de tous les jours et *a posteriori* sur le marché du travail. La définition de la compétence transversale du PFÉQ fait mention d'un savoir-agir qui va au-delà du cadre des disciplines et qui trouve son utilité dans des situations concrètes (Gouvernement du Québec, 2006). Le programme québécois comporte neuf compétences transversales regroupées en quatre ordres : intellectuel, personnel et social, méthodologique ainsi que de la communication. L'ordre intellectuel regroupe les compétences liées à l'exploitation de l'information, à la résolution de problèmes, à l'exercice du jugement critique et à la mise en œuvre de la pensée créatrice. L'ordre personnel et social, quant à lui, aborde la structuration de l'identité et la coopération. Puis, l'ordre méthodologique réunit les compétences relatives aux méthodes de travail efficaces et à l'exploitation des TIC. Enfin, l'ordre de la communication ne contient qu'une compétence : « communiquer de façon appropriée » (Gouvernement du Québec, 2006).

Les compétences transversales du PFÉQ se rapprochent, à plusieurs égards, des « compétences du 21^e siècle », qui ont gagné en importance au cours des dernières années. La littérature consultée (tableau 2) démontre qu'il semble y avoir un cœur commun à tous les modèles des compétences du 21^e siècle : la pensée critique, la créativité, l'innovation, la communication, la résolution de problèmes et la collaboration. Bien que pouvant être associées aux impératifs économiques du marché du travail, les compétences du 21^e siècle peuvent tout de même être considérées dans une perspective pédagogique, sans l'utilisation d'un appareil numérique (Ananiadou et Claro, 2009). Ces auteures en proposent d'ailleurs une définition compréhensive :

« skills and competencies young people will be required to have in order to be effective workers and citizens in the knowledge society of the 21st century » (p. 8).

On retrouve plusieurs similitudes entre les compétences transversales du PFÉQ et les différents modèles présentés dans le tableau 2. Par exemple, il est possible de retrouver l'ensemble des quatre ordres du PFÉQ dans le cadre de Griffin et Care (2015) : intellectuel, méthodologique, personnel et social ainsi que de la communication. La catégorisation en ordres a d'ailleurs été employée par plusieurs modèles (*ways of...thinking, working, etc.*). Bien que les termes puissent varier d'un modèle à l'autre, le contenu est souvent similaire.

	Pensée critique	Résolution de problèmes	Créativité / Innovation	Collaboration / Communication	Citoyenneté	Métacognition	Littératie informationnelle	Littératie numérique	Pensée informatique	Développement professionnel
C21 Canada (2012)	•	•	•	•	•	•	•	•		
Griffin et Care (2015)	•	•	•	•	•	•	•	•		•
Ministère de l'Éducation de l'Ontario (2016)	•	•	•	•	•	•				
Ananiadou et Claro (2009)	•	•	•	•	•	•	•	•		
Romero et Vallerand (2016)	•	•	•	•					•	

Tableau 2. – Tableau synthèse des compétences du 21^e siècle

Nous avons d'ailleurs décidé d'inscrire la thèse dans la foulée des travaux de Romero et Vallerand (2016), qui proposent cinq compétences clés pour le 21^e siècle : la pensée critique, la collaboration, la résolution de problèmes, la créativité et la pensée informatique. Ce modèle (figure 3) est le seul des principaux modèles incluant la pensée informatique explicitement, accordant ainsi une importance non négligeable aux opérations mentales liées à la programmation. Les cinq compétences du modèle de Romero et Vallerand sont presque toutes en adéquation avec le PFÉQ, exception faite de la cinquième (la pensée informatique), qui n'y figure pas. Ce modèle semble donc se distinguer des autres modèles évoqués, qui n'abordent que très superficiellement le lien entre le numérique et la pensée. Dans une recension systématique de la littérature portant sur les compétences du 21^e siècle en enseignement primaire, Chalkiadaki (2018) identifie plusieurs occurrences de la pensée, comme la pensée critique (*critical thinking*), le *high-order thinking*, le *divergent thinking* ou encore *l'inventive thinking*, sans toutefois que l'on puisse identifier explicitement le *computational thinking* (pensée informatique) ou toute autre forme de lien explicite entre l'informatique et la pensée.

Au demeurant, le choix du modèle de Romero et Vallerand s'explique par l'intégration de l'informatique dans les compétences du 21^e siècle, offrant ainsi un angle d'approche tant novateur que pertinent. Le potentiel de développement transdisciplinaire de l'informatique, et par extension de la programmation, a déjà été soulevé dans la littérature (Depover et Noël, 2003; Hromkovic, 2006). Ce modèle nous rapporte également au questionnement, évoqué dans le premier chapitre de ce manuscrit, portant sur le caractère disciplinaire ou transversal de la programmation en tant que contenu scolaire. Considérant les ramifications transversales de la programmation, bien illustrées par son intégration dans les compétences du 21^e siècle, il semble que son caractère transversal sur le plan conceptuel soit une piste d'analyse intéressante. Or, d'un point de vue pratique, pour l'enseignement dans les salles de classe, la question demeure entière et il nous semble peu réaliste d'y apporter une réponse dans le cadre de cette thèse; nous sommes toutefois d'avis qu'il s'agit d'un enjeu notable qui gagnerait à être étudié.

Un aspect particulièrement intéressant de ce modèle est la mise en relation des cinq compétences : les carrefours créés par l'association de compétences font apparaître des entités comme la résolution collaborative de problèmes. Les associations mettent en évidence les liens

étroits entre ces compétences de même que leur cohésion. Par ailleurs, nous avons aussi retenu ce modèle parmi d'autres puisqu'il inclut la pensée informatique, une compétence très fortement liée à la programmation (Brennan et Resnick, 2012; Romero et al., 2017; Tchounikine, 2016).

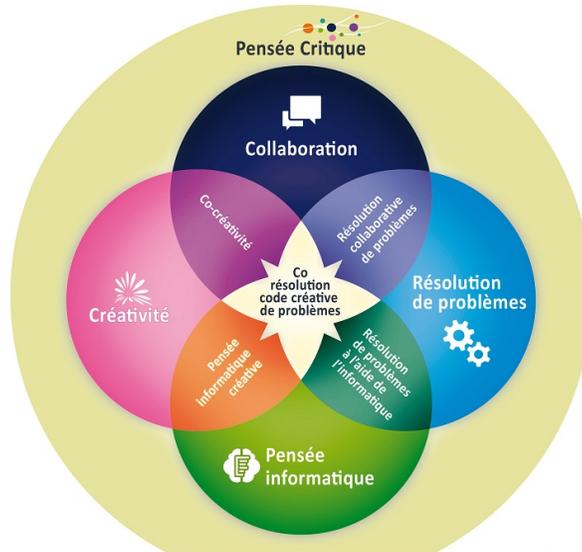


Figure 3. – Compétences clés pour le 21^e siècle (Romero et Vallerand, 2016)¹⁶

En somme, les similitudes observées entre ces modèles et le PFÉQ tendent à confirmer la pertinence de s'intéresser à ces compétences, quel que soit le nom qu'on leur donne (transversales, du 21^e siècle, etc.). Dans le cadre de cette recherche, nous avons ciblé la résolution collaborative de problèmes, que nous avons déclinée en deux unités de sens pour la définir adéquatement, c'est-à-dire les deux compétences correspondantes : la résolution de problèmes et la collaboration.

2.2.1 La résolution de problèmes

La résolution de problèmes occupe une place centrale dans la pensée informatique (PI) selon plusieurs auteurs (Aho, 2012; Barr et Stephenson, 2011; Curzon et McOwan, 2017; Delcker et Ifenthaler, 2017; Hu, 2011; Tchounikine, 2016; Voogt et al., 2015). La littérature positionne la PI à la rencontre de la RP et de l'informatique, et ce, depuis plusieurs décennies. En effet, antérieur à ces travaux, un rapport de recherche publié en 1969 par Feurzeig, Papert, Bloom, Grant et

¹⁶ Cette création de Romero et Vallerand (2016) est mise à disposition selon les termes de la Licence Creative Commons Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique 4.0 International.

Solomon, intitulé *Programming-languages as a conceptual framework for teaching mathematics*, soulignait la proximité entre la programmation et la résolution de problème : « it provides a natural context to concretize the approach to teaching associated with the name of George Polya » (p.8). Le mathématicien américain George Polya a publié de nombreux articles et ouvrages liés au domaine de la mathématique, notamment sur la RP. Dans l'un de ses travaux, il énonce quatre étapes du processus de RP : 1) comprendre le problème (ce qui est demandé; y a-t-il assez d'informations?), 2) faire un plan (chercher des régularités; organiser l'information), 3) exécuter le plan et 4) évaluer l'efficacité du plan (Polya, 1957). Ce processus établit implicitement le caractère itératif du processus de RP puisque chacune de ces étapes ne s'inscrit pas dans une chronologie linéaire, mais plutôt dans un cycle où l'individu répète les étapes pertinentes jusqu'à l'atteinte du but visé. La littérature ultérieure s'inscrit dans cette structure fondamentale, tout en offrant des précisions et des nuances. Par exemple, les recherches de Simon et Newell (1971) abordent la « théorie de la résolution humaine de problème » en décrivant un système de traitement de l'information (*information-processing system*) au sein d'un construit qui n'avait pas été abordé par Polya à l'origine : l'espace problème (*problem space*). Plus d'une décennie plus tard, les travaux de Greeno et Simon (1984) abondent dans le même sens en évoquant des concepts comme le *problem space* et les *problem representations*, c'est-à-dire, les représentations mentales du problème que se construit l'individu, de même que la *means-ends analysis*, la méthode permettant de conjuguer les ressources disponibles aux buts à atteindre. Ces concepts démontrent une forte cohérence tant avec les travaux de Polya qu'avec ceux de Simon et Newell. Près de 30 ans plus tard, les travaux de Jonassen s'inscrivent dans cette lignée. Selon Jonassen (2011), le processus cognitif de RP possède deux caractéristiques fondamentales :

« First, problem solving requires the mental representation of the problem, known as the problem space, problem schema, or mental model of the problem. [...] Second, problem solving requires some manipulation and testing of the mental model of the problem in order to generate a solution. Problem solvers act on the problem space in order to generate and test hypotheses and solutions » (p. 4).

Il s'agit donc de comprendre le problème, de s'en faire une représentation mentale, d'agir et de tester les hypothèses au sein de cette représentation mentale avant d'appliquer la solution dans le réel. Dans ses travaux subséquents, Jonassen (2014) avance que la RP est un processus subdivisé en étapes qui se succèdent de façon logique et itérative : la définition du problème,

l'analyse du problème, la collecte d'informations, l'élaboration de pistes de solutions, l'évaluation de solutions alternatives, l'application des solutions et finalement, la vérification de celles-ci (Jonassen, 2014).

Pour sa part, l'OCDE a explicité les différents processus cognitifs mobilisés par les élèves pendant le processus de RP : l'exploration et la compréhension, la représentation et la formulation, la planification et l'exécution, et enfin, le suivi et la réflexion (OCDE, 2015a). Le processus cognitif d'exploration et de compréhension correspond au premier contact qu'entretient l'élève avec le problème. Il va lire et s'approprier les éléments inhérents au problème en cherchant à comprendre ce qui est attendu de lui. Puis, la représentation et la formulation permettent à l'élève de se représenter le problème, tant mentalement que concrètement. En utilisant des modes de représentation comme le diagramme, le dessin ou la parole, il pourra étoffer sa compréhension et, de fait, ses représentations mentales. Ce processus est aussi le moment propice à la formulation d'hypothèses en fonction des éléments présents dans le problème. Le processus de planification et d'exécution, quant à lui, consiste en l'élaboration d'une marche à suivre en vue de résoudre le problème, impliquant la formulation d'objectifs généraux ou spécifiques. Cette marche à suivre devra ensuite être appliquée. Enfin, le suivi et la réflexion permettent de poser un regard sur la démarche suivie afin d'en tirer des conclusions, qui permettront de prendre des décisions quant au déroulement de la démarche : des conclusions positives suggèrent de poursuivre le travail déjà entamé alors que le contraire pourrait mener à une remise en question et à une modification de la marche à suivre.

Il est donc possible de constater, à travers ces processus cognitifs, que la résolution de problème est fondamentalement structurée et articulée par une démarche intellectuelle à la fois complexe et souple. Le modèle intégrateur de la RP selon Robertson (2017) démontre les forts liens unissant les différents auteurs dont les travaux ont été présentés (figure 4).

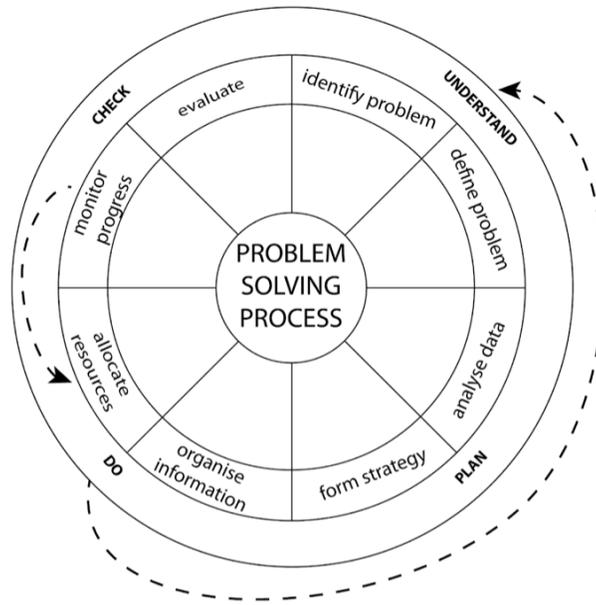


Figure 4. – Modèle du processus de RP (Robertson, 2017)

Robertson expose, dans le cercle externe, les quatre étapes selon Polya (1957), et dans le cercle interne, des étapes qui s'apparentent beaucoup à celles de Jonassen (2014), exception faite de l'allocation des ressources (*allocate resources*), qui semble se rattacher fortement à la sixième étape de Jonassen, c'est-à-dire l'application des solutions dans une perspective itérative. Dans le même ordre d'idée, et afin de faciliter la compréhension, nous proposons un tableau synthèse du processus de RP et des processus cognitifs associés, permettant ainsi une vision intégrée de la RP en fonction de la littérature (tableau 3).

Étapes du processus de RP selon Polya (1957)	Étapes du processus de RP selon Jonassen (2014)	Processus cognitifs mobilisés en RP (OCDE, 2015)
1. Comprendre le problème	1. Définition du problème	Exploration et compréhension
	2. Analyse du problème	Représentation et formulation
2. Faire un plan	3. Collecte des données	Représentation et formulation
	4. Élaboration de pistes de solution	Planification et exécution
	5. Évaluation de solutions alternatives	Planification et exécution
3. Exécution du plan	6. Appliquer les solutions	Planification et exécution
4. Évaluer l'efficacité du plan	7. Vérification des solutions	Suivi et vérification
		Suivi et vérification

Tableau 3. – Tableau synthétique intégrateur de la RP

Or, il existe aussi dans la littérature plusieurs ouvrages abordant la résolution de problèmes complexes (RPComp), cette dernière prenant place principalement dans des conditions réelles, c'est-à-dire dans la vie de tous les jours. Funke (2010) identifie trois façons de concevoir la RPComp:

« (1) as a paradigm to study cognition under real-life conditions (with different foci such as learning, knowledge acquisition, and decision-making), (2) as a descriptor of behavior exhibited while dealing with a certain class of problems usually presented on a computer, and (3) as an ability construct that is related to intelligence » (p. 135)

Quant à eux, Marquardt et Yeo (2012) associent la RPComp à la résolution d'un problème en soi et à l'anticipation de problèmes potentiels causés par l'application de la solution choisie. Ils précisent : « the complexity therefore lies in the uncovering of the more deeply rooted symptoms of a problem » (p. 10). Ils situent la RPComp dans les conditions particulièrement changeantes et contextuelles du marché du travail du 21^e siècle. Qui plus est, ils définissent les problèmes complexes comme étant des problèmes connectés intrinsèquement aux racines d'autres sous-problèmes, obligeant ainsi l'individu à procéder méthodiquement en traitant individuellement chacune des couches (*layers*) du problème (Marquardt et Yeo, 2012).

En somme, il a été possible de constater que la RP est en fait un processus se divisant en plusieurs étapes, qui renvoient à différents processus cognitifs. La complexité de ce processus ne saurait être remise en question, selon la littérature. Il est pertinent ici de souligner que la compétence en RP a été fortement associée à la programmation et au développement de la pensée algorithmique (Modeste, 2012; Wing, 2006), comme c'est également le cas pour l'utilisation de dispositifs robotiques (Komis et Misirli, 2013). En revanche, lorsqu'il est question des finalités de la RP, le contexte dans lequel elle se déroule semble jouer un rôle majeur, faisant ainsi d'elle un processus fondamentalement contextuel. L'apport des pairs dans la mobilisation, voire le développement, de compétences nous amène à considérer ce processus de façon collaborative.

2.2.2 La collaboration

La composante collaborative de cette étude est certainement un facteur qui détermine son ancrage dans le socioconstructivisme (Vygotsky, 1934). Globalement, le socioconstructivisme met l'accent sur l'utilisation du langage dans l'apprentissage et le développement. Qu'il soit oral, écrit

ou intérieur, il nécessite différents niveaux d'abstraction et amène l'apprenant « à prendre conscience de ce qu'il fait et, par conséquent, à utiliser volontairement ses propres savoir-faire » (Vygotsky, 1997, p. 345). On y évoque la notion de développement global, qui succède à l'apprentissage et qui n'est pas segmenté en fonction des matières scolaires. Le niveau de développement de l'enfant peut varier et le postulat principal de Vygotsky est qu'en collaboration avec quelqu'un (un adulte ou un pair), l'élève est capable de résoudre des problèmes plus complexes que lorsqu'il est seul. C'est là le concept de *zone proximale de développement*¹⁷ (Vygotsky, 1934).

« La possibilité plus ou moins grande qu'a l'enfant de passer de ce qu'il sait faire tout seul à ce qu'il sait faire en collaboration avec quelqu'un est précisément le symptôme le plus notable qui caractérise la dynamique de son développement et de la réussite de son activité intellectuelle. Elle coïncide entièrement avec sa zone prochaine de développement » (Vygotsky, 1997, p. 353).

La théorie de l'activité (Leontiev, 1978), qui s'inscrit dans la « tradition historico-culturelle de Vygotsky » (Kaptelinin et Nardi, 2006, p. 30), est particulièrement pertinente dans le cadre de notre recherche, tout comme c'est le cas dans de nombreux autres travaux portant sur les interactions médiatisées par l'ordinateur (voir notamment George, 2001; Hodgson et al., 2012). Le modèle de la théorie de l'activité d'Engeström (1987) présente dans un premier temps le cœur du concept d'activité, c'est-à-dire l'interaction entre le sujet (personne) et l'objet (le but, par ex. : créer un programme) (figure 5). Cette interaction est médiatisée par l'outil, qui peut être concret (par ex. : ordinateur) ou abstrait (opération mentale).

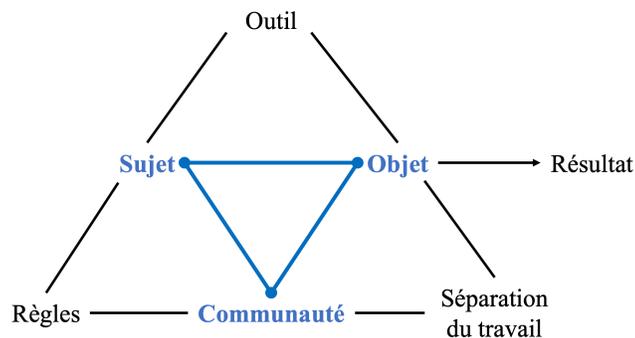


Figure 5. – Modèle de la théorie de l'activité (Engeström, 1987).

¹⁷ Le terme utilisé dans l'ouvrage traduit est *zone prochaine de développement* (Vygotsky, 1997, p.351).

Cela dit, il faut considérer cette interaction au sein de l'environnement du sujet, d'où la nécessité d'ajouter la communauté (par ex. : un ou d'autres élèves). L'ajout de la communauté crée donc deux nouvelles interactions : celle entre le sujet et la communauté, médiatisée par des règles (pour favoriser les interactions optimales), et celle entre la communauté et l'objet, médiatisée par la répartition des tâches, qui renvoie à « l'organisation explicite et implicite d'une communauté en lien avec le processus de transformation de l'objet en résultat » (Kuutti, 1996, p. 28). L'ensemble de ces interactions au sein de l'activité se déroule à trois niveaux hiérarchiques : l'activité, l'action et l'opération (Jonassen et Rohrer-Murphy, 1999). L'activité est composée d'un ensemble d'actions alors que chacune des actions est composée d'un ensemble d'opérations. Par exemple, dans le cas d'une activité de programmation, l'activité pourrait être la création d'un programme, l'action pourrait être de programmer un module et l'opération pourrait être l'utilisation d'une fonction particulière dans le module (Kuutti, 1996). La théorie de l'activité, qui s'est avérée utile pour l'analyse de l'apprentissage (Teasley et Roschelle, 1993), de même que le paradigme au sein duquel s'ancre cette recherche contribuent à préciser comment la collaboration s'inscrit dans cette thèse et pourquoi elle représente une partie importante de notre objectif de recherche.

La collaboration naît donc d'une interaction orientée vers l'objet entre un sujet et un ou plusieurs autres. Selon Henri et Lundgren-Cayrol (2001), la confrontation des points de vue permet de développer la compréhension du monde. L'utilisation des TIC en contexte pédagogique offre d'ailleurs un environnement dans lequel l'apport des autres est une composante majeure (Depover et al., 2007). Or, deux termes s'imposent lorsqu'il est question de l'apport des autres dans l'apprentissage : la coopération et la collaboration. Malgré leur rapprochement sémantique, la coopération et la collaboration possèdent des différences intrinsèques qui les distinguent de façon significative; c'est d'ailleurs le constat auquel arrive George (2001). Dans son effort de les distinguer, il propose cet exemple :

« Imaginons que trois personnes soient engagées dans une activité collective qui consiste à construire un mur de pierre. Si chaque personne va chercher des pierres sur un tas pour les poser sur le mur en construction, ces personnes coopèrent. Le mur se construit par la somme des actions individuelles distinctes (chacun transportant ses pierres), actions coordonnées entre elles (poser correctement les pierres les unes par rapport aux autres) dans le but d'atteindre l'objectif final (construction du mur).

Maintenant, si une pierre est trop lourde et que ces personnes décident de s'y mettre à trois pour la transporter, nous sommes en présence d'une collaboration, une action collective étant entreprise » (George, 2001, p.49).

Ainsi, le terme coopérer renvoie à une division des tâches dans le but de réaliser un projet commun, où tous les membres ont leur propre tâche spécifique à accomplir (Henri et Lundgren-Cayrol, 2001). Quant au terme collaboration, il fait plutôt référence à un travail qui se réalise entièrement de façon commune et où le contrôle de l'enseignant s'estompe pour laisser plus de place à l'autonomie des apprenants (Henri et Lundgren-Cayrol, 2001). Il est à noter que l'autonomie est au cœur de l'approche vygotskienne¹⁸. Cette approche a orienté les décisions prises en amont de l'étude, notamment à l'égard de la méthodologie, c'est-à-dire quant aux façons d'observer la mobilisation d'habiletés de RCP lors d'activités de programmation.

L'autonomie que permet la collaboration est cohérente avec le contexte d'utilisation de la programmation, où l'autonomie est essentielle (Depover et al., 2007). Nous nous référons à la conceptualisation de la collaboration de Hesse *et al.* (2015), qui identifient trois composantes de la collaboration : la participation, la prise de position et la régulation sociale. D'abord, la participation détermine le niveau d'engagement de l'élève dans la tâche et l'interaction avec le ou les autres membres de l'équipe. On reconnaît la participation notamment par la capacité de l'élève à être actif lors des tâches, à interagir avec ses coéquipiers et à mener à bien une tâche. La prise de position, quant à elle, détermine la façon dont les élèves vont affirmer leurs points de vue tout en étant à l'écoute de celui des autres. Cette prise de position se manifeste tant par la réaction de l'élève aux contributions d'un pair que par son habileté à s'adapter pour favoriser le fonctionnement de l'équipe. Enfin, la régulation sociale fait référence à la gestion des conflits et au maintien d'un climat de travail approprié, tout en incluant l'évaluation de la performance des autres membres de l'équipe. Cela implique que l'élève est en mesure de régler les différends à l'aide de compromis, d'évaluer sa performance ou celle d'un pair de façon réaliste et de faire preuve de leadership. Bref, la compétence « collaborer » a été retenue en raison du potentiel collaboratif inhérent à l'utilisation des TIC (Depover et al., 2007), sans oublier son apport majeur à la résolution de problème (Hesse et al., 2015). Le modèle de Hesse *et al.* est particulièrement

¹⁸ Voir le concept de zone proximale de développement.

bien adapté à l'apprentissage à l'école et permet de comprendre ce qu'est la collaboration en contexte technopédagogique. Il présente des critères empiriques compatibles avec notre méthode d'observation, qui est présentée au troisième chapitre de ce manuscrit. La collaboration a été associée par de nombreux auteurs à la pensée informatique, comme Yadav et al. (2017) qui la considèrent comme une « pratique fondamentale de la pensée informatique »¹⁹ (p.1059) chez des élèves du primaire et du secondaire. Il en va de même pour les outils permettant de développer la PI (des outils numériques variés, incluant la programmation informatique), qui sont très propices à la collaboration (Ching et al., 2018).

2.2.3 La résolution collaborative de problèmes (RCP)

À l'instar de Kamga, Romero, Komis et Misirli (2017), nous mobilisons dans cette recherche les deux compétences présentées dans la section précédente de façon unifiée : la résolution collaborative de problèmes (RCP). Dans le cadre de son enquête PISA 2012, l'OCDE (2015a) a défini la résolution collaborative de problèmes comme une démarche où l'élève partage la compréhension du problème avec d'autres élèves en vue de le résoudre, et ce, en mettant en commun leurs connaissances, leurs compétences et leurs efforts. Cela rejoint les propos de Hesse et al. : « participants need to exchange and share their identification of parts of the problem [...]. The steps towards a collaborative solution may be coordinated through the use of verbal and non-verbal observable signals » (2015, p. 39).

Il a également été suggéré que, dans un contexte de RCP, « les apprenants deviennent des ressources puisqu'ils peuvent communiquer et se coordonner au cours de la démarche de résolution de problèmes » (DeBlois et al., 2018, p. 3). En nous appuyant, d'une part, sur les définitions distinctes de la RP et de la collaboration, et d'autre part, sur la définition de la RCP, nous en arrivons à une matrice conçue par l'OCDE (2017), dans le cadre de son étude PISA 2015, fusionnant ces deux compétences pour créer la RCP (tableau 4). Les rangées de la matrice représentent les processus cognitifs mobilisés en résolution de problèmes (A à D) et dans les colonnes, nous retrouvons les trois habiletés majeures en RCP (1 à 3 dans le tableau), c'est-à-dire l'établissement et le maintien d'une compréhension partagée, la prise de décision pour résoudre

¹⁹ Traduction libre (*core computational thinking practice*).

le problème, ainsi que l'établissement et le maintien d'une organisation d'équipe. Ces habiletés composent le modèle de collaboration de Hesse et al. (2015).

	(1) Établir et maintenir une compréhension partagée	(2) Réaliser les actions appropriées à la résolution du problème	(3) Établir et maintenir l'organisation de l'équipe
(A) Exploration et compréhension	(A1) Découvrir les points de vue et les habiletés des membres de l'équipe	(A2) Découvrir le type d'interaction collaborative pour résoudre le problème et atteindre les buts	(A3) Comprendre les rôles nécessaires à la résolution du problème
(B) Représentation et formulation	(B1) Construire une représentation partagée du problème et en négocier le sens (espace partagé)	(B2) Identifier et décrire les tâches à réaliser	(B3) Décrire les rôles et l'organisation d'équipe (protocoles de communication, règles d'engagement)
(C) Planification et exécution	(C1) Communiquer avec les membres de l'équipe au sujet des actions entreprises ou à entreprendre	(C2) Réalisation des plans	(C3) Suivre les règles d'engagement (par exemple, inviter les autres membres à réaliser leurs tâches)
(D) Suivi et vérification	(D1) Effectuer un suivi et rétablir la compréhension partagée	(D2) Effectuer un suivi des résultats des actions et évaluer le succès de la résolution du problème	(D3) Effectuer un suivi, offrir une rétroaction et adapter l'organisation d'équipe et les rôles

Tableau 4. – Matrice de la RCP (OCDE, 2017)

La matrice de l'OCDE se veut donc une synthèse de ce que nous avons abordé dans les précédentes sections : la résolution de problèmes et la collaboration sont rapprochées de façon à créer des habiletés hybrides composant la RCP. Ce rapprochement appuie notre décision de cibler ces deux compétences transversales dans notre objectif de recherche. En effet, en plus d'être cohérentes l'une par rapport à l'autre, elles forment un tout qui fut étudié dans des contextes technopédagogiques, notamment avec des étudiants à la formation des maîtres au Québec (Kamga et al., 2017). L'intérêt grandissant envers la RCP, qui est partie intégrante des compétences du 21^e siècle selon plusieurs modèles (voir notamment Romero et Vallerand, 2016), corrobore la pertinence de notre objectif de recherche.

2.3 Distinction entre développement et mobilisation de compétences

Dans cette étude, nous avons choisi sciemment de nous concentrer sur la mobilisation de la RCP plutôt que sur son développement. Il nous apparaît pertinent à ce point-ci d'explicitier les raisons qui nous ont amené à faire un tel choix. Citons d'abord Legendre (2005), qui définit le développement de la façon suivante :

« Processus d'acquisition ou de changement, dynamique et interne à une personne, laquelle, mue par le désir et la volonté de développement, construit de nouvelles représentations explicatives cohérentes et durables de son réel à partir de la perception de matériaux, de stimulations de son environnement, de l'interaction entre les données internes et externes au sujet et d'une prise de conscience personnelle » (2005, p. 88)

Ainsi, en plus d'être partie intégrante de l'apprentissage, le développement est considéré à long terme et *en profondeur* (Liben, 1987; Tardif, 2006). C'est ainsi que le concept de développement se voit attribuer des épithètes comme profondeur, pérennité et irréversibilité (Tardif, 2006). La notion de temporalité est aussi déclinée dans une perspective évolutive, où le développement est considéré comme une progression vers un niveau supérieur (Legendre, 2005), un cheminement dans lequel il y a une croissance, une extension (LeRobert, 2016). Il est donc clair que le développement doit être considéré dans une perspective à long terme. Or, l'étude du développement aurait exigé une période de collecte de données s'étendant sur une période relativement longue, ce qui n'a pas été la méthode retenue dans le cadre de cette recherche doctorale.

2.3.1 Enquête PISA 2015

En outre, le choix de porter notre intérêt sur la mobilisation des compétences de RCP s'explique aussi par la nécessité de mieux comprendre son opérationnalisation (le *comment*), contrairement à son développement (mesure), dans un contexte pédagogique et dans une dynamique d'humain à humain (H-H). Dans l'enquête PISA 2015, un volet consistait à mesurer la compétence de RCP en amenant les élèves à compléter une série de tâches dans une interface numérique (figure 6).

PISA 2015 Unit name: The Aquarium Time remaining: 13 minutes ?

Task 4 of 7
You and Abby have 5 trials to find the best conditions for the fish to live in the aquarium.

CHAT

Abby
Let's change the scenery again. The results were much better for rocky scenery.

You

- You're right. I'll change it back
- Why do you think that?
- No, the results with plant scenery was better

Control panel

Water type: Fresh Sea
 Scenery: Rocky Plants
 Lightning: Low High

Abby's control panel

Food type: Dry Food blocks
 Fish: Few Many
 Temperature: Low High

Tryout conditions



Results

Bad OK Great

Results: these conditions are suitable, but they can be better.

Figure 6. – Interface de la tâche *The Aquarium* (OCDE, 2017)

Répondant aux contraintes imposées par la passation d'une évaluation à l'échelle internationale (automatisation de l'évaluation, constance à travers les pays, etc.), les concepteurs ont mis en place certaines conditions : (1) une interaction simulée humain à agent (H-A) avec un ou des agents virtuels (Abby, dans la figure 6, est un agent virtuel), (2) des interventions prédéfinies tant pour l'humain que l'agent qui peuvent être sélectionnées par l'utilisateur (le curseur dans la figure 6 sélectionne l'un des trois choix de réponse) et (3) des événements programmés et générés automatiquement.

Il est à noter que cette approche tente de mesurer la RCP, mobilisée entre humains (H-H), à l'aide d'une méthode ne permettant que des interactions entre un humain et un agent virtuel (H-A). Répondant à des critiques formulées à cet égard (voir notamment Zehner et al., 2019), Stadler, Herborn, Mustafic et Greiff (2020) ont publié un article confirmant la validité des tâches de RCP

utilisées dans l'enquête PISA 2015. Ils arrivent à la conclusion que les tâches entre un élève et un agent virtuel (H-A) et celles entre deux humains (H-H) avaient des dynamiques similaires. De plus, ils écrivent : « we found that performance in H-A tasks was moderately related to more conventional assessments of individual CPS skills ». Or, ces résultats s'opposent à ceux d'autres travaux. Par exemple, selon Rosen et Foltz (2014) l'interaction avec des agents virtuels (H-A) crée des situations artificielles qui ne sont pas cohérentes avec l'authenticité d'une interaction humaine (H-H), cette dernière étant composée de changements constants et d'idiosyncrasies. D'autres auteurs encore adoptent une position intermédiaire en soulignant la haute validité des interactions H-H, mais en rappelant la difficulté à contrôler ces interactions dans certains contextes comme une enquête internationale de grande envergure (Greiff et al., 2013); ils désignent l'interaction H-A comme étant un « compromis acceptable ».

Malgré les réserves pouvant être exprimées à l'égard de la posture adoptée dans l'enquête PISA 2015, il demeure que cette dernière est la source principale de données empiriques associées à la RCP dans le monde. Par ailleurs, Stadler et ses collègues (2020) évoquent la nécessité pour de futures études de s'intéresser de façon plus précise à la dimension sociale de la RCP, en contexte H-H.

2.3.2 De la nécessité d'étudier les aspects inhérents à la RCP

Nos travaux ont pour but de comprendre la mobilisation de la RCP. La littérature laisse entrevoir qu'il existe un déficit d'information quant aux mécanismes intrinsèques de la RCP, ce qui nous a amené à penser, dans un souci de pertinence scientifique, qu'il était plus intéressant de nous pencher sur la RCP en application dans un contexte scolaire plutôt que d'en étudier le développement. C'est d'ailleurs ce qu'ont souligné Graesser et al. (2018), en évoquant les faibles résultats des apprenants lors des évaluations en RCP à l'international : « this calls for an analysis of CPS mechanisms, frequent problems, and methods of solving these problems » (p.86). De plus, dans le cadre de cette recherche doctorale, il paraissait artificiel de mesurer le développement de la compétence en RCP à la lumière des informations repérées dans la littérature, notamment en ce qui a trait au concept de développement et de ses implications en termes de durée.

En conclusion, nous retenons que les compétences transversales sont des savoirs-agir qui dépassent les limites des disciplines. Elles doivent être considérées de deux façons : d'une part comme étant elles-mêmes des compétences à développer, et d'autre part, comme étant un vecteur pour le développement des compétences propres à chacune des disciplines. Le développement s'effectue à moyen et long terme, rendant ainsi son observation à court terme peu révélatrice. Néanmoins, il est possible à court terme d'observer la mobilisation, qui précède le développement : c'est pour cette raison que nous avons choisi d'observer la mobilisation de compétences dans le cadre de cette étude. Par ailleurs, l'utilisation du numérique en contexte de résolution collaborative de problèmes ainsi que son potentiel pédagogique ont été abordés dans plusieurs recherches (Arts et al., 2006; Hurme et Järvelä, 2005; Zumbach et al., 2004). Il existe cependant une zone grise dans la recherche concernant l'utilisation de la programmation chez les novices, ce qui correspond à la situation d'élèves du primaire qui apprennent à programmer : « more research is needed to examine the utility and efficacy of block-based programming environments for novice programmers » (Xu et al., 2019, p. 177). Qui plus est, il a été avancé à de nombreuses reprises que la programmation semble être des plus propices à la mobilisation des compétences en résolution de problème et en collaboration. Ces deux compétences constituent les fondations de nos objectifs spécifiques de recherche.

2.4 Objectifs spécifiques de recherche

À la lumière des informations présentées dans ce chapitre et le précédent, l'objectif général est de décrire et comprendre l'incidence de la programmation informatique sur la mobilisation des habiletés de résolution collaborative de problèmes (RCP) d'élèves du primaire. Nous avons décliné cet objectif en trois objectifs spécifiques qui ont permis, dans leur complémentarité, d'apporter un regard nouveau sur ce phénomène :

- **Objectif spécifique 1 (OS1) :**

Déterminer les pratiques effectives de programmation visuelle d'élèves du primaire lors d'activités de programmation visuelle et tangible;

- **Objectif spécifique 2 (OS2) :**

Créer et tester une grille d'observation de la résolution collaborative de problèmes (RCP) en contexte de programmation visuelle et tangible;

- **Objectif spécifique 3 (OS3) :**

Décrire et comprendre comment sont mobilisées les habiletés résolution collaborative de problèmes (RCP) en contexte de programmation visuelle et tangible en équipe.

Le premier objectif spécifique nous permet de comprendre ce que font les élèves lors des activités de programmation visuelles et tangibles. Il s'agit en quelque sorte du *quoi* de l'activité de programmation. Puis le second objectif spécifique est la conception d'un outil qui nous permettra, dans le troisième objectif spécifique, de comprendre le processus de résolution collaborative de problèmes inhérent à l'activité de programmation qui se déroule pendant que les élèves mettent en œuvre ces pratiques (OS1). Il s'agit ici du *comment*.

Chacun de ces trois objectifs spécifiques a fait l'objet d'un article présenté dans cette thèse par articles. Nous abordons dans le chapitre suivant la méthodologie utilisée pour mener à bien ces objectifs spécifiques de recherche.

Chapitre 3 - Méthodologie

Les objectifs de recherche énoncés dans le chapitre précédent sont les jalons qui guident cette recherche. La méthodologie est l'élément nous permettant d'assembler toutes les pièces du casse-tête. En effet, selon Savoie-Zajc et Karsenti (2018), alors que la recherche peut être séparée en quatre grandes questions²⁰, la méthodologie est représentée par la troisième : comment faire une recherche? Ainsi, ce chapitre présente la méthodologie employée (le *comment*) en vue de répondre à nos objectifs spécifiques de recherche. Nous présentons d'abord le type de devis de recherche choisi, puis les cas (sélection et présentation), les participants, la collecte des données et les instruments. Enfin, nous abordons le mode d'analyse des données recueillies ainsi que les forces et limites inhérentes à notre méthodologie.

3.1 Type de recherche

Notre but, dans cette étude de cas multiples (R. E. Stake, 1995), est d'observer un phénomène dans des contextes authentiques en étant guidé par nos objectifs de recherche. Plus précisément, il s'agit d'une étude de cas instrumentale, c'est-à-dire que nous mettons « l'accent sur une question ou sur une préoccupation et [choisissons des] cas pour l'illustrer » (Fortin et Gagnon, 2016, p. 198). La préoccupation de cette recherche est de décrire et comprendre les impacts de la programmation visuelle et tangible sur la sollicitation des composantes de la RCP. Quant aux cas, il s'agit de six classes de trois écoles du Québec où une série d'activités de programmation nommée *Deviens un maître NAO* (Karsenti et al., 2019a) a été mise en place pour des élèves du primaire. En cohérence avec la structure de l'étude de cas multiples, nous procédons à l'aide d'une démarche inductive, c'est-à-dire de façon exploratoire dans un but de compréhension (Karsenti et Demers, 2018).

Selon Paillé (2007), l'étude de cas se subdivise en six étapes : « 1) le choix du ou des cas critiques, uniques ou révélateurs, 2) le choix de la méthode de collecte des données, 3) le choix de la

²⁰ Les quatre grandes questions sont : « Pourquoi faire une recherche?; Que faire dans une recherche?; Comment faire une recherche? et Quels sont les résultats d'une recherche? » (Savoie-Zajc et Karsenti, 2018, p. 139).

méthode d'analyse des données, 4) l'étude [des] cas en soi, 5) la mise en évidence des aspects les plus distinctifs et/ou les plus instructifs [des] cas, et enfin 6) le tracé des implications théoriques et/ou pratiques des résultats » (p. 146). Les étapes 1 à 3 relèvent de la méthodologie alors que les autres appartiennent davantage à l'analyse des résultats. De fait, en nous appuyant sur la structure suggérée par Paillé, nous abordons, dans les prochaines sections, la première étape de l'étude de cas en décrivant les participants de notre étude (3.2 et 3.3), la seconde étape en déclinant les étapes de la collecte de données (3.4) ainsi que les instruments de collecte (3.5), et enfin, la troisième étape en traitant du mode d'analyse des données collectées (3.6).

3.2 Sélection des cas

L'utilisation de la programmation pédagogique pour la mobilisation de compétences transversales à l'école primaire est l'occasion d'aborder un aspect peu étudié et qui reste encore à approfondir. Cet ordre scolaire s'avère un contexte particulièrement pertinent pour l'observation du phénomène qui nous intéresse puisque la proportion d'études s'intéressant aux élèves du primaire est relativement faible au sein de l'ensemble des études menées sur la programmation à l'aide de dispositifs robotiques en contexte scolaire. Les trois cycles d'enseignement du primaire, chacun composé de deux années, offrent ensemble six niveaux différents pour le déroulement de notre étude. La population visée par cette recherche est donc constituée d'élèves du primaire. Nous choisissons de situer notre travail au Québec non seulement dans une visée de cohérence, puisque cette recherche est ancrée dans une approche par compétences au cœur du système scolaire québécois tel qu'il est au moment de la réalisation de cette recherche, mais aussi dans une visée pratique, alors que la proximité de la population permet un processus de recherche plus efficient. Les élèves de trois écoles primaires au Québec, que nous appellerons les écoles A, B et C, constituent donc la population accessible, c'est-à-dire la portion de la population à laquelle nous avons eu accès dans le cadre de cette recherche (Fortin et Gagnon, 2016). LeCompte et Preissle (1993) parlent d'un *groupe naturel* pour définir un collectif qui existe en dehors du contexte de la recherche, comme dans le cas qui nous occupe, c'est-à-dire des groupes-classes du primaire.

3.2.1 Échantillon

Nous avons déterminé un échantillon incluant différents milieux socioéconomiques, assurant ainsi une certaine diversité au sein des groupes de participants. En effet, le statut socioéconomique, parmi d'autres caractéristiques comme l'origine et le genre, peut avoir un effet sur les habiletés et les usages d'outils numériques (Hargittai, 2010). Bien qu'il aurait été intéressant d'adopter une approche sociocritique dans le cadre de ces travaux, ce n'est pas la voie qui a été choisie. À cet effet, Collin et Brotcorne (2019) soulignent que, dans une perspective sociocritique, l'utilisation de la programmation en contexte pédagogique pour atténuer les inégalités numériques a plusieurs limitations. Ainsi, sans prévoir une analyse différenciée en fonction de ces caractéristiques et sans prétendre à un échantillon représentatif de la population ciblée, nous avons néanmoins cherché à en diversifier la composition. D'ailleurs, considérant la nature du phénomène que nous souhaitons observer, qui est fondamentalement ancrée dans le contexte au sein duquel ce phénomène se manifeste, nous choisissons d'effectuer un échantillonnage par contraste-approfondissement (Pires, 1997). Toujours selon Pires, ce type d'échantillonnage implique la juxtaposition de plusieurs cas dans le but d'obtenir davantage d'informations ou encore d'avoir un portrait global plus précis.

Les écoles A, B, et C, dans le cadre de notre recherche, représentent des corpus empiriques nous permettant d'étudier en profondeur ce qui s'y déroule (Pires, 1997). Notre échantillon, non-probabiliste de type intentionnel (Fortin et Gagnon, 2016), est composé de six classes. Malgré les nombreuses réserves pouvant être exprimées à l'égard d'un échantillon non probabiliste, Deslauriers et Kérisit (1997) soutiennent qu'il demeure néanmoins un procédé utile dans la mesure où il « donne accès à une connaissance détaillée et circonstanciée de la vie sociale » (p. 97). Ils poursuivent en affirmant que c'est « au regard des résultats auxquels il donne lieu et de sa pertinence que l'échantillon non probabiliste se justifie » (Deslauriers et Kérisit, 1997, p. 97).

3.2.2 Processus de recrutement

La première étape du processus de recrutement des participants a été l'obtention de l'aval de l'unité administrative des commissions scolaires concernées. Une fois l'autorisation reçue, nous avons contacté les enseignants des classes pour établir la logistique du déroulement de la collecte

des données. Enfin, une fois ces deux étapes franchies, nous avons contacté les participants. Considérant leur âge, nous devons obtenir le consentement des parents ou du tuteur légal pour la participation de l'enfant à notre étude, comme prescrit par l'Université de Montréal (2014). Cette lettre constitue la dernière étape du processus d'échantillonnage intentionnel. Notre échantillon n'a pas fait l'objet de critères d'inclusion ou d'exclusion, si ce n'est le retour de la lettre, dûment remplie et signée, signifiant l'approbation des parents ou du tuteur légal pour la participation de l'enfant à l'étude. Cette lettre demande également l'autorisation d'enregistrer la voix ou de filmer leur enfant.

3.3 Présentation des cas

Chacun des cas de cette étude correspond à une classe de l'une des trois écoles de notre échantillon. L'emplacement, le type de pédagogie (école alternative ou «régulière») et le statut socioéconomique des élèves de ces établissements varient grandement. Nous présentons brièvement ici les trois écoles ainsi que les classes participant à l'étude (cas).

3.3.1 Cas #1 : Classe A1

L'école A est une école alternative de la région de Montréal. Située en milieu urbain, cette école accueille des élèves dans des classes multiniveau, c'est-à-dire qu'elles sont composées d'élèves d'âges et de niveaux scolaires différents. Lors des activités de programmation, les élèves (n=29) étaient installés dans différentes salles : une grande salle multifonction où se trouvaient quatre équipes, puis trois autres locaux pour les trois autres équipes. En plus de l'équipe de recherche composée de quatre personnes encadrant le déroulement de l'activité, deux enseignants de l'école étaient présents pour gérer le comportement de certains élèves et les moments de transition (par ex. : début de l'activité, départ à la récréation, etc.).

3.3.2 Cas #2 à 5 : Classes B1 à B4

L'école B est une école primaire située à Montréal, en milieu urbain. Cet établissement de la Commission scolaire Marguerite-Bourgeoys s'est vu attribuer les cotes maximales (10) aux indices de seuil de faible revenu et de défavorisation du ministère de l'Éducation et de l'Enseignement supérieur (2020). Cette école est d'ailleurs reconnue pour son utilisation de la programmation,

du préscolaire à la sixième année. Les activités de programmation se déroulaient en une journée au cours de laquelle les quatre groupes visitaient de façon successive la bibliothèque et une salle adjacente, où le matériel était installé en sept stations. En plus de l'équipe de recherche, l'enseignante du groupe était présente pour gérer le comportement de certains élèves et les moments de transition (par ex. : début de l'activité, départ vers la classe, etc.).

3.3.3 Cas #6 : Classe C1

L'école C se situe dans la grande région de Québec, dans un milieu rural. La classe C1 réunit 10 élèves de 5^e année ayant des besoins particuliers. En effet, le nombre d'élèves réduit s'explique par le fait que plusieurs de ces élèves sont en situation de handicap ou ont des difficultés d'apprentissage importantes. L'activité de programmation se déroulait dans les classes aménagées de façon à répartir le matériel afin que chaque équipe puisse avoir son espace. En plus de l'équipe de recherche, l'enseignante du groupe et un technicien en éducation spécialisée étaient présents pour gérer le comportement de certains élèves et les moments de transition.

Comme nous l'avons évoqué, nous avons procédé à un échantillonnage par contraste-approfondissement. Cette méthode permet de tirer profit de la complémentarité des différents cas observés dans une étude de cas multiples.

École	Milieu	Groupe	Niveau scolaire	<i>n</i>
École A	Urbain	A1	Multiniveau	29
École B	Urbain	B1	4 ^e année	14
		B2	5 ^e année	17
		B3	5 ^e année	20
		B4	6 ^e année	19
École C	Rural	C1	5 ^e année	10

Tableau 5. – Tableau synthèse des participants (par cas)

Nous faisons ainsi appel à la notion de diversification interne (Pires, 1997) : tout en misant sur l'homogénéité de notre échantillon (élèves du primaire d'écoles du Québec), nous le diversifions en ciblant certaines caractéristiques des participants. Dans le cadre de cette étude, la diversité interne se manifeste notamment par la présence de différents niveaux scolaires (de la 4^e à la 6^e

année), par le milieu dans lequel se trouvent les établissements (rural et urbain) et par les caractéristiques des apprenants (les handicaps, difficultés d'apprentissage, etc.).

3.4 Processus de collecte de données

Afin d'étudier la RCP dans des activités de programmation visuelle et tangible, nous avons collecté des données à différents moments de la présence sur le terrain, qui a débuté au mois de septembre 2019. Quelques contraintes ont dû être considérées en ce qui concerne la présence sur le terrain : les vacances d'hiver, la nécessité d'espacer les séances pour que les élèves puissent avoir suffisamment de temps pour réinvestir leurs nouvelles habiletés dans les activités régulières de la classe et, dans le cas des classes de sixième année, les examens du ministère en fin d'année qui nécessitent une préparation considérable, rendant ainsi notre présence peu appropriée aux mois d'avril à juin. En bout de ligne, nous avons atteint une saturation des données peu après les vacances d'hiver, ce qui représente donc une collecte d'une durée approximative de quatre mois (septembre 2019 à janvier 2020, inclusivement). La saturation des données survient lorsque la collecte ne mène plus à l'obtention de nouvelles données (Fortin et Gagnon, 2016). Ci-après, nous présentons les activités de programmation qui ont été réalisées par les élèves en déclinant ses trois grandes composantes : le robot NAO, le logiciel *Choregraphe* et le scénario pédagogique *Deviens un maître NAO*.

Nous avons choisi d'utiliser ce scénario pédagogique, de même que les outils qui l'accompagnent (robot NAO et logiciel *Choregraphe*), pour deux principales raisons. Premièrement, comme nous avons conçu le scénario et que nous l'avons mis à l'essai à de nombreuses reprises dans plusieurs écoles du Québec au cours des dernières années, nous avons une très bonne connaissance de son déroulement, des défis potentiels rencontrés lors des séances et du comportement général des élèves. Cette expérience antérieure a joué un rôle significatif dans la collecte des données, puisque toute notre attention était portée sur les pratiques effectives de programmation et la mobilisation de la RCP chez les élèves plutôt que sur l'opérationnalisation du scénario, qui implique une quantité considérable de matériel et une installation parfois complexe. Deuxièmement, les différentes fonctionnalités du robot NAO, que nous abordons de façon détaillée ci-après, dépassent largement celles d'autres dispositifs robotiques disponibles dans les

écoles, comme les robots de sol (par ex. : *BeeBot*, *Dash et Dot*, etc.). Le fait de pouvoir activer un plus grand nombre de fonctionnalités permet aux élèves d'effectuer le paramétrage de codes de différentes natures (chiffres [*integers*], lettres [*strings*], musique, angles (°), mouvement, déplacement, etc.). Évidemment, le robot NAO possède également les mêmes avantages que les petits robots de sols, comme la rétroaction immédiate (résultat observable et instantané du programme) et la motivation pouvant être sollicitée chez les élèves.

3.4.1 Le robot humanoïde NAO

Il s'agit d'une activité de programmation tangible, c'est-à-dire que les programmes conçus auront pour but d'animer un dispositif robotique. Nous avons choisi d'utiliser NAO, un robot humanoïde mesurant 58 centimètres et pesant 4,8 kg, qui possède une multitude de dispositifs internes permettant de déterminer son inclinaison dans l'espace, de mesurer sa température interne, d'activer des lumières LED, d'émettre des sons et des paroles en plusieurs langues, de même que des capteurs lui permettant d'interagir avec son environnement. Les capteurs visuels rendent possible la reconnaissance d'objets et de visages, de même que la détection d'obstacles. Le robot possède aussi des capteurs infrarouges qui détectent les objets à proximité et mesurent la distance à laquelle ils se trouvent. Les capteurs tactiles, quant à eux, sont placés à plusieurs endroits stratégiques sur son corps (tête, mains, torse et pieds) et engendrent différentes actions lorsqu'ils sont activés. Qui plus est, il ne s'agit pas d'un dispositif fermé : tous les capteurs peuvent être programmés et paramétrés à l'aide du logiciel de programmation associé. Au demeurant, ce robot est un outil qui peut susciter la motivation des élèves non seulement en raison de ses nombreuses fonctionnalités telles que la parole, la vision et le mouvement, mais aussi en raison de son caractère humanoïde, qui le rend plus attachant. Comparé à d'autres dispositifs robotiques programmables, le robot NAO se distingue par sa complexité et par les possibilités qu'il offre (plusieurs capteurs, mouvements complexes, etc.). Comme nous avons accès à plusieurs de ces robots et que nous avons collaboré à des travaux de recherche antérieurs utilisant ce type de robot, nous avons choisi de tirer profit de notre expérience avec ce dispositif particulier.

3.4.2 Le logiciel Chorégraphe

Le logiciel utilisé pour animer le robot NAO est appelé *Choregraphe* (Aldebaran Robotics, 2014). Celui-ci permet d'utiliser la programmation visuelle, c'est-à-dire qu'il est possible d'utiliser des boîtes que l'on organise à la façon d'un diagramme plutôt que des lignes de codes sous forme textuelle. Il est donc possible pour l'utilisateur de visualiser son programme en un coup d'œil et de constater la logique et la chronologie qui structurent le programme (figure 7).

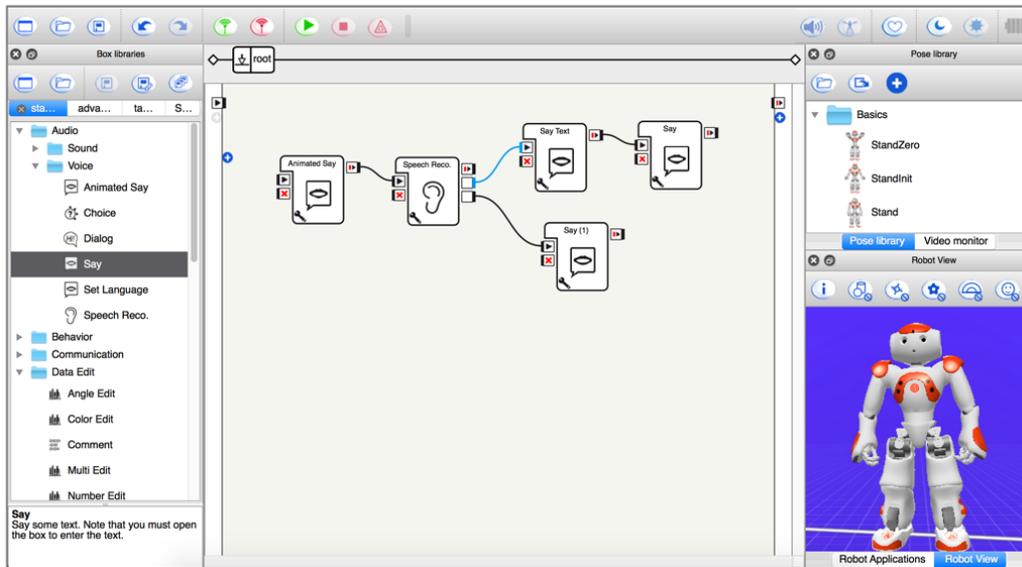


Figure 7. – Interface du logiciel Choregraphe (Aldebaran Robotics, 2014)

Le logiciel est déjà muni d'une bibliothèque de boîtes (ex. : avancer vers..., se lever, etc.). Ces boîtes possèdent un programme interne prédéfini, sous forme textuelle : elles peuvent être programmées par la modification du texte, ou simplement en modifiant dans une interface simplifiée les réglages prédéfinis. Cette dernière option est adéquate pour les élèves du primaire alors que la première, c'est-à-dire la modification des codes internes, nécessite une connaissance de la programmation textuelle. Les activités abordent plusieurs aspects de la programmation visuelle avec *Choregraphe*, ne nécessitant pas la modification de codes, si ce n'est à une seule occasion. Un guide est également offert aux élèves afin de les orienter dans leur exploration du logiciel et dans l'accomplissement des tâches (Karsenti et al., 2019b). Pendant que les élèves en dyades complètent les activités, nous effectuons la collecte de plusieurs données, alors que

d'autres données seront récoltées avant et après la série d'activités. Nous abordons ci-après l'ensemble de nos instruments de collecte et leur utilisation.

3.4.3 Scénario pédagogique *Deviens un maître NAO*

Les élèves ont été appelés, lors des activités, à compléter les exercices de programmation d'un scénario pédagogique nommé *Deviens un maître NAO* (Karsenti et al., 2019a), présenté intégralement à l'annexe 2 de cette thèse. Pour ce faire, les élèves sont placés en équipes, un choix méthodologique expliqué par la nécessité d'observer la composante collaborative de la RCP. Nous avons initialement choisi de limiter le nombre de coéquipiers par équipe à deux élèves. Ce choix était motivé par deux aspects : le premier aspect tenait compte de l'âge des élèves et de leur niveau d'expérience en collaboration. En effet, la coordination de plusieurs points de vue n'est pas une tâche aisée et est souvent suggérée pour des élèves plus âgés (Henri et Lundgren-Cayrol, 2001). Le second aspect tenait compte du potentiel cognitif significatif attribué au travail en dyades, tel que suggéré par plusieurs auteurs (Cesar et al., 2000; Werner et al., 2012). Or, différentes contraintes des milieux, notamment le nombre d'élèves par groupe-classe et le nombre de robots disponibles, ont forcé l'augmentation du nombre d'élèves par équipe. Cela dit, l'observation de la résolution collaborative de problèmes auprès d'équipes de plus de deux personnes a été évoquée dans des écrits théoriques et des études empiriques (Feltovich et al., 1996; Goldman, 1996; Webb et Lewis, 2008). D'ailleurs, le principe de multiplicité, tel que compris par Koschmann, Kelson, Fletovich et Barrows (1996), semble favoriser les équipes de plus de deux personnes. Conférant une certaine richesse à la diversité de plusieurs perspectives, ce principe confirme la pertinence du « développement de multiples représentations via l'exposition aux idées de tous les membres du groupe »²¹ (Koschmann et al., 1996, p. 100).

Contrairement à la plupart des études empiriques dans la littérature consultée, il s'agit ici d'interactions humain à humain (H-H) puisque les élèves travaillent de façon synchrone, et ce, dans le même environnement physique. Le processus de cette activité de programmation est présenté de façon détaillée ci-après (figure 8).

²¹ Traduction libre.

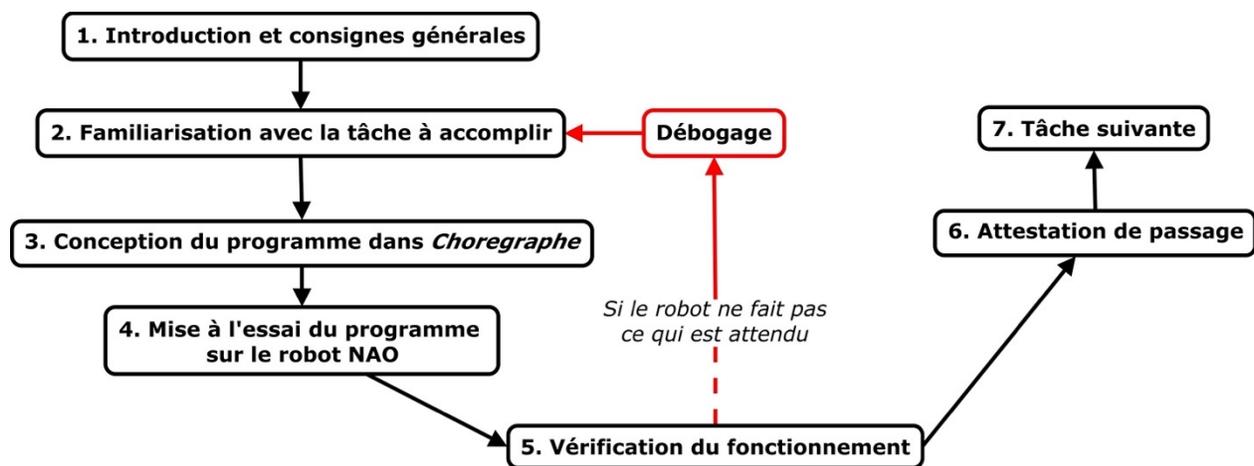


Figure 8. – Démarche générale de l'activité de programmation

Selon Shim, Kwon et Lee (2016), l'enseignement efficace de la programmation²² s'effectue chez des élèves du primaire lorsque trois éléments sont présents : « effective educational tools such as an educational robot, games that can motivate and interest student learning, and a tangible interface suitable for elementary students' cognition » (Shim et al., 2016, p. 168). Le scénario pédagogique *Deviens un maître NAO* rencontre ces trois conditions.

Les activités se sont déroulées à compter du mois de septembre 2019, et ce, pour une durée de quatre mois, c'est-à-dire jusqu'au début du mois de janvier 2020. En effet, comme la recherche qualitative interprétative est un processus itératif, nous avons collecté des données tant et aussi longtemps qu'elles contribuaient à une meilleure compréhension du phénomène que nous observions, c'est-à-dire jusqu'à une certaine saturation des données (Fortin et Gagnon, 2016).

3.5 Instrument de collecte de données

L'instrument qui nous a permis d'y parvenir est l'observation structurée (à l'aide de deux grilles d'observation), aussi appelée observation systématique (van der Maren, 2004). Les objectifs spécifiques de recherche requièrent l'utilisation de deux grilles d'observation fondées sur les différents construits ciblés. Nous avons opté pour une méthode d'observation qui puisse bénéficier à la fois des avantages de l'observation non structurée et de l'observation structurée.

²² Traduction libre (*effective programming education*)

En effet, alors que l'observation non structurée permet au chercheur de s'immerger dans le groupe et d'avoir une meilleure compréhension du phénomène observé, l'observation structurée permet quant à elle d'éviter les risques de perte d'objectivité et permet de noter tous les comportements plus ou moins complexes, ce qui n'est pas toujours possible en observation non structurée (Fortin et Gagnon, 2016). Notre posture se rapproche d'ailleurs beaucoup de l'observation participante, qui implique une participation active du chercheur à la dynamique ambiante sur le terrain (Savoie-Zajc, 2018). Lors de la collecte de données sur le terrain, nous étions accompagné de deux collaborateurs. Dès l'arrivée dans l'école, notre équipe procédait à l'installation et à la préparation du matériel, c'est-à-dire sortir puis mettre en marche les robots, les ordinateurs et les tablettes. Lors des activités de programmation avec les élèves, nous avions avec nos deux collaborateurs un rôle d'intervenant, c'est-à-dire que nous étions dans la classe et répondions aux questions des élèves, et parfois des enseignants. Nous avons ainsi, en tant que chercheur, participé activement à chacune des séances et avons pu comprendre la dynamique au sein de chacune d'elles. En étant actif sur le terrain, bien qu'étant en mesure d'acquérir une compréhension pratique et authentique du contexte d'activité tel que vécu par les participants (Fortin et Gagnon, 2016), nous n'avons pas pu consigner l'ensemble des comportements étudiés. Nous avons donc fait appel à un large appareillage d'observation vidéographiée.

3.5.1 Appareillage d'observation vidéographiée

Les défis inhérents à l'observation de la programmation visuelle et tangible en contexte scolaire sont nombreux. Considérant le nombre d'élèves travaillant de façon simultanée et à plusieurs endroits, l'observation vidéographiée s'est avérée essentielle. Ce type d'observation réfère à l'utilisation de caméras, ce qui permet d'effectuer une observation structurée *a posteriori* :

« in order for collaborative learning processes to be visible to researchers, the participant interaction must be available for careful study and the researchers must be capable of interpreting them appropriately [...] If communications cannot otherwise be captured, such as in face-to-face collaboration, they can be videotaped; the tapes can be digitized and manipulated to aid in detailed analysis » (Stahl, 2006, p. 216).

De plus, comme il s'agit d'une activité de programmation visuelle et tangible, nous devons tenir compte de deux facteurs : 1) les élèves effectuent les opérations de programmation dans l'interface numérique, ce qui implique de voir ce qui se déroule dans l'écran des ordinateurs qu'ils

utilisent, et 2) les tâches impliquent l'animation du robot NAO, ce qui signifie que les élèves doivent se déplacer dans l'espace et que l'appareillage d'observation doit être en mesure de capter l'ensemble des actions posées par les élèves. Enfin, comme nous souhaitons observer la collaboration entre les élèves d'une équipe, nous devons être en mesure d'observer les réactions et les comportements non verbaux, même lorsque les élèves étaient regroupés face à l'ordinateur. Au demeurant, nous avons opté pour un appareillage en trois composantes.

Le premier appareil utilisé est une tablette installée sur un trépied (A). Cette tablette a permis d'observer l'activité d'une équipe d'élèves dans la classe, lorsqu'ils avaient à manipuler le robot par exemple. Cette vue d'ensemble a aussi contribué à une meilleure compréhension de la dynamique d'équipe et des interactions entre les élèves. Le second appareil utilisé était la caméra intégrée de l'ordinateur, située au-dessus de l'écran (B). Celle-ci était dirigée directement sur le visage des élèves qui effectuaient la programmation dans l'interface numérique. Le troisième appareil (C) était l'écran lui-même : à l'aide de la fonction « enregistrement d'écran », nous avons consigné toutes les actions posées par les élèves dans l'interface de programmation. Nous avons donc observé l'activité des élèves sous trois angles distincts et complémentaires. Ces enregistrements vidéos ont par la suite été analysés à l'aide de nos deux grilles d'observation.

3.5.2 Typologie des pratiques effectives de programmation

Cette typologie, inspirée des travaux de Ruf, Berges et Hubweiser (2015), avait pour utilité de répondre à notre premier objectif spécifique de recherche, c'est-à-dire la détermination des pratiques effectives de programmation des élèves.

Type de tâche	Formes de représentation de la connaissance
Écrire le code	Texte → Code Diagramme → Code
Écrire le code en utilisant un autre code	Texte et code → Code
Ajuster, étendre ou compléter le code	Texte et code → Code
Optimiser le code	Texte et code → Code
Déboguer le code	Code (et texte) → Code (et texte)
Définir les bonnes conditions pour le code	Texte et code → Texte
Tester le code	Code → Code
Transformer le code	Code → Code
Tracer/expliquer le code	Code → Texte
Spécifier un problème pour le code	Code → Texte
Dessiner un diagramme pour le code	Code → Diagramme

Tableau 6. – Types de tâches de programmation (Ruf, Berges et Hubweiser, 2015)

Nous avons utilisé les travaux de Ruf *et al.* comme référence, tout en sachant que de nombreux codes seraient inutilisés (considérant la nature des activités proposées à des élèves du primaire) et que d'autres pourraient être créés (codage mixte²³).

3.5.3 Grille d'observation de la RCP

Cette grille, que nous avons conçue dans le cadre de cette recherche doctorale (OS2), répond au troisième objectif spécifique (OS3), qui est de comprendre la mobilisation de la RCP. Le processus entourant la conception de cette grille fait l'objet du second article de cette thèse par articles.

En étant présents et actifs sur le terrain, nous sommes aussi à même d'observer, de façon informelle, l'activité des élèves, ce qui a facilité l'interprétation des données analysées. La conjugaison de ces deux grilles permet d'obtenir un portrait complet du phénomène que nous cherchons à observer, c'est-à-dire la mobilisation de la RCP lors d'activités de programmation visuelle et tangible au primaire.

²³ Voir la section 3.6.1. de ce manuscrit.

3.6 Traitement et analyse des données

Le mode d'analyse des données est d'une importance considérable, puisqu'il détermine la valeur des données qui sont récoltées et surtout des conclusions qui sont tirées. Chaque instrument utilisé dans cette recherche se voit attribuer un mode d'analyse propre qui fait en sorte de tirer le plus d'informations pertinentes. Nous présentons dans cette section le processus d'analyse de données empiriques en trois étapes (Dionne, 2018) : prise de connaissance du matériel en vue d'en dégager un sens général, codage et regroupement des données, ainsi que l'interprétation à l'aide des référents théoriques.

Comme nous avons eu recours à une méthode d'observation participante, nous avons d'emblée une très bonne connaissance et compréhension du corpus de données. La première étape du processus d'analyse relevait donc davantage de l'organisation des différentes composantes du corpus : comme nous avons six groupes dans trois écoles, où nous avons fait pour tous les groupes trois visites, comportant chaque fois trois sources de données (A, B et C)²⁴, nous nous sommes retrouvé avec une quantité considérable de données sous la forme de fichiers vidéos. À la suite de la structuration des fichiers dans le disque dur, nous avons procédé au codage des données : nous avons utilisé le logiciel NVivo (QSR International, 2020), qui permet de faciliter grandement le processus d'analyse de données vidéos (Karsenti, Komis, et al., 2018). Nous avons utilisé deux types de codage : le codage mixte (OS1 et OS2) et le codage fermé (OS3) (van der Maren, 2004).

3.6.1 Codage mixte

Le codage mixte consiste à utiliser une liste de catégories qui est appelée à changer au cours de l'analyse des données : certains éléments peuvent s'ajouter ou être modifiés, voire retirés (van der Maren, 2004). Ce type de codage, qui répond mieux aux exigences de la recherche exploratoire selon van der Maren, nous a permis d'analyser les données relatives à notre premier objectif spécifique, qui vise à déterminer les pratiques effectives de programmation d'élèves du primaire. Ainsi, notre liste originale était composée des indicateurs proposés par Ruf, Berges et

²⁴ Voir la section 3.5.1. de ce manuscrit.

Hubweiser (2015), dont les travaux ont été réalisés dans le cadre de l'enseignement postsecondaire. Nous anticipions donc une certaine adaptation de la catégorisation en cours d'analyse dans le but de mieux refléter la réalité du contexte scolaire à l'enseignement primaire. Lors de l'analyse, nous avons effectivement constaté que certaines pratiques n'étaient pas appropriées : elles ont donc été retirées ou modifiées. Nous n'avons pas fait appel à une validation externe ou à un accord interjuge. La typologie finale est le résultat du premier objectif spécifique, présenté dans le premier article de cette thèse (chapitre 4).

3.6.2 Codage fermé

Les catégories ou rubriques, dans un codage fermé, sont déterminées dès l'élaboration du cadre théorique et méthodologique : « le code fait partie des outils, et l'analyste dispose d'un lexique associant unité de sens et code » (van der Maren, 2004, p. 436). Le codage fermé a été utilisé pour répondre à notre troisième objectif spécifique, qui portait spécifiquement sur la mobilisation de la RCP lors d'une activité de programmation informatique au primaire. Comme nous avons conçu une grille d'observation inédite dans le cadre de notre second objectif spécifique, nous l'avons appliquée intégralement. Comme sa conception a fait l'objet d'un rigoureux processus (Strayer et Gauthier, 1982) validé à l'aide d'un corpus empirique et qu'elle a mené à la formulation d'indicateurs, ainsi que d'exemples et de contre-exemples empiriques, il n'était pas pertinent de procéder à un codage mixte. En effet, la rigueur du processus deux principaux volets : la validation de la grille préliminaire auprès d'experts et le calcul de l'accord intercodeur à deux moments dans le processus. En ce qui concerne la validation, nous avons consulté quatre experts, dont trois sont professeurs en sciences de l'éducation et un chercheur-praticien. Ils ont reçu une version de la grille accompagnée d'un texte expliquant le contexte de notre étude, quelques repères théoriques, ainsi que l'utilisation anticipée de l'outil. Les commentaires de ces experts ont mis en lumière des problèmes relatifs à la formulation des énoncés des indicateurs, qui étaient parfois difficilement observables en termes d'occurrence, et des problèmes de cohérence ou de mauvaise complémentarité entre certains indicateurs. À l'issue de cet exercice de validation, c'est-à-dire après la modification, l'ajout ou la suppression d'indicateurs au regard des commentaires formulés, nous avons produit une seconde version de la grille. Puis, à la troisième phase du processus, nous avons procédé au calcul de l'accord intercodeur à plus d'un moment

dans le processus de codage (Harris et Lahey, 1982). La première mesure a été faite au départ (79,55%) et la seconde a été faite en cours de codage (80,68%), approchant la fin. Cet accord a été calculé en divisant le nombre de segments codés communs aux deux codeurs par le nombre de segments codés total.

3.7 Forces et limites de la recherche

La grande pertinence du phénomène observé constitue une des principales forces de cette recherche. À l'heure où l'utilisation de la programmation en contexte scolaire est adoptée dans les curriculums nationaux de plusieurs pays et où les enseignants sont de plus en plus appelés à l'utiliser dans le cadre de leur enseignement, il semble essentiel de développer les connaissances scientifiques à l'aide d'observations empiriques et ancrées dans des contextes authentiques. Cet ancrage dans les réalités du terrain fait en sorte que les résultats présentés dans les trois articles de cette thèse trouvent une utilité certaine, tant auprès des praticiens que des chercheurs en sciences de l'éducation. De plus, en s'intéressant aux élèves de l'enseignement primaire, cette étude repousse l'idée selon laquelle, dans le sens commun, la programmation serait trop complexe pour des enfants de cet âge. Rappelons que l'utilisation de robots humanoïdes au primaire est particulièrement novatrice : bien que l'utilisation de petits robots programmables ait été observée dans divers contextes (voir notamment Bugmann et Karsenti, 2018b; Komis et Misirli, 2011), il demeure que la valeur ajoutée associée à l'utilisation d'un robot humanoïde est significative.

Une autre force de cette recherche est associée aux données qu'elle mobilise. En effet, à l'aide d'un important dispositif méthodologique et d'un appareillage instrumental varié, nous avons été en mesure de collecter une quantité considérable de matériel vidéo. Cela dit, ce n'est pas tant la quantité de données qui est digne de mention, mais bien la diversité qui les caractérise. L'échantillon de participants a été composé en tenant compte de multiples facteurs : le milieu socioéconomique (urbain ou rural), le type d'enseignement offert (école régulière ou alternative) et les caractéristiques des élèves (difficultés d'apprentissage ou handicaps). Cette étude de cas multiples permet donc une interprétation verticale des données poussées (par cas), de même qu'une complémentarité horizontale (intercas) intéressante.

Or, la quantité de données a aussi mené à l'une des limites de cette recherche : considérant la constitution du corpus, les efforts voués au traitement, à l'analyse puis à l'interprétation des données sont considérables. C'est ce qui explique en grande partie la grande difficulté à mener des entrevues en aval de l'analyse des données d'observation. Comme il s'agit d'une étude de cas multiples fortement ancrée dans une approche interprétative et dont le but est la compréhension du phénomène observé, des entrevues d'autoconfrontation (Mollo et Falzon, 2004) auraient contribué grandement à l'analyse des résultats. La réalisation de ces entrevues aurait exigé, dans un premier temps, d'analyser le corpus de données comme nous l'avons fait. Puis, à la suite de l'analyse des données, les principales conclusions que nous présentons ci-après dans ce manuscrit auraient été illustrées par différents extraits vidéos du corpus de données (Holmbeck et Shapera, 1999; Powers et al., 1994). Ces extraits auraient ensuite été présentés aux élèves y figurant, afin que ces derniers expliquent leur perception et les raisons, s'il y a lieu, qui expliquent leur comportement. Les entrevues auraient donc permis de trianguler les données puis de préciser l'évaluation et l'analyse des données (Savoie-Zajc, 2018). Cela est particulièrement pertinent, considérant le caractère intersubjectif de la collaboration :

« The interpretation may also be discussed with the actual participants. Collaboration is an intersubjective occurrence and its scientific study requires intersubjective confirmation rather than statistical correlations to assure its acceptability » (Stahl, 2006, p. 216).

En amenant les élèves à se prononcer sur leur propre comportement, les données et les analyses associées auraient été plus poussées. Ce procédé est d'ailleurs très cohérent avec l'approche socioconstructiviste, et plus particulièrement la théorie de l'activité (Cahour et Licoppe, 2010), qui constitue une part importante de notre cadre théorique. Il s'agit certainement d'une piste de recherche qui sera explorée ultérieurement, lorsque le contexte le permettra.

3.8 Présentation des articles

Nous avons choisi de présenter les résultats de cette thèse sous la forme d'articles. Nous sommes d'avis que cette façon de faire permet de faciliter la diffusion des résultats de cette recherche. Des intervenants de milieux scientifiques seront en mesure de consulter ces articles, ce qui est particulièrement intéressant considérant l'importance d'inscrire cette recherche dans une série

d'études complémentaires portant sur le phénomène qui nous intéresse. De plus, comme nous vouons une grande importance aux applications pratiques et concrètes pouvant émerger de cette thèse doctorale, nous favoriserons le transfert de connaissance vers les milieux de pratiques à la suite de la parution de ces articles (activités de transfert de connaissances). Ces trois articles répondent chacun à l'un des trois objectifs spécifiques de cette étude de cas multiples et ensemble, ils répondent à notre objectif général. Dans le but d'apporter un regard réflexif et approfondi sur l'utilisation de la programmation à l'école, et plus particulièrement la programmation visuelle et tangible avec des élèves du primaire, nous avons mis en place un important dispositif nous permettant d'observer ce phénomène à divers niveaux de complexité.

Le premier article, intitulé *Pratiques effectives de programmation chez des élèves du primaire : proposition d'une typologie compréhensive et adaptée*, répond à notre premier objectif spécifique de recherche. Celui-ci porte sur les pratiques effectives de programmation des élèves en contexte scolaire d'enseignement primaire. En nous appuyant sur une typologie de Ruf, Berges et Hubweiser (2015), nous avons cherché à concevoir une typologie des différentes opérations de programmation effectuées par des élèves du primaire. De plus, comme la typologie de Ruf et ses collègues émanait d'une analyse de manuels scolaires, nos travaux sont d'autant plus intéressants qu'ils ajoutent une composante empirique à la constitution de cette typologie améliorée. À l'aide d'une observation à la fois participante et systématique, nous avons analysé les enregistrements d'écran des ordinateurs utilisés par les élèves afin de consigner toutes les opérations effectuées par les élèves. À l'issue de ce processus d'analyse, nous avons été en mesure de présenter une typologie compréhensive et adaptée au contexte de l'enseignement primaire.

Le second article, intitulé *Conception d'une grille d'observation de la résolution collaborative de problèmes (RCP)*, présente le processus complexe de conception d'une grille d'observation d'après le modèle de Strayer et Gauthier (1982), qui comporte trois volets : la phase descriptive, la phase exploratoire et l'évaluation systématique. Au terme de ce processus, nous présentons une grille d'observation fondée sur les éléments conceptuels et théoriques de la RCP, et vérifiée empiriquement à l'aide d'une mise en application préliminaire et empirique de la grille. Cette dernière trouve sa pertinence notamment dans son originalité. En effet, la RCP a souvent été étudiée dans la littérature à l'aide de questionnaires à échelles de Likert ou d'interfaces

numériques fermées et automatisées (PISA 2015) : notre approche interprétative et empirique se démarque donc du lot.

Dans le troisième et dernier article, intitulé *Comprendre le processus de résolution collaborative de problèmes (RCP) lors d'une activité de programmation informatique avec des élèves du primaire*, nous utilisons la grille créée dans le second article afin de comprendre le processus de résolution collaborative de problèmes à l'aide de la programmation par blocs et d'un robot humanoïde, en contexte scolaire. Les observations menées dans trois écoles du Québec ont permis de mettre en évidence certaines habiletés de la RCP plus mises en œuvre que d'autres. Elles ont également permis d'explicitier la dynamique entre les composantes internes de chacune des dimensions de la RCP telles que présentées dans la grille. Les retombées des résultats présentés dans cet article sont à la fois pratiques et théoriques. Les résultats offrent un point de départ intéressant à une réflexion portant sur l'étude de la RCP à l'aide d'une méthode interprétative (contrairement à des mesures plus quantitatives, utilisant des équations et des questionnaires à échelles de Likert). Ils contribuent aussi aux réflexions portant sur les impacts de l'utilisation de la programmation en classe et orientent les pratiques enseignantes. En effet, à l'heure où la programmation pourrait devenir partie intégrante du *Programme de formation de l'école québécoise*, il est important de comprendre comment cet outil peut contribuer à l'apprentissage et à la mobilisation, voire au développement, de compétences transversales comme la résolution de problèmes et la collaboration.

Ces trois articles sont proposés de façon complémentaire. Dans le premier article, nous avons cherché à *identifier* (pratiques de programmation mises en œuvre par les élèves). Puis, avec le troisième, nous avons voulu *comprendre* (mobilisation de la RCP). Ensemble, ces deux articles offrent une compréhension empirique du déroulement d'une activité de programmation à l'enseignement primaire. Quant au second article, il a mené à l'élaboration d'une grille d'observation de la RCP, qui a permis d'atteindre cette compréhension sur laquelle le troisième article est fondé. En somme, bien que ces articles soient individuellement autoportants, il demeure néanmoins que leur lecture concomitante offre un portrait particulièrement détaillé et compréhensif du phénomène que nous souhaitons observer dans le cadre de cette recherche.

Chapitre 4 - Pratiques effectives de programmation visuelle chez des élèves du primaire : proposition d'une typologie compréhensive et adaptée

ARTICLE 1

RÉSUMÉ

Dans cet article, nous présentons les résultats d'une étude de cas multiples menée auprès de 109 élèves du primaire au Québec. L'un des objectifs de cette étude était de déterminer les pratiques effectives de programmation visuelle d'élèves du primaire lors d'activités de programmation visuelle et tangible. En plus d'offrir un portrait détaillé des pratiques mobilisées par les élèves dans le cadre de cette recherche, nous proposons une typologie des tâches de programmation visuelle pour des élèves du primaire en nous appuyant d'une part sur la littérature, et d'autre part sur les données empiriques de l'utilisation du dispositif *Deviens un maître NAO*, qui permet aux élèves de mobiliser leurs habiletés en programmant un robot humanoïde appelé NAO. Cette proposition de typologie compréhensive et adaptée a un potentiel pédagogique non négligeable, que ce soit quant à la conception de scénarios pédagogiques mobilisant la programmation visuelle à l'enseignement primaire, ou quant au développement de manuels ou guides pédagogiques destinés aux élèves ou aux enseignants du primaire.

Mots-clés : programmation, typologie, robotique, primaire, pratiques effectives.

ABSTRACT

This article presents the results of a multiple case study of 109 primary school students in Québec. We examined effective coding practices during a learning activity. The students used coding skills by programming a humanoid robot called NAO using the educational challenge *Become a NAO Master*. A detailed portrait of the students' practices is provided. A comprehensive typology of visual programming tasks for primary school students is then proposed, based on the literature and the empirical data on the students' programming practices. This customizable typology has broad educational potential for primary education, including the design of programming lesson plans and the development of manuals and learning guides for primary school students and teachers.

Key words: programming, coding, typology, robot, primary, effective practices

4.1 Contexte

Le 21^e siècle, synonyme de progrès technologique, est le théâtre d'un changement de paradigme dans plusieurs sphères de la société. L'une d'elles est le marché du travail, où l'évolution des méthodes et des procédés amenée par les innovations numériques cause une réorganisation des structures, mais surtout, un changement des compétences attendues des travailleurs actuels et futurs (Forum économique mondial, 2016). Au-delà du secteur professionnel, les nombreux changements technologiques survenus au cours des dernières décennies ont une incidence sur la vie personnelle des citoyens. Avec l'adoption de plus en plus généralisée des appareils numériques comme les téléphones intelligents (Centre facilitant la recherche et l'innovation dans les organisations, 2020), il semble indéniable que cette technologie est là pour rester et qu'elle affecte, positivement ou négativement, la vie de tous. Comment alors faire en sorte de sensibiliser et préparer les citoyens à cette nouvelle conjoncture ? Pensons à l'école, qui semble être le lieu tout indiqué pour amener les apprenants à connaître les outils, à en développer une utilisation responsable et à en avoir une perception critique.

Les compétences du 21^e siècle sont identifiées comme étant l'une des façons d'approcher ce phénomène (Forum économique mondial, 2015). Parmi ces compétences du 21^e siècle, on retrouve la collaboration, la résolution de problèmes, la créativité, la pensée critique et plusieurs autres (Chalkiadaki, 2018). En filigrane de ces nombreuses compétences, on retrouve la programmation (Romero, 2017). En 2019, le Gouvernement du Canada a mis sur pied le Conseil des compétences futures, un groupe ayant comme mandat d'examiner « l'influence des technologies et autres tendances émergentes dans la création de nouvelles possibilités pour les Canadiens [et] de conseiller le ministre de l'Emploi, du Développement de la main-d'œuvre et du Travail sur les priorités régionales et nationales en matière de développement des compétences et de formation » (Gouvernement du Canada, 2020). Le mandat donné au Conseil démontre de façon claire le lien inhérent entre l'innovation technologique et le marché du travail.

La pandémie de COVID-19 a eu notamment pour effet de mettre en lumière les ramifications de ce phénomène. Le caractère particulier de la crise sanitaire a fait en sorte qu'au Québec, comme ailleurs dans le monde, les mêmes secteurs d'activité économique (services nécessitant un

contact en personne notamment) et les mêmes populations (travailleurs peu qualifiés) ont été sévèrement touchés, alors que d'autres l'ont été peu ou beaucoup moins. La perspective de requalifier la main-d'œuvre des secteurs les plus touchés – et dont la reprise complète des activités est incertaine – vers des emplois plus porteurs a maintes fois été soulevée par l'Organisation pour la coopération et le développement économique (OCDE, 2020b). On estime qu'en raison des effets de la pandémie qui bouleversent le monde du travail, les entreprises doivent investir dans le rehaussement des compétences de leur main-d'œuvre, notamment sur le plan des habiletés numériques des travailleurs (Kumar, 2020). L'apprentissage de la programmation et la connaissance du fonctionnement des appareils technologiques offrent aux apprenants un bagage de connaissances pertinentes avec le marché du travail, qui a vu se déployer ce que plusieurs appellent déjà la quatrième Révolution industrielle, où la technologie (intelligence artificielle, robotique, etc.) est omniprésente (Gaspar et al., 2018; Johannessen, 2018; Lee et al., 2018). Le Forum économique mondial, dans un rapport prospectif, affirme que les individus qui auront du succès dans l'économie du futur seront ceux qui peuvent compléter le travail des algorithmes, autrement dit, de travailler avec « les machines » (2018, p. 3). Cela étant dit, notre intérêt pour l'utilisation de la programmation à l'école va bien au-delà de l'impératif économique ou des préoccupations relatives au marché du travail.

Au Québec, bien avant la pandémie de la COVID-19, trois documents officiels révélaient déjà la volonté récente du ministère de l'Éducation et de l'Enseignement supérieur du Québec (MÉES)²⁵ d'intégrer la programmation informatique dans son cursus scolaire. En effet, dès 2018, le Plan d'action numérique en éducation et en enseignement supérieur (Gouvernement du Québec, 2018) présentait le souhait d'« accroître l'usage pédagogique de la programmation informatique » (p. 27). En 2019, le Cadre de référence de la compétence numérique (Ministère de l'Éducation et de l'Enseignement supérieur, 2019), dernière politique en matière de numérique pour notre système d'éducation, y faisait aussi référence : « développer sa pensée informatique, notamment par le développement de sa compréhension et de ses habiletés à l'égard de la programmation informatique » (p. 14). Enfin, plus récemment a été publié le

²⁵ En 2020, le ministère de l'Éducation et de l'Enseignement supérieur (MEES) a été séparé en deux ministères distincts : le ministère de l'Éducation (MEQ) et le ministère de l'Enseignement supérieur (MES).

guide *L'usage pédagogique de la programmation informatique*, dans lequel on souligne le grand potentiel de la programmation informatique, notamment pour « structurer sa pensée », c'est-à-dire pour développer son « raisonnement logique et l'esprit critique » (Ministère de l'Éducation, 2020a, p. 7).

Plusieurs auteurs se sont intéressés à l'utilisation de différents dispositifs avec des élèves du primaire. Par exemple, certains ont documenté les représentations d'élèves de maternelle d'opérations élémentaires de programmation tangible pour faire déplacer le robot BeeBot (Komis et Misirli, 2011), et d'autres ont étudié empiriquement le lien entre la pensée informatique d'élèves et la programmation tangible (Chen et al., 2017; Noh et Lee, 2020). Plus récemment, des auteurs ont observé les habiletés associées à la pensée informatique lors d'activités de programmation en dyades au primaire (Wei et al., 2021). Notre recherche s'inscrit en quelque sorte dans la continuité de ces études réalisées sur l'usage pédagogique de la programmation informatique au primaire. Cela montre en quelque sorte la pertinence scientifique de notre étude, d'autant plus qu'elle a pour objectif d'étudier les pratiques effectives²⁶ de programmation d'élèves novices d'écoles primaires. Cette étude a même permis, comme nous le présenterons dans la discussion, la création d'une typologie adaptée aux élèves en contexte à l'enseignement primaire. Cette dernière peut d'ailleurs s'avérer fort utile pour les enseignants qui souhaiteraient expérimenter la programmation informatique avec leurs élèves.

4.2 Cadre théorique

L'approche socioconstructiviste offre des assises théoriques fort utiles pour l'étude des pratiques effectives de programmation d'élèves du primaire. C'est donc cette approche qui a servi de cadre d'analyse à notre étude. Lev Vygotsky, figure emblématique du socioconstructivisme et père de la théorie historico-culturelle, a souligné l'apport de la collaboration dans l'apprentissage. Le passage de l'apprentissage seul à l'apprentissage avec autrui, et vice-versa, a un effet significatif sur le développement et la réussite de l'activité intellectuelle (Vygotsky, 1934). Cette approche met en lumière les dynamiques inhérentes à la collaboration ; par exemple, Wood, Bruner et Ross

²⁶ Nous entendons par « pratiques effectives » les pratiques que les élèves mettent en œuvre concrètement, dans un contexte authentique.

(1976) avancent que malgré les capacités naturelles de résolution de problème des enfants, il convient de tenir compte de l'apport d'autres pairs plus compétents pour l'assister dans le processus. D'ailleurs, cette disparité des compétences individuelles évoque l'asymétrie de dyades ou d'équipes, référant ici à la notion de conflit sociocognitif, une composante majeure de l'approche socioconstructiviste. Ce conflit représente en fait deux ou plusieurs points de vue différents pour la résolution d'un même problème, se traduisant à terme par un apprentissage par la voie de l'autre (ou des autres). Notons ici que l'intérêt principal de cette confrontation des idées est l'effet sur l'apprentissage, puisque selon Mugny et Doise (1978), les performances effectuées en équipe surpassent les performances que chacun des individus aurait pu offrir seul, à l'instar de Vygotsky, qui affirmait qu'en collaboration « l'enfant peut toujours faire plus et résoudre des problèmes plus difficiles que lorsqu'il agit tout seul » (Vygotsky, 1997).

La programmation, fortement associée au processus de résolution de problème, est définie comme étant l'action d'écrire, à l'aide d'un langage informatique, une série d'actions qui sont interprétées puis exécutées par un ordinateur (Blackwell, 2002). Le programme permet donc de médiatiser les interactions entre l'humain et l'ordinateur. Considérant la nature de l'agent de traitement de l'information (l'ordinateur), il est essentiel que les informations transmises soient dépourvues de toute équivoque ou ambiguïté (Hardouin-Mercier, 1974; Turski, 1978). Par ailleurs, il n'est pas rare d'entendre dans le discours francophone les termes *programmer* et *coder* employés de façon synonymique. Or, nous suggérons de différencier ces deux procédés : le premier, programmer, consiste à réaliser une série de consignes organisées de façon structurée, c'est-à-dire un algorithme, et le deuxième, coder, consiste à traduire en quelque sorte les consignes en langage naturel vers des termes qui pourront être compris par l'ordinateur. Il est donc vrai de dire qu'il faut coder pour programmer, mais l'inverse ne l'est pas. Les programmes, qui peuvent être rédigés à l'aide de différents langages informatiques²⁷, fonctionnent à partir d'opérateurs variés. Par exemple, il est possible d'établir des conditions, des boucles, des variables booléennes (comme vrai et faux), etc. Il est aussi possible d'assembler plusieurs de ces

²⁷ Les langages de programmation peuvent être comparés aux langues comme le français et l'anglais Duchâteau, C. (1989). *Quand le savoir faire ne suffit plus.* : le sens d'un mot ou d'un concept demeure le même, même si sa forme change (par ex. : le mot pomme s'écrit *apple* en anglais, mais les deux mots désignent le même concept).

opérateurs en un tout qui permet d'effectuer une certaine opération : on parle alors de *fonction*. L'agencement de ces opérateurs et codes permet la création d'algorithmes, qui eux permettent la création de programmes entiers dont le niveau de complexité peut varier grandement en fonction du contexte.

Bien que la programmation soit une activité des plus propices à la mobilisation, voire au développement, de nombreuses compétences et processus cognitifs comme la résolution de problèmes (Barr et Stephenson, 2011; Komis et Misirli, 2013; Lai et Yang, 2011) et la collaboration (Mauch, 2001; Nugent et al., 2009; Petre et Price, 2004), il nous semble que sa complexité inhérente puisse rendre son utilisation au primaire parfois difficile à concevoir. Or, avec l'avènement d'une multitude d'applications et de sites internet voués à l'apprentissage ludique de la programmation (i.e. Scratch, Code.org, Swift Playgrounds, etc.), nous pouvons sans doute parler d'une démocratisation de cette activité, à travers les âges.

Les deux principales adaptations permettant de rendre la programmation accessible à des enfants de 6 à 12 ans sont, d'une part, la programmation visuelle, et d'autre part, la programmation tangible. Alors que la première permet de représenter les lignes de codes par des boîtes unies à l'aide de liens (Green et Petre, 1996), l'autre consiste en l'animation de dispositifs robotiques à l'aide de la programmation, ce qui permet aux apprenants de voir concrètement le résultat de leur programme, donc, d'avoir une rétroaction immédiate quant à la validité (ou la non-validité) de leur programme. Cette dernière est d'ailleurs associée au maintien de la motivation et au développement d'une certaine confiance vis-à-vis de la programmation considérant qu'elle peut être, dans sa forme écrite, très complexe et peu motivante pour des élèves du primaire (Pullan, 2013). Enfin, outre le fait de représenter un objet d'apprentissage (apprendre *la* programmation), elle peut aussi être un outil (apprendre *avec* la programmation). Romero, Lepage et Lille (2017) déclinent ces deux volets en cinq types d'usages inscrits dans un continuum allant de la consommation passive à la co-création participative de connaissances, à laquelle les auteurs associent un potentiel socioconstructiviste important (figure 9).

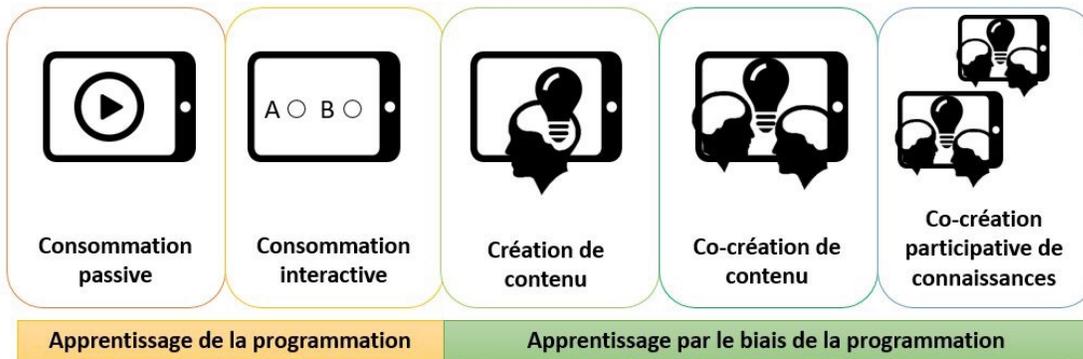


Figure 9. – Types d’usages de la programmation (Romero, Lepage et Lille, 2017).

En développant leur connaissance des concepts de programmation, les élèves sont en mesure de l’utiliser en vue d’apprentissages n’étant pas associés à la programmation, notamment en procédant à la création de contenu, seuls ou en collaboration avec des pairs.

4.2.1 Les pratiques de programmation : aperçu de la littérature

Plusieurs études empiriques ont été réalisées en lien avec l’utilisation de la programmation en contexte pédagogique. Il est notamment question d’impacts sur la compétence en résolution de problèmes (Bugmann et Karsenti, 2018a; Karsenti, Bugmann, et al., 2018; Komis et Misirli, 2011), en collaboration (Moreno-León et Robles, 2015; Pinto-Llorente et al., 2017) ou encore sur la motivation (Sáez-López et al., 2016). Or, avant de poser la question des impacts de la programmation, il semble pertinent et nécessaire de chercher à comprendre comment elle se déroule au primaire. C’est la raison pour laquelle nous nous sommes intéressé aux pratiques effectives d’élèves.

Mis à part la taxonomie des types de tâches de programmation proposée par Bower (2008), qui offre une perspective théorique et macroscopique peu adéquate pour l’enseignement primaire, il a été difficile de trouver des études proposant des classifications ou des catégorisations des pratiques associées à la programmation au primaire. En revanche, les travaux de Ruf, Berges et Hubweiser (2015) offrent des pistes intéressantes propices à une transposition dans le contexte de l’enseignement primaire. En effet, ils ont recensé puis analysé des manuels ou cours d’apprentissage de la programmation pour l’enseignement secondaire (n=4) et universitaire (n=3), sans prétendre à une exhaustivité quelconque. Les tâches identifiées étaient destinées à

des programmeurs novices, ce qui facilite l'adaptation pour des élèves plus jeunes. Parmi les 1098 « tâches » recensées, les auteurs ont été en mesure d'établir 11 types différents²⁸ (tableau 7).

Type de tâche
Écrire le code
Écrire le code en utilisant un code donné
Ajuster, étendre ou compléter le code
Optimiser le code
Déboguer le code
Définir les bonnes conditions pour le code
Tester le code
Transformer le code
Tracer/expliciter le code
Spécifier un problème pour le code donné
Dessiner un diagramme pour le code

Tableau 7. – Types de tâches de programmation (Ruf, Berges et Hubweiser, 2015)

Il s'avère que par sa constitution, et le fait qu'elle résulte de l'analyse d'ouvrages pédagogiques destinés à des programmeurs novices, cette classification de Ruff et ses collègues (2015) est très pertinente à la réalisation de nos travaux avec des élèves au primaire. L'objectif de cette étude est donc de déterminer, en s'appuyant sur les travaux des auteurs précités, les pratiques effectives de programmation visuelle d'élèves du primaire lors d'activités de programmation visuelle et tangible.

4.3 Méthodologie

Dans cette étude de cas multiples (Karsenti et Demers, 2018; R. E. Stake, 1995), nous avons observé les pratiques effectives de programmation visuelle d'élèves du primaire. Pour ce faire, nous avons constitué un échantillon par contraste-approfondissement (Pires, 1997), c'est-à-dire que le fait de juxtaposer les cas nous permet d'aller plus en profondeur dans l'explication du phénomène observé. Cet échantillon intentionnel (Fortin et Gagnon, 2016) est composé de six groupes-classes situés dans trois écoles du Québec, chaque cas étant associé à un groupe-classe,

²⁸ Traduits librement de l'anglais.

qui s'avèrent être des groupes naturels existant à l'extérieur du contexte de la recherche (LeCompte et Preissle, 1993). Aucun critère d'exclusion n'a été appliqué au sein des groupes.

L'école A, située à Montréal, est une école alternative²⁹ accueillant des élèves du primaire et du secondaire. Les groupes n'étant pas formés en fonction de l'âge, les participants du groupe A1 (n=29) sont de différents niveaux scolaires, bien que les élèves du primaire (3^e cycle) soient plus fortement représentés. L'école B, aussi située à Montréal, a un indice du seuil de faible revenu de même qu'un indice de milieu socioéconomique au 10^e rang décile, les niveaux maximums pour ces deux indices de défavorisation du ministère de l'Éducation et de l'Enseignement supérieur du Québec (2020). Les participants de cette école, c'est-à-dire des groupes B1, B2, B3 et B4 (n=70), doivent faire face à différents défis inhérents au contexte socioéconomique. Ils sont aux deuxième et troisième cycles du primaire, c'est-à-dire en 4^e, 5^e et 6^e année. Enfin, l'école C se situe dans la grande région de Québec, en milieu rural³⁰. Les élèves du groupe C1 (n=10) sont en 5^e année et sont dans un groupe-classe réduit puisque plusieurs d'entre eux ont des difficultés d'apprentissage variées.

École	Groupe	Année	Milieu ³¹	Filles (n)	Garçons (n)	Élèves (n total)
École A	A1	Multi	Urbain	9	20	29
École B	B1	4 ^e année	Urbain	9	5	14
	B2	5 ^e année	Urbain	8	9	17
	B3	5 ^e année	Urbain	10	10	20
	B4	6 ^e année	Urbain	6	13	19
École C	C1	5 ^e année	Rural	5	5	10
				47	62	109

Tableau 8. – Participants à l'étude

Il est intéressant ici de souligner le caractère diversifié des différents groupes composant l'échantillon, tant au niveau du statut socioéconomique, du milieu – rural ou urbain – et du type

²⁹ Selon le Réseau des écoles publiques alternatives du Québec (RÉPAQ), « l'école alternative est un milieu éducatif dynamique, prônant une approche participative, communautaire et humaniste dans laquelle chaque intervenant (équipe de direction, enseignants, parents) joue un rôle actif dans l'épanouissement de l'élève » (RÉPAQ, 2020).

³⁰ Selon les données de l'Institut national de santé publique du Québec (<https://inspq.qc.ca/santescope/milieux-ruraux-urbains>)

³¹ *Ibid.*

d'école, qu'elle soit alternative ou *traditionnelle*, privée ou publique. Cela répond au critère de diversification externe des échantillons pas cas multiples (Pires, 1997).

4.3.1 Collecte de données

Nous avons implémenté un scénario pédagogique intitulé *Deviens un maître NAO* (Karsenti et al., 2019a), qui présente une série de tâches (réparties en 20 niveaux) visant à animer un dispositif robotique à l'aide de la programmation. Numérotés de 1 à 20, ces niveaux induisent une progression graduelle permettant à l'élève, d'une part, de se familiariser avec le robot et son fonctionnement, et d'autre part, de réaliser des opérations de programmation de plus en plus complexes et variées.

Ce scénario pédagogique a été utilisé dans divers contextes (Bugmann et Karsenti, 2018a, 2018b; Karsenti, Bugmann, et al., 2018). Sa version actuelle est particulièrement intéressante pour notre étude dans la mesure où elle permet aux élèves du primaire d'effectuer les principales opérations de programmation accessibles à leur niveau (voir la section résultats). Ce scénario favorise également la mobilisation des habiletés ou connaissances de disciplines comme les mathématiques, le français ou l'histoire, à travers des mises en contexte ludiques et captivantes de la programmation d'un robot humanoïde. La programmation du robot NAO s'effectue avec le logiciel *Chroregraphe* (Aldebaran Robotics, 2014), dont l'interface est en anglais. Ce logiciel repose sur le principe de programmation visuelle, c'est-à-dire que la principale façon d'animer le robot est de glisser des boîtes d'actions dans l'espace de travail, puis de les associer à l'aide de liens pour créer une séquence, un algorithme (figure 10).

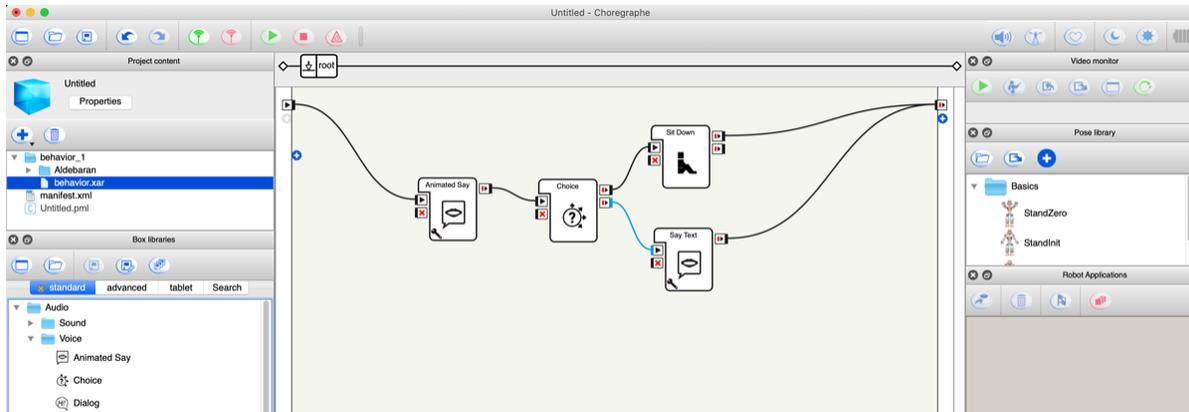


Figure 10. – Interface de programmation visuelle du logiciel Choregraphe

Bien qu'il soit possible d'accéder aux lignes de code textuel, ce scénario pédagogique insiste sur l'aspect visuel, exception faite du niveau 17 qui invite l'apprenant à explorer les coulisses de ces boîtes (le *back-end*, en anglais). La bibliothèque de boîtes (*box library*) du logiciel propose différentes fonctions qui permettent d'activer les fonctions du robot, allant des capteurs tactiles à la synthèse vocale, en passant par la reconnaissance visuelle et le mouvement de ses bras, de ses jambes ainsi que de sa tête. Les boîtes permettent aussi de reconnaître visuellement le type d'information qui circule entre elles, c'est-à-dire le type « bang » (un signal active la boîte), « string » (valeurs alphabétiques) ou « numbers » (valeurs numériques).

4.3.2 Modalités

Les participants ont été placés en équipes pour réaliser les tâches du scénario pédagogique. Cette décision est motivée par le potentiel socioconstructiviste d'activités de cocréation à l'aide de la programmation (Romero et al., 2017) et, par extension, par la possibilité que les élèves puissent réinvestir seuls les apprentissages qu'ils auront réalisés en équipe (Vygotsky, 1934). Le nombre d'élèves dans chaque équipe variait selon le contexte du groupe (de 2 à 5 élèves par équipe). Chaque groupe se voyait remettre une valise contenant un robot NAO, un ordinateur portable doté du logiciel *Choregraphe*, plusieurs copies imprimées des tâches *Deviens un maître NAO* ainsi qu'un guide d'accompagnement en version électronique (Karsenti et al., 2019b), sur une tablette. Afin d'observer les opérations de programmation accomplies par les équipes, nous avons utilisé deux instruments, le premier étant l'enregistrement des écrans des ordinateurs utilisés par les

élèves, et le second étant l'observation vidéographiée (Karsenti, Komis, et al., 2018). Des caméras ont été installées devant chaque équipe afin d'offrir une vue générale des actions des élèves à l'extérieur de l'interface *Choregraphe*. Comme il s'agit de programmation tangible impliquant l'animation d'un dispositif robotique, ce second instrument s'avère particulièrement utile pour contextualiser les données des enregistrements d'écran.

4.3.3 Traitement et analyse des données

L'analyse systématique des données a été effectuée sur les enregistrements d'écran afin de pouvoir observer les pratiques effectives de programmation dans l'interface du logiciel *Choregraphe*. Parmi le corpus de données total, nous avons analysé les séances de trois équipes, et ce, pour trois visites par équipe, soit pour un total de 9 séances. Cette sélection d'une portion du corpus s'explique par le fait que toutes les équipes réalisent les mêmes tâches, présentées dans le scénario pédagogique commun. Nous avons donc atteint une saturation empirique (Pires, 1997). L'analyse des données a été réalisée avec le logiciel NVivo 12 (QSR International, 2020), où les catégories – et les codes associés – (Paillé et Mucchielli, 2005) correspondent aux opérations de programmation selon la classification proposée par Ruf, Berges et Hubweiser (2015) et aux niveaux du scénario pédagogique. En amont de ce codage mixte (van der Maren, 2004), la grille a été adaptée pour refléter la réalité des activités de programmation à l'enseignement primaire. Puis, de façon itérative, des modifications ont été apportées à la grille afin qu'elle tienne compte d'aspects émergents (d'où le caractère mixte du codage).

4.4 Résultats

Nous présentons les résultats en suivant une progression logique, qui débute avec la présentation de notre adaptation de la classification de Ruf, Berges et Hubweiser (2015), qui se poursuit avec l'explicitation des pratiques de programmation observées, puis se termine par la mobilisation des pratiques au sein du scénario pédagogique.

4.4.1 Adaptation de la classification

En procédant, en amont, à l'analyse des notes de terrain, puis à l'analyse du corpus de données, il est devenu évident que la classification de Ruf et ses collègues devait faire l'objet d'une

adaptation pour refléter les réalités du contexte pour lequel nous souhaitons proposer une typologie des pratiques de programmation.

Pratiques de programmation
Assembler (opérations fondamentales)
- Chercher
- Sélectionner
- Lier
- Paramétrer
Assembler en utilisant un code donné
Ajuster, étendre ou compléter
Optimiser
Déboguer
Tester
Transformer
Tracer/expliciter
Spécifier un problème
Dessiner un diagramme

Tableau 9. – Classification adaptée de Ruf, Berges et Hubweiser (2015)

Le principal changement est lié au terme « écrire », qui n'est pas cohérent avec la programmation visuelle. Nous avons donc opté pour le terme « assembler », que nous avons subdivisé en opérations secondaires : chercher les boîtes, sélectionner les boîtes, lier les boîtes entre elles et enfin paramétrer ces boîtes. Certaines des pratiques de la classification de Ruf et ses collègues n'ont pu être observées : nous aborderons cela dans la discussion.

4.4.2 Analyse des pratiques de programmation visuelle observées

Nous présentons ici chacune des pratiques de programmation observées en offrant des exemples concrets tirés du corpus de données.

4.4.2.1 Assembler

L'assemblage représente une agglomération des pratiques que nous qualifions de fondamentales pour la programmation visuelle. Leur caractère fondamental se manifeste par leur nécessité et leur utilisation, ce qui fut le cas dans notre corpus de données où trois pratiques observées sur quatre (76%) relevaient de l'assemblage. Ainsi, la pratique d'assemblage devient en quelque sorte

une métapratique incluant la recherche, la sélection, la liaison et le paramétrage. Le fait de décliner l'assemblage en différentes pratiques permet d'obtenir un portrait beaucoup plus précis de l'activité de programmation.

4.4.2.1.1 Chercher

Contrairement à la programmation traditionnelle (écrite), la programmation visuelle nécessite une recherche afin d'identifier la boîte, c'est-à-dire le code ou la fonction, qui sera utilisée dans le programme. Le logiciel *Choregraphe* offre une bibliothèque de boîtes parmi lesquelles il est possible de chercher soit en naviguant dans l'arborescence de plusieurs dossiers thématiques, soit en utilisant un moteur de recherche. Ainsi, la pratique de programmation *chercher le code* correspond au moment pendant lequel l'élève consulte la bibliothèque.

4.4.2.1.2 Sélectionner

Une fois la boîte identifiée dans la bibliothèque, l'utilisateur doit sélectionner la boîte avec le curseur et la déplacer dans l'espace de travail. Il est également possible de créer des boîtes à partir de formats préétablis, ne laissant à l'élève que la tâche de la paramétrer. Par ailleurs, certaines fonctionnalités du robot exigent l'activation d'un bouton. C'est notamment le cas du mode *Animation*, qui est utilisé afin d'animer les différents membres du robot NAO à l'aide d'une technique reposant sur le principe d'animation en volume (*stop-motion*). La pratique *sélectionner le code* représente donc les instances où l'élève a déplacé une boîte provenant de la bibliothèque dans l'espace de travail, a créé une nouvelle boîte dans l'espace de travail, ou encore a activé différentes fonctionnalités, comme le mode *Animation*.

4.4.2.1.3 Lier

Cette pratique est propre à la programmation visuelle. L'utilisation de boîtes de code distinctes induit *ipso facto* la nécessité de lier ces boîtes d'une façon ou d'une autre. Dans notre cas, le logiciel permet de lier les boîtes entre elles à l'aide d'un fil noir, un système comparable à un réseau électrique, où l'impulsion du départ, c'est-à-dire au lancement du programme, parcourt un fil avant de traverser (activer) chacune des boîtes. Cet influx est d'ailleurs rendu visible par un point vert parcourant les fils dans l'interface de programmation du logiciel, ce qui permet à l'utilisateur de suivre la progression du programme en temps réel. La pratique de liaison renvoie

donc à l'action d'associer les boîtes entre elles, de même qu'avec les points de départ et de fin du programme. Il est à noter ici que certains élèves éprouvaient parfois des difficultés à tracer ces liens, notamment associées à la motricité fine.

4.4.2.1.4 Paramétrer

Le paramétrage fait référence à toute action impliquant d'entrer dans une boîte pour ajouter, modifier, ou supprimer du contenu. Par exemple, lorsqu'il était question de faire parler NAO, les élèves utilisaient la boîte *Say* ou *Animated Say*, dont le paramétrage consiste notamment à écrire le texte que le robot devra réciter. Il est également possible de paramétrer la vitesse et la tonalité de la voix du dispositif. Un autre exemple de paramétrage est la création de boîtes *Timeline*, qui sont utilisées avec le mode *Animation*. Comme il s'agit d'animation de volume (*stop motion*), il est nécessaire de bouger le robot dans une position, d'enregistrer cette position, puis de le mettre dans une autre position, l'enregistrer, et ainsi de suite. L'enregistrement des positions sur une ligne du temps représente aussi une action de paramétrage. La pratique de paramétrage consiste donc à insérer des valeurs numériques ou alphabétiques, à ajouter ou déplacer des indicateurs sur des échelles, ou à modifier des valeurs à l'aide de boutons.

4.4.2.2 Assembler en utilisant un code donné

L'assemblage à partir d'un code donné a été distingué des opérations fondamentales d'assemblage en raison de son caractère particulier. La particularité réside dans le fait que cette pratique ne peut être réalisée que dans un contexte où une certaine forme d'encadrement est offerte, et non en pratique autonome. En effet, cette pratique implique que l'utilisateur se soit vu offrir une sélection de boîtes et que ce dernier choisisse l'une d'elles pour composer son programme. Par exemple, dans le scénario pédagogique que nous avons proposé, au niveau 11, les élèves sont appelés à réaliser une tâche pour laquelle quatre codes sont suggérés : « voici certaines des boîtes à utiliser » (Karsenti et al., 2019a). Ces boîtes sont proposées à l'extérieur de l'espace de travail, ce qui implique que l'élève doit mobiliser les pratiques fondamentales d'assemblage afin d'utiliser ces boîtes suggérées (codes donnés).

4.4.2.3 Ajuster, étendre ou compléter

Similaire à l'assemblage à partir d'un code donné, cette pratique consiste à utiliser un code déjà présent dans l'espace de travail. La distinction est donc la présence des boîtes dans l'espace de travail, ce qui n'était pas le cas pour la pratique précédente. Dans notre corpus de données, nous avons été en mesure d'observer cette pratique au sein d'un même niveau, puisque chaque niveau de 1 à 9 est séparé en trois tâches. Ainsi, considérant la progression d'une tâche à l'autre, il est possible pour l'élève d'ajouter des boîtes ou de paramétrer à nouveau des boîtes utilisées antérieurement. Ces pratiques impliquent donc la modification d'un code déjà présent : il peut avoir été généré automatiquement dans l'interface, ou alors avoir été assemblé par l'élève précédemment.

4.4.2.4 Tester

Cette pratique concerne la vérification des programmes réalisés en les lançant, puis en observant le résultat de l'implémentation. Ce résultat peut prendre plusieurs formes selon le type de programmation. Dans le cas de la programmation tangible, la vérification peut s'effectuer soit en analysant le déroulement du programme dans l'interface du logiciel, soit en observant le comportement du robot. Cette pratique permet donc de prouver le bon fonctionnement du programme conçu. Les données de notre corpus démontrent que les élèves ont parfois lancé le programme simplement pour voir le robot s'animer à nouveau, ayant préalablement confirmé, lors du premier test, que le programme fonctionnait adéquatement.

4.4.2.5 Déboguer

Subséquente à la pratique de vérification (*tester*), la pratique de débogage survient lorsqu'il y a discordance entre le résultat obtenu et le résultat attendu. Le débogage s'impose alors comme un processus dont le but est d'identifier l'erreur (le bogue) et se conclut par une vérification réussie. Cette pratique est en fait une transposition du processus de résolution de problème, mobilisant ainsi plusieurs autres pratiques en vue de résoudre la situation. Le débogage est donc une pratique d'ordre général que l'on pourrait aussi qualifier de métapratique, comme l'assemblage. Or, ici les pratiques sous-jacentes ne sont pas propres au débogage en soi, mais bien aux pratiques d'assemblage fondamentales et de vérification notamment. En procédant par

essai-erreur, les élèves identifient le problème, tentent de le régler en modifiant le programme, effectuent un test, et ce, de façon itérative jusqu'à ce que le problème (bogue) soit résolu.

4.4.3 Mobilisation des pratiques de programmation visuelle

L'analyse des données a permis de démontrer dans quelles proportions les opérations de programmation ont été utilisées par les élèves (fréquence). La figure 11 présente donc les proportions d'encodage³² des pratiques de programmation à chaque niveau du scénario pédagogique. Le niveau 1 n'a permis l'observation d'aucune pratique de programmation puisque ce niveau amène l'élève à interagir avec le robot à l'aide de ses capteurs tactiles et de la synthèse vocale : ces fonctions font partie de la vie autonome du robot, c'est-à-dire qu'elles sont actives sans programmation. Il est également à noter que le niveau maximal atteint par les participants est le niveau 13, ce qui explique l'absence d'encodage pour les niveaux 14 à 20.

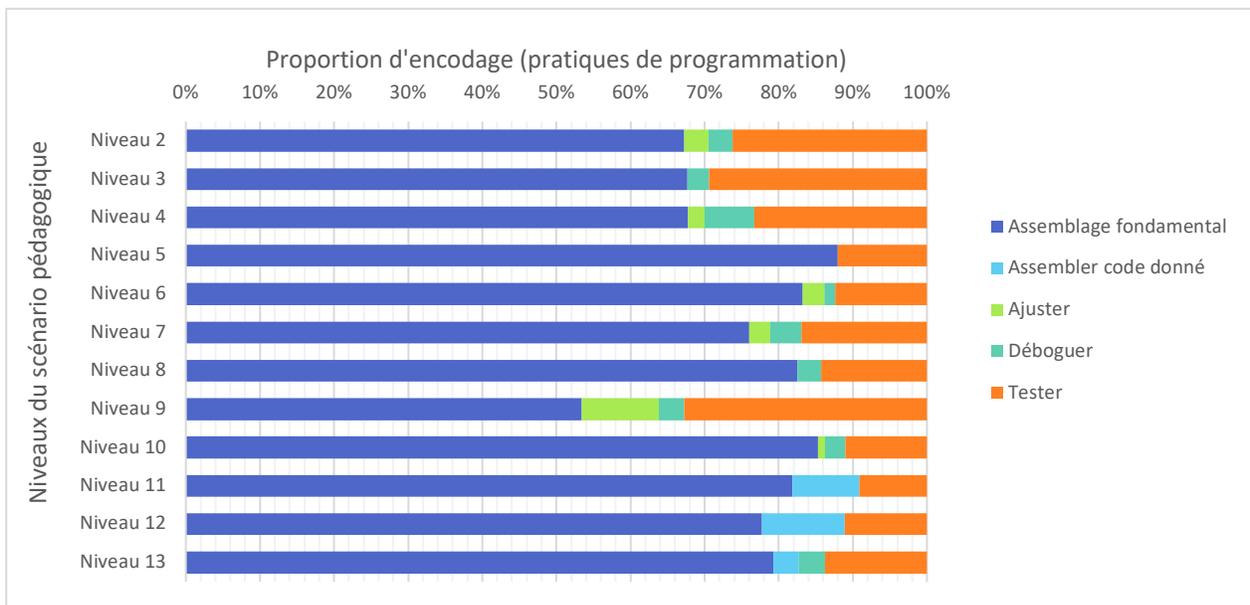


Figure 11. – Proportions de fréquence d'encodage des pratiques de programmation par niveau du scénario pédagogique

³² Le verbe *encoder* réfère à l'action d'attribuer un code à un segment du corpus de données vidéos van der Maren, J.-M. (2004). *Méthodes de recherche pour l'éducation* (2^e éd.). Presses de l'Université de Montréal. . Nous utilisons le terme *encodage*, dans son emploi nominal, pour désigner l'ensemble de codes attribués.

On constate que l'**assemblage fondamental** représente une proportion importante des pratiques de programmation effectuées par les élèves lors des activités. En effet, bien qu'étant moins utilisé au niveau 9, il représente néanmoins une majorité des pratiques à chaque niveau ($M=75,8\%$). La figure 12 illustre la proportion attribuée à chaque pratique d'assemblage fondamentale (fréquence), par niveau.

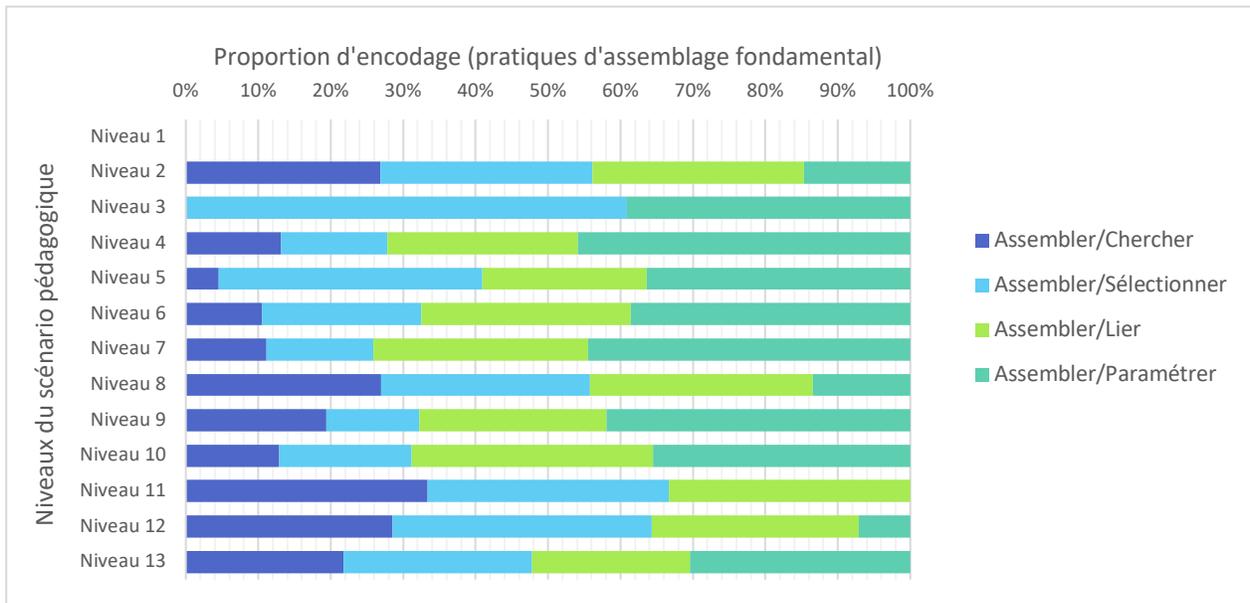


Figure 12. – Proportions de fréquence d'encodage des pratiques d'assemblage fondamentales (4) par niveau

Outre les proportions associées à la fréquence des opérations de programmation, nous avons cherché à savoir si le temps accordé à chaque opération était similaire. Le tableau 10 indique : (a) le nombre de minutes passées par les élèves à effectuer chacune des pratiques de programmation observées ; puis (b) compare les proportions de durée et de fréquence.

Pratiques de programmation	Durée d'encodage (minutes)	Proportion (durée)	Fréquence	Proportion (fréquence)
Assembler (opérations fondamentales)				
Chercher	33	6,5%	82	11,6%
Sélectionner	13	2,6%	126	17,8%
Lier	28	5,5%	149	21,1%
Paramétrer	215	42,5%	180	25,5%
Assembler en utilisant un code donné	5	1,0%	4	0,6%
Ajuster, étendre ou compléter	25	4,9%	17	2,4%
Déboguer	119	23,5%	22	3,1%
Tester	68	13,4%	126	17,8%
	506	100%	706	100%

Tableau 10. – Comparaison des proportions de durée et de fréquence d'encodage

Les données obtenues indiquent une disparité entre la fréquence de certaines pratiques et leur durée. Cette disparité souligne une proportion considérablement plus importante tantôt pour la durée, tantôt pour la fréquence. Parmi les pratiques dont la fréquence surpasse la durée, on retrouve notamment la sélection, dont la proportion de fréquence est 6,8 fois plus élevée, et la liaison, dont la proportion de fréquence est 3,8 fois plus élevée. Inversement, la proportion de temps accordé au débogage est 7,6 fois plus élevée que la proportion de fréquence de cette pratique, ce qui signifie qu'elle est moins fréquente, mais qu'elle dure longtemps.

4.5 Discussion

Les résultats ont dépeint un portrait détaillé des pratiques de programmation effectives des élèves. Qu'en est-il alors des interprétations que nous pouvons en faire ? Cette discussion des résultats présente notre proposition de typologie des pratiques de programmation.

4.5.1 Les pratiques les plus utilisées

Les données font état d'un phénomène intéressant : alors que certaines pratiques sont très fréquentes et de courte durée, d'autres sont peu fréquentes et de très longue durée. Lorsque l'on s'intéresse à la fréquence, on remarque que les pratiques d'assemblage fondamentales dominent à ce chapitre, suivies de la pratique de vérification (tester). Lorsqu'il est question de la durée des

pratiques, c'est le paramétrage et le débogage qui se démarquent. Comme le paramétrage représente une partie cruciale du processus de programmation, il semble logique de constater qu'il occupe 42,5% du temps observé pour l'ensemble des pratiques. Cette pratique recèle une importante partie de la complexité inhérente à l'activité de programmation puisqu'elle suscite la manipulation de variables et de données qui déterminent la logique interne de chaque code. Le paramétrage est d'ailleurs une pratique souvent utilisée dans le processus de débogage. Cela dit, pour le débogage, c'est le rapport entre la durée et la fréquence qui est digne de mention : la proportion de durée est 7,6 fois plus élevée que la proportion de fréquence. Ainsi, le processus de débogage peut s'avérer chronophage.

Les figures 11 et 12 démontrent que la mobilisation des pratiques de programmation dans chaque niveau du scénario pédagogique *Deviens un maître NAO* est similaire. Bien que des différences subsistent, expliquées notamment par la nature plurielle des défis proposés aux élèves, il demeure que les apprenants ont été en mesure de mettre en œuvre les pratiques de façon régulière au fil de leur progression dans les niveaux du scénario.

4.5.2 Typologie des pratiques de programmation

Cette étude empirique a mené, dans un processus itératif, à la déclinaison d'un ensemble de pratiques effectives de programmation d'élèves du primaire. Or, il aurait été peu judicieux de proposer une typologie strictement basée sur les observations réalisées dans le cadre de cette recherche, puisque le scénario pédagogique utilisé possède certaines limites inhérentes. En effet, tels qu'ils ont été conçus, ces niveaux ne sollicitent pas certaines pratiques qui sont pourtant pertinentes dans le contexte de l'enseignement primaire. Ainsi, cette proposition de typologie empirique tient compte d'autres pratiques qui ont été identifiées par Ruf, Berges et Hubweiser (2015) et qui ont été repérées à de nombreuses reprises dans la littérature. Le tableau 11 présente la typologie à l'aide de descripteurs et d'exemples : ces derniers ne sont présentés qu'à titre indicatif et n'ont pas pour but de restreindre ou limiter l'interprétation de ces pratiques, qui s'inscrivent dans une multitude de contextes variés.

Pratiques de programmation	Descripteur
1. Assembler (fondamental) : 1.1. Chercher	L'élève cherche dans l'interface de programmation, à l'aide d'une bibliothèque ou d'un moteur de recherche, un code donné. Dans le cas des bibliothèques, il peut s'agir d'une liste du nom des codes, ou encore des boîtes elles-mêmes.
1.2. Sélectionner	À partir de la bibliothèque ou de tout autre emplacement dans l'interface, l'élève sélectionne ou génère une boîte, puis la déplace dans l'espace de travail, le cas échéant. [Robot] Lorsque le programme anime un dispositif robotique, la sélection peut aussi représenter l'activation de fonctionnalités du robot via l'interface de programmation.
1.3. Lier	L'élève associe les codes (boîtes) entre eux et, s'il y a lieu, aux points de départ et de fin du programme. Cette pratique pourrait être sous-utilisée, voire inutilisée, dans le cas d'interfaces où le lien entre les boîtes est effectué par la juxtaposition de ces dernières (à la manière d'aimants). La pratique <i>sélectionner</i> possède alors une double finalité qui inclut la liaison.
1.4. Paramétrer	L'élève ajoute des valeurs ou modifie et retire des valeurs préexistantes (par défaut). Ce paramétrage peut être effectué tant en surface, lorsque le paramétrage apparaît sur la boîte, qu'en entrant dans une boîte. [Robot] Lorsque le programme anime un dispositif robotique, cette pratique peut impliquer des actions réalisées auprès du robot, par exemple pour créer une danse (création d'une séquence d'animation en volume).
2. Assembler en utilisant un code donné	L'élève conçoit un programme incluant un ou plusieurs codes suggérés par un agent réel ou virtuel.
3. Ajuster, étendre ou compléter	L'élève utilise un ou plusieurs codes liés, déjà présents dans l'espace de travail, afin de créer un programme. Contrairement à l'assemblage à partir d'un code donné, où le code était suggéré, la pratique d'ajustement s'effectue sur des éléments déjà présents dans l'espace de travail (qu'ils y aient été placés par un agent réel, virtuel, ou par l'élève lui-même, pour une autre tâche antérieure).

(Ce tableau continue à la page suivante.)

4. Tester	<p>L'élève lance le programme et en vérifie le résultat. Cette pratique peut amener à améliorer (optimiser), déboguer le programme. La vérification peut aussi servir d'étape intermédiaire dans le processus de programmation, c'est-à-dire pour valider un code ou une portion de programme.</p> <p>[Robot] Lorsque le programme anime un dispositif robotique, la vérification consiste à observer le robot et à confirmer que toutes les actions (mouvements, voix, etc.) ont été effectuées.</p>
5. Déboguer	Après avoir constaté que le programme n'a pas donné le résultat escompté, l'élève part à la recherche du problème (bug) dans l'implémentation du programme. Il s'agit d'un processus itératif d'essai-erreur dans lequel sont utilisées de nombreuses autres pratiques ; surtout la vérification (tester) et les pratiques d'assemblage fondamentales.
6. Transformer*	L'élève remplace un code par un autre code dont les fonctionnalités sont similaires et induisent un résultat semblable, voire identique. Il s'agit ici de comprendre que plusieurs codes peuvent mener à des résultats similaires.
7. Optimiser*	L'élève adapte son programme ou son code de façon à retirer les boîtes inutiles ou les structures trop complexes. Contrairement à la transformation, cette pratique s'effectue lorsque l'élève manifeste explicitement son intention de rendre le code ou le programme plus efficace ou mieux structuré. Par exemple, plutôt que d'utiliser six boîtes pour une répétition d'actions, il pourrait utiliser la boîte une seule fois et y adjoindre un code dont la fonction est de répéter à n reprises.
8. Spécifier un problème*	Se voyant offrir un programme donné, l'élève doit proposer un problème qui est cohérent avec la structure et les codes présents dans le programme. Cette pratique est induite par un intervenant, que ce soit l'enseignant ou un agent virtuel. Cette pratique ne nécessite pas l'utilisation d'un appareil numérique.

* Pratiques n'ayant pas été observées dans le cadre de cette étude.

(Ce tableau continue à la page suivante.)

9. Tracer ou expliquer*	<p>L'élève démontre sa compréhension du programme en expliquant, visuellement (avec son doigt) ou verbalement, son déroulement. Cette pratique peut être initiée par l'élève ou sollicitée par un intervenant. Bien que pouvant être effectuée en lançant le programme sur l'ordinateur, cette pratique ne nécessite pas l'utilisation d'un appareil numérique.</p> <p>[Robot] Lorsque le programme anime un dispositif robotique, le principal intérêt doit être accordé au programme en soi, et le comportement du robot peut être utilisé pour illustrer l'explication.</p>
10. Concevoir un ordinogramme*	<p>L'élève conçoit un diagramme représentant la structure d'un programme donné et toutes ses composantes. Cette pratique, qui ne nécessite pas l'utilisation d'un appareil numérique, permet de développer une vision d'ensemble du programme et de réfléchir au rôle de chaque composante (codes, paramètres, liaisons, etc.).</p>

* Pratiques n'ayant pas été observées dans le cadre de cette étude.

Tableau 11. – Typologie des pratiques effectives de programmation

4.6 Discussion

D'emblée, il est important de mentionner que le scénario pédagogique a une forte incidence sur les pratiques de programmation effectives des élèves. En effet, si les tâches ou les mises en contexte proposées aux élèves ne favorisent pas la mise en œuvre de certaines pratiques, il est fort probable que ces dernières ne soient pas utilisées. Comme le scénario pédagogique *Deviens un maître NAO* est plutôt fermé, c'est-à-dire qu'il propose des tâches spécifiques, il est normal que certaines pratiques de programmation non sollicitées par le scénario n'aient pas été mises en œuvre par les élèves. Nous présentons ici les pratiques n'ayant pas été observées dans le cadre de cette étude, mais qui ont tout de même été incluses dans la typologie que nous proposons.

Les pratiques de transformation (*transformer*) et d'optimisation (*optimiser*) s'inscrivent dans une suite logique de pratiques plus complexes, mais tout de même accessibles à des élèves du primaire. La transformation implique de changer le corps d'un code tout en conservant sa fonctionnalité élémentaire. Ce changement de corps peut être au niveau du langage (passer d'un langage informatique à un autre) ou encore au niveau des fonctions (remplacer une fonction par une autre). La finalité de ce changement est de comprendre que plusieurs codes différents

peuvent mener à des résultats similaires. Quant à l'optimisation, elle fait référence à l'action de rendre le code ou le programme plus efficace, notamment en évitant les combinaisons d'éléments pouvant être remplacées par moins d'éléments faisant le même travail. L'optimisation est d'ailleurs un enjeu majeur dans le travail des programmeurs experts, qui se doivent d'effectuer des programmes hautement complexes tout en nécessitant le moins de ressources possible (Hoos, 2012; Sedgewick et Wayne, 2011).

Les pratiques restantes, c'est-à-dire la spécification d'un problème, le traçage ou l'explication, ainsi que la conception d'un ordinogramme relèvent davantage d'un processus réflexif, voire métacognitif. En effet, ces pratiques permettent de poser un regard différent sur un programme donné. Avec la spécification d'un problème, le processus de résolution de problème est inversé : plutôt que de se voir offrir un problème à résoudre à l'aide de la programmation, l'élève se voit offrir un programme pour lequel il doit trouver un problème dont les composantes et la structure sont en adéquation avec l'algorithme du programme en question. Le traçage ou l'explication sont utilisés pour amener l'apprenant à verbaliser le déroulement d'un programme, inspiré de la méthode *think aloud*, qui « consiste à demander à des individus de penser à voix haute pendant le processus de résolution d'un problème, puis d'analyser les protocoles verbaux qui en résultent »³³ (van Someren et al., 1994, p. xi). Alors que le traçage peut se faire avec le doigt à l'écran, au fur et à mesure que le programme progresse, l'explication se fait verbalement et consiste à décrire les actions qui surviennent, de même que la logique derrière celles-ci (s'il y a lieu). Enfin, la dernière pratique (concevoir un ordinogramme) a pour but de représenter visuellement la structure globale d'un programme donné. Ceci offre une opportunité tant d'avoir une vue d'ensemble du programme que de réfléchir à la coexistence de plusieurs codes qui se complètent ou qui, parfois, génèrent des contradictions ou des bogues. Il est à noter que les deux dernières pratiques évoquées, c'est-à-dire le traçage ou l'explication et la conception d'un ordinogramme, sont fortement liées à la programmation *offline* (hors ligne ou débranchées). Ce type de programmation insiste sur la possibilité de réaliser un programme sans support numérique. Il peut donc s'agir d'activités sur une feuille de papier, avec des blocs de bois, ou

³³ Traduction libre (« *The think aloud method consists of asking people to think aloud while solving a problem and analysing the resulting verbal protocols* »)(van Someren, Barnard et Sandberg, 1994, p. xi).

encore au tableau à l'avant de la classe (Computer Science Education Research Group, 2018; Conde et al., 2017; Horn et al., 2009; Romero et al., 2018). Ces activités mettent l'accent sur les processus cognitifs qui sous-tendent l'action de programmer en soi (le plus complexe) et non l'action, plus technique, d'implémenter le code à l'aide d'un ordinateur (plus technique) (Lopez, 1986).

4.6.1 Pertinence de la typologie proposée

Cette proposition de typologie est pertinente dans plusieurs contextes. D'abord, considérant la démocratisation de l'usage pédagogique de la programmation informatique, il semble nécessaire d'offrir plusieurs balises afin d'encadrer cette activité à l'école. Le risque de limiter le potentiel de la programmation à son aspect ludique et motivant est bien réel. Ainsi, il est souhaitable que soient développés des scénarios pédagogiques ou des activités, comme celles de Romero et Vallerand (2016), mobilisant la programmation de façon réfléchie et intégrale, sans toutefois prétendre à l'exhaustivité de toutes ses facettes. Cette typologie offre donc à la fois des repères théoriques et empiriques pour la conception ou l'adaptation d'activités visant à mobiliser et à développer les compétences d'élèves du primaire par la programmation. De plus, cette typologie pourrait également permettre donner un cadre de référence à des manuels scolaires ou pédagogiques à l'attention des enseignants, tant du primaire que du secondaire. En effet, même si nos données ont été recueillies auprès d'élèves du primaire, il semble que cette typologie gagnerait aussi à être utilisée auprès d'élèves et d'enseignants au secondaire. En outre, comme la programmation fait désormais partie des politiques ministérielles (Ministère de l'Éducation, 2020a; Ministère de l'Éducation et de l'Enseignement supérieur, 2019), une telle typologie ne pourra que faciliter l'enseignement et l'apprentissage de la programmation informatique.

4.7 Limites

L'une des principales limites de notre recherche est le fait de n'avoir pu observer empiriquement cinq des pratiques incluses dans la typologie. Bien que notre choix d'inclure ces pratiques soit appuyé sur la littérature et sur de nombreuses observations informelles effectuées sur le terrain, il demeure néanmoins qu'elles n'ont pas été observées dans le cadre de cette étude, qui cherche à identifier des pratiques effectives de programmation. Il y aurait certainement avantage à

reproduire cette étude dans un contexte authentique de programmation pédagogique où il s'agirait d'un scénario pédagogique distinct et plus ouvert.

Une autre limite a trait au corpus de données de cette étude de cas multiples, collecté par la mise en place du scénario pédagogique *Deviens un maître NAO*. Il n'a pas été possible d'observer l'entièreté du scénario, c'est-à-dire jusqu'au vingtième niveau. Par ailleurs, nous avons également un échantillon surtout composé d'élèves du deuxième et du troisième cycle du primaire (3^e à 6^e année). Plus complexe au premier cycle, la programmation est néanmoins une activité réalisable avec des élèves de cet âge. Il aurait donc été intéressant – et c'est une piste de recherche future que nous envisageons – de mettre cette typologie à l'épreuve au premier cycle du primaire.

4.8 Conclusion

L'objectif de cette étude de cas multiples était d'offrir un portrait détaillé des pratiques effectives de programmation d'élèves à l'enseignement primaire et, surtout, de proposer une typologie compréhensive et adaptée aux réalités de ces élèves. Après avoir décliné l'ensemble des pratiques observées auprès des apprenants de notre échantillon, il est devenu évident que les typologies et taxonomies tirées de la littérature n'étaient pas applicables au primaire. Cela a d'ailleurs confirmé la pertinence de notre proposition de typologie. Ainsi, en nous appuyant tant sur les données empiriques que nous avons collectées que sur la littérature, nous avons développé une typologie complète et détaillée qui a le potentiel de devenir un cadre de référence pour quiconque souhaite concevoir une activité mobilisant la programmation au primaire, ou encore créer des manuels ou guides pédagogiques pour encadrer l'apprentissage de la programmation. Cela dit, cette typologie gagnerait à être mise à l'épreuve dans différents contextes de programmation pédagogique afin de préciser les pratiques non observées dans le cadre de cette étude. Notre travail ouvre sur de nombreuses pistes d'approfondissements potentielles que nous souhaiterions explorer lors de recherches futures, notamment la validation de cette typologie auprès d'élèves du premier cycle du primaire, voire du préscolaire, ainsi que l'application de la typologie dans le cadre d'un autre scénario pédagogique.

Références

Les références de cet article sont présentées à la section *Références bibliographiques* de cette thèse.

Chapitre 5 - Conception d'une grille d'observation de la résolution collaborative de problèmes (RCP)

ARTICLE 2

RÉSUMÉ

Cet article présente les étapes ayant mené à la conception d'une grille d'observation de la résolution collaborative de problèmes (RCP). En nous appuyant sur le modèle de Strayer et Gauthier (1982), nous proposons une présentation du processus structurée en trois volets, soit la phase descriptive, la phase exploratoire et l'évaluation systématique. Composée de trois dimensions au travers desquelles sont répartis 15 indicateurs, descripteurs et exemples, cette grille offre une perspective intéressante pour l'observation qualitative et interprétative de la mobilisation des compétences de RCP : il ne s'agit donc pas uniquement de calculer des fréquences, mais aussi d'analyser les comportements associés à chacun d'eux afin de formuler des hypothèses (aspect scientifique) ou permettre aux enseignants d'orienter leurs pratiques pédagogiques en fonction des besoins des élèves (aspect pratique). Le contexte entourant la conception de cette grille est l'usage pédagogique de la programmation en contexte scolaire, mais cela ne doit pas limiter les champs et contextes d'application potentiels de cette grille qui se veut adaptable à une variété de contextes et d'ordres d'enseignement.

Mots-clés : grille d'observation, résolution de problème, collaboration, RCP, programmation.

ABSTRACT

This article presents the steps that led to the design of an observation grid of collaborative problem solving (CPS). Based on Strayer and Gauthier's model (1982), we propose a presentation of the process structured in three parts: the descriptive phase, the exploratory phase and the systematic evaluation. Composed of three dimensions through which 15 indicators, descriptors and examples are distributed, this grid offers an interesting perspective on the qualitative and interpretative observation of the CPS skills: it is therefore not only a question of calculating frequencies, but also of analyzing the behaviours associated with each of them in order to formulate hypotheses (scientific aspect) or allow teachers to orient their teaching practices according to the needs of students (practical aspect). The context surrounding the design of this grid is the pedagogical use of programming in a school context, but this should not limit the potential fields and contexts of application of this grid, which is intended to be adaptable to a variety of contexts and teaching levels.

Keywords: observation grid, problem solving, collaboration, CPS, programming.

5.1 Contexte

À l'époque de progrès technologiques marquant des changements de paradigmes dans de nombreuses sphères de la société, ce que l'on décrit comme la quatrième Révolution industrielle (Forum économique mondial, 2016, 2018) a rapidement induit un changement dans la façon d'interpréter les priorités relatives aux compétences des citoyens du 21^e siècle. Que doit-on savoir et être capables de faire dans son quotidien, que ce soit dans sa vie personnelle ou sur le marché du travail ? La réponse à cela s'est manifestée dans la littérature scientifique par l'apparition d'un champ de recherche portant sur les compétences du 21^e siècle (Chalkiadaki, 2018; Griffin et Care, 2015; Ministère de l'Éducation de l'Ontario, 2016; Romero, 2017). Ces compétences, misant notamment sur la mobilisation d'outils numériques variés, offrent aux individus la possibilité de tirer profit du potentiel offert par ces outils numériques tout en développant des attitudes et comportements adéquats à leur égard. Avec le souci de répondre aux questions émergentes de l'intérêt porté à ce phénomène, le Gouvernement du Canada a mis sur pied un regroupement de spécialistes dont le mandat est notamment de proposer un plan stratégique à l'égard des nouvelles compétences recherchées, le Conseil des compétences futures (Gouvernement du Canada, 2020). Il est possible de percevoir dans cette initiative la nécessité d'étudier, de comprendre et de chercher des façons d'introduire le développement de ces compétences dites du futur chez les apprenants de tous âges au Canada. Or, outre ces compétences, il convient de savoir comment elles peuvent être mobilisées et développées ; c'est ce qui nous amène à la programmation informatique.

La programmation est une activité visant à créer des programmes, c'est-à-dire des séries de consignes définies, puis envoyées à l'ordinateur (Lu et Fletcher, 2009). Ces consignes doivent être sans équivoque et rédigées dans un langage parfaitement formalisé (Baron et Bruillard, 2001; Lopez, 1986). Le processus de création d'un programme fait appel à de nombreuses habiletés et compétences, notamment en raison de son caractère fondamentalement complexe et fondé sur l'abstraction (Guttag, 2017; Hromkovič et al., 2016a), et peut induire des apprentissages dans diverses disciplines (Papert, 1980; Romero, 2016a). L'organisme Canada Learning Code (2020) souligne d'ailleurs qu'il serait souhaitable que les élèves canadiens, à la fin de l'enseignement secondaire, possèdent des habiletés de programmation. Plusieurs de ces habiletés et

compétences figurent parmi les compétences du 21^e siècle, et nous y retrouvons de façon plus notable la résolution de problèmes et la collaboration (Barr et Stephenson, 2011; Delcker et Ifenthaler, 2017; Neubert et al., 2015; Romero, 2017; Voogt et al., 2015). La mobilisation ou le développement de la compétence en résolution de problèmes a été observé dans le cadre de nombreuses études portant sur l'utilisation pédagogique de la programmation informatique. Par exemple, la programmation permet aux élèves de « participer activement à la résolution de problèmes »³⁴ (Bers et al., 2014, p. 156). Il en va de même pour la collaboration : dans une étude quasi expérimentale menée par Moreno-León et Robles (2015) auprès de 65 élèves de 4^e et 5^e année en Espagne, ces derniers ont déclaré que la programmation avait une influence positive sur leurs habiletés de collaboration. D'autres travaux ont quant à eux démontré le lien entre la programmation et à la fois la résolution de problèmes et la collaboration (Fessakis et al., 2013; Pinto-Llorente et al., 2017).

Considérant le potentiel pédagogique de la programmation, le Gouvernement du Québec a fait le choix d'officialiser l'intérêt d'inclure la programmation dans le cursus scolaire de tous les ordres d'enseignement, mais plus particulièrement des élèves du primaire et du secondaire. Bien que de nombreuses initiatives aient déjà été documentées (Bélanger, 2016; Fondation de la Commission scolaire de Montréal, 2020; Ministère de l'Éducation et de l'Enseignement supérieur, 2018), le lancement du *Plan d'action numérique en éducation et en enseignement supérieur* a lancé un message clair : « le Plan d'action vise à ce que la programmation informatique soit utilisée dans la majorité des écoles primaires et secondaires du Québec, tant publiques que privées, d'ici l'année scolaire 2020-2021 » (Gouvernement du Québec, 2018). Dans la foulée de cette implantation de la programmation dans le système scolaire québécois, il nous semble important de comprendre son incidence sur l'apprentissage des élèves. De plus, en tenant compte du lien très documenté dans la littérature entre la programmation et les compétences de résolution de problèmes et de collaboration, il nous a paru pertinent d'orienter notre intérêt de recherche sur l'observation de la résolution collaborative de problèmes (RCP). Quels que soient le contexte ou les outils utilisés, il semble nécessaire de comprendre la mobilisation de ces compétences en salle de classe, ce qui pourrait alors mener à de nombreuses retombées intéressantes pour l'éducation.

³⁴ Traduction libre (« *children actively engage in problem-solving* »).

Dans une perspective empirique, nous postulons que la programmation en contexte pédagogique offre un contexte intéressant pour l'observation de la RCP.

Or, les méthodes d'observation de la RCP utilisées dans la littérature consultée relèvent surtout d'approches quantitatives en contextes inauthentiques. Par exemple, dans le cadre de l'enquête PISA 2015 (OCDE, 2017), on a utilisé une interface numérique dans laquelle l'élève interagissait avec un agent virtuel et dans laquelle il devait choisir les actions à poser parmi des choix de réponses³⁵. L'observation dans une perspective qualitative, en contexte authentique de salle de classe et avec des équipes d'élèves est le besoin que nous avons identifié et auquel nous souhaitons répondre avec cette étude. L'objectif de cet article est de présenter le processus ayant mené à la conception d'une grille d'observation de la résolution collaborative de problèmes (RCP). La grille a été mise à l'essai dans le cadre d'activités de programmation en collaboration à l'aide de l'ordinateur, à l'enseignement primaire.

5.2 Cadre théorique

Dans cette section, nous présentons les assises théoriques ayant orienté la conception d'une grille d'observation de la RCP. Nous y définissons la programmation informatique, ses usages pédagogiques, ainsi que les compétences qu'elle permet de mobiliser et potentiellement développer. Une attention particulière est accordée à la RCP, de même qu'aux méthodes et outils qui peuvent être utilisés pour l'observer.

Le fonctionnement de l'ordinateur repose essentiellement sur le traitement et la mémorisation d'informations (Bossuet, 1982; Guttag, 2017). N'ayant pas la capacité de réfléchir, l'ordinateur doit être contrôlé par des indications claires, précises et ne laissant pas de place à l'interprétation (Hardouin-Mercier, 1974). Ces indications claires représentent le cœur de la programmation (Turski, 1978). Elle est divisée en deux volets : l'acte de coder, qui consiste à traduire les opérations que l'on souhaite effectuer dans un langage de programmation donné (Baron et Bruillard, 2001), et l'acte de programmer, qui consiste en l'assemblage de plusieurs codes de

³⁵ Nous abordons cette méthode de façon détaillée à la section 5.2.2. de ce manuscrit.

façon séquentielle, structurée et logique, c'est-à-dire un algorithme (Henri, 2014; Rice et Desautels, 1969).

Dans le contexte pédagogique, la programmation a une double utilité : elle peut à la fois être objet et outil d'apprentissage. À l'instar de Seymour Papert (1994), Romero, Lepage et Lille (2017) soulignent que l'usage de la programmation induit un potentiel d'apprentissage important, particulièrement lorsque cet apprentissage s'effectue en collaboration. Ce caractère collaboratif inhérent à l'usage du numérique a d'ailleurs été souligné dans la littérature (Depover et al., 2007), notamment dans le champ qui s'intéresse au *computer supported collaborative learning* (CSCL), l'apprentissage collaboratif avec l'ordinateur (Koschman et al., 2002; Koschmann, 1996). Ce champ, qui souligne comment le fait d'interagir et de travailler en équipe à l'aide de l'ordinateur nourrit l'apprentissage (Dillenbourg et al., 2009; Hao et al., 2015; McConnell, 2000), peut notamment être étudié à l'aide de la théorie de l'activité (Leontiev, 1978). Cette théorie, ancrée dans l'approche socioconstructiviste (Vygotsky, 1934), offre un cadre pertinent pour aborder l'activité de programmation collaborative en contexte scolaire. Selon le modèle d'Engeström (1987), l'*activité* se décline en trois principaux éléments : le sujet (l'individu), l'objet (l'objet de l'apprentissage) et la communauté (les pairs). La médiatisation exprime la nature des interactions entre chacun de ces éléments : l'outil (numérique), la séparation du travail ainsi que les règles qui régissent la collaboration entre l'individu et les pairs.

D'un point de vue pédagogique, différentes retombées potentielles ont été associées à la programmation dans un contexte collaboratif. Nous évoquons notamment la mobilisation, voire le développement, de compétences comme la résolution de problèmes et la collaboration (Bugmann et Karsenti, 2018a; Fessakis et al., 2013; Komis et Misirli, 2013; Moreno-León et Robles, 2015; Romero et al., 2018). La résolution de problème est définie par Polya (1957) comme un processus itératif de quatre étapes qui ne saurait s'inscrire dans une chronologie linéaire. S'inscrivant dans la foulée des travaux de Polya, Jonassen (2014) a décliné de façon plus détaillée le processus de résolution de problèmes en sept étapes : la définition du problème, l'analyse du problème, la collecte d'informations, l'élaboration de pistes de solutions, l'évaluation de solutions alternatives, l'application des solutions et finalement, la vérification de celles-ci. Quant à la collaboration, elle désigne le travail qui se réalise de façon commune, où l'apport de chacun ne

saurait être discerné dans le produit final, et où le contrôle de l'enseignant s'estompe peu à peu pour laisser plus de place à l'autonomie des apprenants (Henri et Lundgren-Cayrol, 2001).

La littérature a exprimé maintes fois le lien unissant ces deux compétences (Merrill et Gilbert, 2008; Wood et al., 1976), menant à la création de la résolution collaborative de problèmes (RCP), aussi qualifiée de compétence du 21^e siècle (Care et Griffin, 2017; Neubert et al., 2015). L'intérêt accordé à la RCP avec le numérique est souligné non seulement par son étude dans de nombreux travaux au fil des années (Dillenbourg et Traum, 2006; Hanko, 2016; Hesse et al., 2015; Hurme et Järvelä, 2005; Tausczik et al., 2014), mais également par son utilisation dans l'étude PISA 2012 et 2015 de l'Organisation de coopération et de développement économiques (OCDE, 2015a, 2017). L'OCDE présente la RCP de façon matricielle, où se recoupent les composantes de la résolution de problèmes et de la collaboration. Kamga, Romero, Komis et Misirli (2017) ont proposé une déclinaison de ce modèle et l'ont mobilisé dans une étude menée auprès de 180 futurs enseignants de l'éducation préscolaire et de l'enseignement primaire (voir Annexe 3).

Ainsi, sachant que l'usage pédagogique de la programmation permet de mobiliser, voire développer la RCP et que cette dernière a fait l'objet de nombreuses études scientifiques, tout en tenant compte de l'introduction de la programmation dans les politiques éducatives au Québec, il nous semble pertinent de nous intéresser à ce phénomène. En nous référant au modèle de l'activité d'Engeström (1987), nous avons choisi l'observation comme principal outil d'étude de la mobilisation de la RCP : ce modèle présente, à l'aide d'un triangle, l'interaction entre le sujet (élève), la communauté (les pairs) et l'objet (d'apprentissage). Notre observation s'intéresse particulièrement à la médiatisation entre l'individu et les pairs (*règles*), ainsi qu'entre les pairs et l'objet d'apprentissage (*séparation du travail*). Dans cette perspective, il est clair que l'activité de l'élève et celle des pairs sont indissociables et doivent être observées puis interprétées de façon concomitante.

5.2.1 L'observation

L'observation, outil crucial pour la recherche en sciences humaines, implique « l'activité d'un chercheur qui observe personnellement et de manière prolongée » un phénomène (Jaccoud et Mayer, 1997, p. 212). Elle peut varier de passive à active. Selon Savoie-Zajc (2018), l'observation

active implique que le chercheur soit engagé et participe à la dynamique ambiante. Cette posture, issue du modèle d'imprégnation, permet au chercheur d'avoir une meilleure compréhension du phénomène observé puisqu'il en développe une compréhension au contact des participants (Fortin et Gagnon, 2016; Jaccoud et Mayer, 1997). Dans une recension portant sur l'observation directe auprès de familles, Gardner (2000) soulève que selon les études consultées, « la réactivité à l'observateur ne constitue pas une menace substantielle pour la validité des données d'observation »³⁶ (p.191). C'est donc pour ce type d'observation que nous avons opté, tout en comptant sur l'aide d'outils numériques pour faciliter le processus, c'est-à-dire l'enregistrement vidéo des séances et l'utilisation d'une grille.

5.2.2 L'observation de la RCP dans la littérature

Afin de concevoir cette grille, nous avons sondé la littérature à la recherche de grilles ou d'outils ayant trait à la RCP réalisée à l'aide du numérique. Délaissant les outils comme un questionnaire avec échelles de Likert (Yin et Abdullah, 2013), nous avons repéré quelques grilles intéressantes. L'une d'elles émanait d'une typologie des comportements en contexte de RCP (Taggar et Brown, 2001). Cela dit, il était question d'observer le comportement d'individus, ce qui ne correspond pas à notre intérêt d'observer les processus collaboratifs en soi (les comportements et les dynamiques d'équipe). Care et Griffin (2017), quant à eux, proposent un dispositif d'évaluation de la RCP en deux volets : cognitif et social. L'opérationnalisation de ce dispositif s'effectue à l'aide d'une interface virtuelle où les élèves sont appelés à réaliser des tâches de résolution de problèmes en collaboration à distance. La mesure des composantes de RCP s'appuie donc sur la base de données, qui compile de façon automatisée toutes les actions posées par les élèves ainsi que le clavardage (qui permet aux apprenants d'interagir).

Cette méthode est d'ailleurs très similaire à celle employée dans le cadre de l'étude PISA menée en 2015 (OCDE, 2017). Elle a mesuré la RCP, mobilisée entre humains (H-H) en contexte authentique, à l'aide d'une méthode ne permettant que des interactions entre un humain et un agent virtuel (H-A). Répondant à des critiques formulées à cet égard (voir notamment Zehner et al., 2019), Stadler, Herborn, Mustafic et Greiff (2020) ont publié un article confirmant la validité

³⁶ Traduction libre (« *observer reactivity does not pose a substantial threat to the validity of observational data* »).

des tâches de RCP utilisées dans l'enquête PISA 2015. Ils arrivent à la conclusion que les tâches entre un élève et un agent virtuel (H-A) et celles entre deux humains (H-H) avaient des dynamiques similaires. De plus, ils écrivent : « we found that performance in H-A tasks was moderately related to more conventional assessments of individual CPS skills ». Or, ces résultats s'opposent à d'autres travaux. Par exemple, selon Rosen et Foltz (2014) l'interaction avec des agents virtuels (H-A) crée des situations artificielles qui ne sont pas cohérentes avec l'authenticité d'une interaction humaine (H-H), cette dernière étant composée de changements constants et d'idiosyncrasies. D'autres auteurs encore adoptent une position intermédiaire en soulignant la haute validité des interactions H-H, mais en rappelant la difficulté à contrôler ces interactions dans certains contextes comme une enquête internationale de grande envergure (Greiff et al., 2013); ils désignent l'interaction H-A comme étant un « compromis acceptable ». Malgré les nombreuses réserves pouvant être exprimées à l'égard de la posture adoptée dans l'enquête PISA 2015, il demeure que cette dernière est la source principale de données empiriques associées à la RCP dans le monde. En revanche, Stadler et ses collègues (2020) évoquent la nécessité pour de futures études de s'intéresser de façon plus précise à la dimension sociale de la RCP, en contexte H-H. C'est donc dans la foulée de ces réflexions très récentes que s'inscrit notre étude, justifiant de fait la pertinence d'observer la RCP à l'aide d'une grille d'observation empirique dans un contexte authentique d'interactions H-H.

Dans ce manuscrit, nous présentons donc le processus de création d'une grille d'observation de la RCP qui puisse être utilisée dans tout contexte de résolution de problèmes en collaboration, quel que soit l'outil utilisé. L'intérêt est avant tout que cette grille puisse être utilisée tantôt par les chercheurs, dans l'analyse de la manifestation de la RCP, tantôt par les intervenants des milieux de pratique. Pour ce faire, nous avons également consulté plusieurs travaux scientifiques portant sur les grilles d'observation.

5.3 Méthodologie

À l'instar de Guikas, Morin et Bigras (2016), nous nous sommes inspiré du modèle de Strayer et Gauthier (1982), qui décline le processus de développement d'une grille d'observation en trois

phases (figure 13). Guikas et al. (2016) ont aussi proposé des pistes méthodologiques importantes sur la façon d’aborder ce processus et de le présenter.

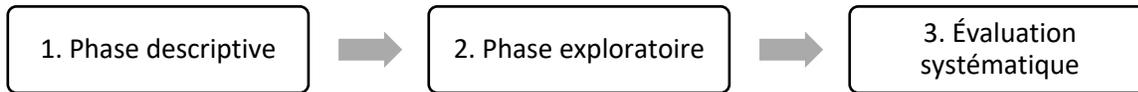


Figure 13. – Représentation graphique du modèle de Strayer et Gauthier (1982)

La phase descriptive, en amont des observations, permet tracer un portrait détaillé des concepts sur lesquels la grille est fondée. La phase exploratoire permet de développer la grille de façon opérationnelle puis de la mettre à l’essai de façon tentative. Enfin, l’évaluation systématique consiste en l’application de la grille auprès d’un corpus de données. Ces trois phases seront explicitées dans cette section

5.3.1 Phase descriptive

Cette première phase a pour but de favoriser une compréhension approfondie du phénomène observé par le chercheur. Pour ce faire, Strayer et Gauthier (1982) suggèrent de combiner une recension de la littérature à une observation informelle afin d’obtenir un portrait complet des composantes de ce phénomène, en plus d’entrer en contact avec le milieu dans lequel la recherche se déroulera.

5.3.2 Phase exploratoire

La phase exploratoire du modèle de Strayer et Gauthier (1982) se caractérise par une mise en application expérimentale de la grille, en contexte réel. Cet outil, dont la validité théorique a été démontrée par une recension de la littérature et un comité d’experts, devait alors être soumis à l’épreuve du terrain. Il s’agit donc d’une étape fondamentale visant à assurer sa cohérence au sein d’un contexte pédagogique concret et authentique. Les milieux de recherche ciblés, des écoles primaires des régions de Montréal et de Québec, seront présentés plus loin. Plusieurs considérations relatives à l’utilisation de la grille sont soulevées à cette étape : déterminer l’organisation des groupes lors des collectes de données, l’organisation des équipes dans l’environnement de la classe ou de la salle, la méthode d’enregistrement la plus appropriée considérant le bruit ambiant et le risque qu’il affecte la compréhension des propos des

participants, etc. Ces considérations ont été consignées dans le journal de bord (Savoie-Zajc, 2018) pour référence future. En nous référant à nos expériences antérieures dans ces milieux, et en adéquation avec les fondements théoriques et conceptuels de notre grille, nous avons convenu que l’observation participante (Jaccoud et Mayer, 1997; Savoie-Zajc, 2018) jumelée à une analyse informatique *a posteriori* serait la méthode la plus adéquate pour observer le phénomène qui nous intéresse, c’est-à-dire la mobilisation des compétences de RCP en contexte scolaire. Selon van der Maren (2004), bien que l’utilisation de caméras et d’autres dispositifs d’enregistrement en classe puisse avoir des effets indésirables, notamment un potentiel sentiment d’inconfort chez les participants, elle permet une plus grande profondeur de l’analyse par des reprises de l’observation, c’est-à-dire le fait de visionner les enregistrements à nouveau.

5.3.3 Évaluation systématique

La troisième et dernière phase du développement d’une grille d’observation, selon Strayer et Gauthier (1982), est l’évaluation systématique. Cette dernière consiste à utiliser la grille finale dans un corpus de données suffisamment grand et diversifié. Pour ce faire, nous avons observé le déroulement d’un scénario pédagogique de programmation intitulé *Deviens un maître NAO* dans trois écoles d’enseignement primaire du Québec, auprès de 109 élèves. Ces trois écoles se trouvent dans des milieux différents, notamment quant au contexte socioéconomique, à l’emplacement géographique et au type d’enseignement (tableau 12).

École	Groupe	Année	Élèves (<i>n</i>)
École A	A1	Multi	29
École B	B1	4 ^e année	14
	B2	5 ^e année	17
	B3	5 ^e année	20
	B4	6 ^e année	19
École C	C1	5 ^e année	10
			109

Tableau 12. – Participants

La constitution de l’échantillon n’a pas fait l’objet de critères d’inclusion ou d’exclusion, si ce n’est, pour les directions d’établissement et les enseignants, d’accepter de participer à la mise en place

du scénario pédagogique. L'école A est une école alternative de Montréal, où les élèves étaient aux deuxième et troisième cycles du primaire, et quelques-uns de niveau secondaire (n=29). L'école B, aussi à Montréal, est une école primaire située dans un quartier défavorisé. Les élèves des classes observées étaient en quatrième, cinquième et sixième année (n=70). L'école C, dans la région de Québec, est une école en milieu rural. La classe ciblée par l'observation comporte un nombre réduit d'élèves (n=10) en raison des difficultés éprouvées par les élèves qui la composent.

5.4 Résultats

5.4.1 Phase descriptive

Notre recension de la littérature a permis d'offrir des balises conceptuelles et théoriques à l'égard de notre intérêt de recherche principal, c'est-à-dire la mobilisation des compétences de résolution collaborative de problèmes (RCP) à l'aide du numérique (plus particulièrement de la programmation). La littérature pertinente était donc variée dans la mesure où nous avons approfondi le volet plus technique de la programmation afin de bien comprendre comment cette activité pouvait interagir avec la RCP, de même que le volet conceptuel rattaché à la mobilisation de compétences, à la résolution de problèmes, à la collaboration, ainsi qu'à la RCP en tant que telle. Le lien entre ces concepts s'est concrétisé dans le champ du CSCL (*computer supported collaborative learning*), pour lequel l'approche socioconstructiviste et la théorie de l'activité représentent des perspectives épistémologiques intéressantes. Pour compléter ces données empiriques et théoriques provenant de la littérature, Strayer et Gauthier (1982) proposent d'effectuer des observations informelles sur le terrain de la recherche. Or, cette étape ne s'est pas avérée nécessaire dans le cadre de cette étude. En effet, l'échantillon et le terrain de recherche étaient déjà connus avant le début de cette étude ; nous avons déjà eu l'occasion de mener d'autres projets de recherche dans ces milieux. Ainsi, nous possédions une très bonne compréhension des milieux dans lesquels allait se dérouler la collecte de données et avons déjà tissé des liens avec les intervenants impliqués.

Cette connaissance préalable du terrain de recherche a été jumelée aux travaux scientifiques qui nous ont permis d'identifier les composantes observables de la RCP (Care et Griffin, 2017; Greiff

et al., 2013; Hesse et al., 2015; Kamga et al., 2017; OCDE, 2015a, 2017), afin de concevoir une première version de la grille d'observation de la RCP. Cette première itération était constituée de trois dimensions au sein desquelles étaient déclinés des indicateurs, c'est-à-dire des unités d'observation (Savoie-Zajc, 2018). La première dimension, *Établissement et maintien d'une compréhension partagée*, était composée de cinq indicateurs ayant notamment trait au point de vue des membres de l'équipe, à la compréhension du problème à résoudre et à l'établissement d'une communication claire. La seconde dimension, *Réalisation des actions appropriées à la résolution du problème*, présentait six indicateurs portant sur les étapes du processus de résolution de problèmes, que ce soit quant à la réalisation de tâches, la vérification de ces dernières, ou à l'utilisation de pistes de solutions des pairs pour en élaborer d'autres. La troisième dimension, *Établissement et maintien de l'organisation de l'équipe*, contenait cinq indicateurs surtout associés à l'attribution de rôles et à la reconnaissance des forces et des faiblesses des pairs. Dans cette première version de la grille, chaque indicateur était formulé dans une visée opérationnelle, c'est-à-dire de façon à décrire un comportement observable (Champoux et al., 1992).

La grille a par la suite fait l'objet d'une vérification auprès d'experts issus du domaine de la recherche en psychopédagogie, et plus particulièrement du numérique en éducation. Ces quatre experts, dont trois sont professeurs en sciences de l'éducation et un chercheur-praticien, ont reçu une version de la grille accompagnée d'un texte expliquant le contexte de notre étude, quelques repères théoriques, ainsi que l'utilisation anticipée de cet outil sur le terrain. Les commentaires de ces experts ont mis en lumière des problèmes relatifs à la formulation des énoncés des indicateurs, qui étaient parfois difficilement observables en termes d'occurrence, et des problèmes de cohérence ou de mauvaise complémentarité entre certains indicateurs. À l'issue de cet exercice de validation, c'est-à-dire après la modification, l'ajout ou la suppression d'indicateurs au regard des commentaires formulés, nous avons produit une seconde version de la grille.

5.4.2 Phase exploratoire

Les observations préliminaires ont mené à la constitution d'un corpus de données restreint dans lequel nous avons puisé pour mettre la grille à l'essai. Le matériel représente 168 minutes d'enregistrements vidéos effectués dans 2 classes de 2 écoles, auprès d'équipes d'élèves de 4^e à 6^e année en action. Globalement, cela a induit plusieurs changements de la grille, notamment quant aux définitions opérationnelles (empiriques), qui ont été ajustées pour mieux refléter la réalité, et à la formulation d'exemples et de contre-exemples de comportements observés pour chaque indicateur. De plus, des indicateurs ont été modifiés, d'autres réorganisés. Nous avons d'ailleurs invité des étudiants du baccalauréat, qui allaient aussi agir à titre de codeurs lors du traitement et de l'analyse des données, afin qu'ils commentent la grille et son opérationnalisation. Cette pratique, recommandée par van der Maren (2004), consiste à jumeler la formation des codeurs au processus de conception de la grille, ce qui permet notamment d'éviter des problèmes de disparité du codage (Harris et Lahey, 1982).

Cette seconde phase du développement a donc mené à l'élaboration d'une troisième version de notre grille d'observation, qui a dirigé la conception du système de codification des données. Ce dernier repose sur le principe d'observation continue : toutes les apparitions du comportement sont codées selon différents aspects, comme la fréquence, la durée et l'intensité (Champoux et al., 1992). Les indicateurs de fréquence se rattachent au premier aspect correspondant et les indicateurs de séance renvoient au troisième aspect, l'intensité. La durée des comportements n'a pas été utilisée dans ce processus de conception de la grille, mais il serait possible de le faire dans le cadre de futurs travaux. Il est important de mentionner que la finalité de la grille est de comprendre *comment* sont mobilisées les compétences de RCP, ce qui implique bien plus qu'un calcul des fréquences de chaque indicateur. Le processus de codage préliminaire fut de type mixte (van der Maren, 2004), dans la mesure où le codage a été entamé à l'aide d'une liste qui a ensuite été ajustée à la lumière des constats tirés du processus en soi. Chaque indicateur de la grille s'est vu attribuer un code, formant ainsi des unités de sens permettant de repérer et classer les comportements ciblés. Le codage du corpus de données s'est effectué à l'aide du logiciel NVivo 12 (QSR International, 2020), que nous avons retenu en raison des fonctionnalités qu'il offre pour l'analyse manuelle de sources nombreuses et variées (Roy et Garon, 2013). Les indicateurs de la

grille ont été transposés dans l'interface du logiciel, puis les fichiers vidéos ont été importés dans le logiciel.

En outre, au cours de l'analyse des données de l'observation préliminaire, nous avons constaté que certains indicateurs n'étaient pas du même ordre que les autres : certains font référence à des moments précis (incidences d'un comportement), alors que d'autres font référence à un ensemble d'incidences d'un comportement, menant à la formulation d'une interprétation générale applicable à l'intégralité de la séance³⁷. Cela a donc mené à préciser et enrichir la grille par la distinction de deux types d'indicateurs : des indicateurs d'occurrence (comportement observable) et des indicateurs de séance (interprétation portée sur un ensemble d'occurrences d'un comportement, au cours d'une séance donnée). Les indicateurs de séance sont présentés sous forme d'échelle de 1 (faible/rarement) à 3 (élevé/souvent). Par exemple, constatant qu'il était peu pertinent et surtout difficilement réalisable de coder toutes les occurrences de l'utilisation d'un vocabulaire précis, nous avons choisi de définir *Communiquer clairement* comme un indicateur de séance.

5.4.3 Évaluation systématique

Lors des visites, les élèves se voyaient remettre le matériel nécessaire au déroulement du scénario pédagogique de programmation. En équipes de trois à cinq personnes, les élèves étaient appelés à réaliser une série de 20 tâches qui consistaient principalement à animer un robot humanoïde nommé NAO. Doté de plusieurs capteurs tactiles et de fonctionnalités avancées, comme la synthèse vocale et la reconnaissance visuelle, le robot est en mesure d'entamer des actions et réagir en fonction de stimuli visuels, sonores ou tactiles. Pendant l'activité, des intervenants circulaient pour offrir une assistance aux équipes bloquées à un niveau ; cette aide était proposée uniquement lorsque les élèves avaient mis en œuvre différentes stratégies infructueuses pour résoudre le problème. Parmi ces intervenants, on compte le chercheur, deux assistants de recherche, ainsi que l'enseignant. Il est important de souligner que les problèmes techniques associés au fonctionnement du robot étaient résolus par les intervenants adultes,

³⁷ Le terme « séance » renvoie à la période pendant laquelle les élèves ont participé à l'activité de programmation dans une journée. Ces séances, selon les établissements, étaient d'une durée de 60 à 180 minutes.

puisqu'ils exigent des connaissances particulières et avancées. Les observations se sont déroulées de septembre 2019 à janvier 2020. Trois séances ont été effectuées dans chacune des classes de notre échantillon. Pour la consignation des données, nous avons installé une caméra devant chaque équipe, filmant l'ordinateur et le robot. Ces enregistrements, qui offrent une vue d'ensemble de l'activité des élèves, ont été analysés à l'aide de notre grille d'observation.

À partir du moment où nous avons lancé l'évaluation systématique de notre grille, c'est le codage fermé (van der Maren, 2004) qui prévaut, c'est-à-dire que le corpus n'a eu aucune incidence sur la composition de la grille, comme c'était le cas lors de la phase exploratoire. Ce type de codage est associé à une meilleure fidélité intercodeur et intracodeur, puisque sa composition fait en sorte que la grille peut être utilisée par des non-spécialistes. Le codage des fichiers vidéos du corpus a été réalisé avec le concours de deux étudiants de premier cycle universitaire. Leur participation a commencé dès la phase exploratoire, où ils ont été en mesure d'être des témoins actifs du processus de conception de la grille à partir des données de la littérature scientifique et des observations préliminaires. Avant de débiter le codage individuel avec la plus récente version de la grille, nous avons procédé au codage collectif d'un fichier vidéo avec ces étudiants. Le but de cette séance était de verbaliser les réflexions entourant la codification d'un comportement à l'aide de tel ou tel code. Ce fut également l'occasion pour les étudiants de poser des questions sur certains aspects plus précis de la grille, notamment en lien avec la frontière entre deux indicateurs. Par la suite, l'équipe a procédé à un codage individuel d'une portion du corpus afin de mesurer un accord intercodeur. Une fois la mesure acceptable, nous avons entamé le codage de l'ensemble des données à l'aide de la grille. En outre, pour assurer l'uniformité du codage, nous avons procédé au calcul de l'accord intercodeur à plus d'un moment dans le processus de codage (Harris et Lahey, 1982). La première mesure a été faite au départ (79,55%) et la seconde a été faite en cours de codage (80,68%), approchant la fin. Soulignons que la mesure de l'accord intercodeur n'a pas été sans obstacle. Nous croyons pertinent d'évoquer ici ces écueils pour contextualiser ce processus et offrir des pistes de réflexion.

5.4.3.1 Précisions sur la mesure de l'accord intercodeur

Le pourcentage obtenu par le calcul de l'accord intercodeur (ou interjuge) représente le coefficient d'accord entre deux codeurs, c'est-à-dire le nombre d'instances où ils étaient en accord dans l'intégralité du codage.

$$\text{Accord intercodeur (\%)} = \frac{\text{Nombre de segments codés communs}}{\text{Nombre total de segments codés}}$$

Alors que Miles et Huberman (2003) suggèrent un accord minimal de 70%, Cohen (1960) propose plutôt un coefficient minimal de 75% avant d'entreprendre le codage individuel. D'emblée, nous avons choisi d'opter pour un accord minimal de 75%. Ainsi, en vue de calculer cet accord, nous avons entamé le codage sur une portion de notre corpus, c'est-à-dire environ 14% des fichiers. Une fois l'opération complétée dans le logiciel d'analyse de données NVivo, nous avons appliqué la fonction de mesure automatisée du Kappa (κ) de Cohen (1960). Le coefficient général obtenu était à de 0,3 et plusieurs codes avaient un coefficient de zéro. Après vérification, nous avons constaté que le logiciel NVivo calculait le coefficient par unité de temps, ce qui avait pour effet de faire diminuer drastiquement le véritable pourcentage d'accord. En effet, l'obtention du coefficient exact dépendrait d'un encodage effectué à la seconde près, ce qui n'est pas réaliste. Ainsi, nous avons opté pour un calcul manuel de l'accord intercodeur : chaque segment encodé a été comparé entre les codeurs. Lorsque l'encodage pour un segment présentant un comportement donné était présent chez les deux codeurs, il y avait accord. Cette méthode, bien que chronophage, permet de calculer un accord intercodeur particulièrement compréhensif.

5.5 Discussion

Au terme de ces trois volets du processus de conception d'une grille d'observations, nous sommes en mesure de présenter une version finale qui prend appui tant dans la littérature scientifique que dans sa mise à l'essai en contexte réel, dans un corpus de données vaste et diversifié. La grille présente les indicateurs regroupés en dimensions, les descripteurs des indicateurs, ainsi que des exemples tirés de nos observations. La grille est présentée de façon détaillée au tableau 13. Une autre version, plus synthétique et destinée aux chercheurs ou aux praticiens, se retrouve à l'Annexe 4.

Les trois grandes dimensions proviennent du modèle proposé par l'OCDE. Les énoncés, qui représentent les composantes de la RCP, ont été repris intégralement. Ensemble, ces dimensions créent un tout cohérent permettant d'aborder chacune des étapes du processus de résolution de problème en collaboration. La première, *Établissement et maintien d'une compréhension partagée*, et la dernière, *Établissement et maintien de l'organisation de l'équipe*, se rattachent notamment à la composante collaborative de la RCP. Quant à la seconde dimension, *Réalisation des actions appropriées à la résolution du problème*, elle est davantage orientée vers le processus de résolution de problème en soi. Ainsi, les trois dimensions proposent un modèle intégré de la RCP. Les indicateurs qui s'y trouvent ont été formulés en tenant compte de la littérature (Care et Griffin, 2017; Kamga et al., 2017; OCDE, 2017), tout en tirant profit des connaissances expérientielles acquises au cours de la réalisation de projets de recherche antérieurs en lien avec la mise en place de scénarios pédagogiques fondés sur la collaboration en contexte scolaire. Concis et formulés à l'infinitif, ils évoquent clairement les actions auxquelles ils renvoient de façon à simplifier l'utilisation de la grille. Nous avons également identifié à l'aide d'astérisques les indicateurs de séance, c'est-à-dire les indicateurs qui nécessitent une forme d'interprétation de la part de l'utilisateur de la grille quant à la manifestation d'un comportement de façon globale pendant la séance observée.

Les descripteurs correspondant à chaque indicateur ont été formulés à partir de la littérature, mais surtout en collaboration avec les experts consultés, de même que les étudiants de premier cycle ayant contribué au processus d'analyse des données. Ces descripteurs décrivent de façon empirique la manifestation du comportement désigné par chaque indicateur, en mettant les élèves au cœur de celui-ci (*Les élèves [verbe d'action]...*). Enfin, les exemples (et contre-exemples) ont été tirés des observations réalisées dans le cadre du processus de conception de la grille, ce qui assure une forme de cohérence entre les volets scientifique et pratique.

Indicateur	Descripteur	Exemples
D1. Établissement et maintien d'une compréhension partagée		
1.1. Échanger	Au cours du processus de RP, les élèves échangent à propos des caractéristiques inhérentes au problème à résoudre et s'assurent d'en avoir une compréhension partagée.	Les élèves discutent pour tenter de déterminer s'ils doivent faire avancer le robot de 0,5 m ou de 5 m.
1.2. Solliciter les points de vue	Les élèves cherchent à connaître le point de vue des autres membres de l'équipe quant au problème à résoudre.	Un élève demande à un pair ce qu'ils devraient faire ensuite, selon lui.
1.3. Adapter les interventions	Les élèves adaptent leurs interventions en fonction d'un ou de plusieurs coéquipiers afin de maintenir la compréhension de la tâche.	Dans une équipe multiniveau, un élève plus vieux offre une définition vulgarisée d'un terme en anglais afin qu'un élève plus jeune puisse comprendre.
D2. Réalisation des actions appropriées à la résolution du problème		
2.1. Mettre en œuvre les actions	Les élèves discutent de la façon d'effectuer une ou plusieurs actions (séquence) pour résoudre le problème, pendant l'activité (on ne cherche pas à savoir quelles actions doivent être faites [<i>Échanger</i>], on cherche à savoir comment les mettre en œuvre).	Après avoir paramétré la boîte pour faire avancer le robot de 0,5 m, des élèves signalent qu'il faut ajouter une boîte <i>Say</i> .
2.2. Vérifier les actions	Les élèves vérifient que les actions ont produit les effets escomptés. Il s'agit d'une vérification partielle de la solution.	Dans un programme au cours duquel le robot doit faire deux mouvements, les élèves vérifient que le premier mouvement est adéquat avant de poursuivre le programme.
2.3. Vérifier la solution	Les élèves vérifient que l'ensemble des actions a permis de résoudre le problème. Il s'agit d'une vérification de la solution intégrale.	Dans un programme au cours duquel le robot doit faire deux mouvements, les élèves vérifient que ces derniers ont été effectués adéquatement en vue de confirmer la résolution du problème donné.
2.4. Adapter les solutions inadéquates	Le cas échéant, les élèves adaptent les solutions inadéquates dans le but de les rendre adéquates.	Après avoir constaté que le robot lève le bras droit alors qu'il devait lever le bras gauche, les élèves modifient les paramètres d'un code pour rectifier la situation.
2.5. Participer activement*	De façon générale, les élèves participent activement aux tâches en vue de résoudre le problème.	La plupart des membres de l'équipe sont physiquement et verbalement impliqués et près du robot, manipulent le matériel fréquemment, etc.) [élevé]. OU La plupart des élèves sont éloignés du matériel, regardent par la fenêtre, parlent peu [faible].
2.6. Persévérer*	De façon générale, les élèves persèverent dans l'activité de résolution de problème.	Malgré un bogue survenu dans le programme et quelques vérifications de solution échouées, les élèves ont poursuivi leur travail et sont parvenus à trouver une solution adéquate [élevé]. OU Après avoir constaté un bogue dans le programme, les élèves cessent le travail, ne regardent plus l'écran ou le robot, et discutent ensemble [faible].

D3. Établissement et maintien de l'organisation de l'équipe

3.1. Attribuer des rôles	Les élèves se sont attribués, entre eux, explicitement ou implicitement, des rôles. Cette attribution peut être unique, en amont de l'activité, ou itérative, selon les conditions mises en place par les membres de l'équipe. Il est possible, voire souhaitable, que des équipes effectuent des rotations constantes.	Deux élèves suggèrent que Magalie s'occupe de faire le programme sur l'ordinateur.
3.2. Identifier les forces et faiblesses	Les élèves identifient les forces et les faiblesses des membres de l'équipe.	Un élève dit : « Wow, tu es vraiment bon pour trouver les codes dans la librairie ». OU Un élève dit : « Laisse-moi chercher le code, tantôt ça t'a pris beaucoup trop de temps pour le trouver ».
3.3. Résoudre des différends	Les élèves sont capables de résoudre les différends ou parviennent à un consensus lorsqu'il y a des divergences d'opinions avec conflit (les divergences d'opinions sans conflit peuvent survenir lors des échanges et de la sollicitation des points de vue sans qu'il y ait de différend à résoudre).	Des élèves s'obstinent sur le code qu'ils devraient utiliser, puis commencent à monter le ton et à s'insulter. Les autres élèves du groupe parviennent à désamorcer la situation.
3.4. Communiquer clairement*	De façon générale, les élèves utilisent des termes précis, spécifiques ou techniques pour établir une communication claire quant à la tâche à réaliser.	Les élèves utilisent les termes associés à l'outil utilisé plutôt qu'un synonyme provenant du langage courant et peu précis (chose, affaire, truc, etc.). <i>La gradation de cette composante se fait de rarement (1) à souvent (3).</i>
3.5. Agir en tant qu'équipe*	De façon générale, les élèves démontrent un intérêt pour la cohésion du groupe et le travail en collaboration.	Les élèves utilisent le pronom « nous » et continuent d'être intéressés aux actions même lorsqu'elles sont posées par d'autres élèves [élevé]. OU Un élève choisit d'effectuer les actions seul pendant que les autres membres de l'équipe regardent et sont peu engagés [faible].
3.6. Respecter les rôles attribués*	De façon générale, les élèves respectent les rôles qu'ils se sont attribués, explicitement ou implicitement. Cette attribution peut être unique, en amont de l'activité, ou itérative, selon les conditions mises en place par les membres de l'équipe.	Un élève dit : « Non, c'est moi qui m'occupe de bouger le robot, occupe-toi de l'ordinateur ». Il est possible, voire souhaitable, que des équipes effectuent des rotations constantes. Le respect des rôles est alors affecté lorsque des membres de l'équipe se disputent un même rôle et que cela entrave le bon déroulement de la résolution de problème.

* Les astérisques désignent des indicateurs de séance.

Tableau 13. – Grille d'observation de la RCP

5.5.1 Justesse de la grille

Tout au long du processus de conception de la grille, nous avons fait appel à de nombreuses ressources pour trianguler nos résultats et ainsi conférer à la grille une plus grande justesse, nous éloignant ainsi d'un exercice purement théorique. Comme nous l'avons mentionné à de nombreuses reprises, les fondements de cette grille reposent sur la littérature scientifique qui traite de résolution de problèmes, de collaboration, de RCP, d'utilisation du numérique et d'usages pédagogiques de la programmation. Or, divers moyens supplémentaires ont été employés pour donner à la grille un meilleur ancrage dans le contexte scolaire, de même qu'une constitution compréhensive et accessible à des non-experts. Nous avons fait appel à des experts, professeurs d'université et praticiens, de même qu'à des étudiants de premier cycle universitaire. En outre, nous nous sommes appuyé sur le modèle de Strayer et Gauthier (1982), à l'instar de Guikas et al. (2015), pour inscrire la création de cette grille dans une méthode avérée et structurée. Enfin, la vérification de l'accord intercodeur a démontré la solidité de la grille quant aux aspects conceptuels et opérationnels.

5.5.2 Usages potentiels de la grille

Le principal point fort de cette grille est sa vérification empirique. Nous avons eu la démonstration qu'elle peut être utilisée dans un contexte réel auprès d'élèves en contexte pédagogique authentique. Contrairement aux dispositifs employés dans l'étude PISA 2015, par exemple, notre grille repose sur un principe interprétatif fondamental, mettant ainsi l'humain au centre du processus, que ce soit quant aux comportements observés ou aux personnes administrant la grille. D'ailleurs, le contexte d'application proposé par les principales grilles trouvées dans la littérature implique l'utilisation d'interfaces numériques où l'élève peut résoudre un problème en collaboration avec un agent virtuel. Nous avançons que cette méthode, bien qu'utile dans une perspective logistique de collecte de données à l'échelle internationale, montre d'importantes lacunes quant à l'authenticité des résultats obtenus. Ces derniers reposent sur des interactions effectuées entre un humain et un agent virtuel, qui sont tout à fait dépourvues de la complexité inhérente aux interactions entre humains (langage non verbal, idiosyncrasies, etc.). Il est également à noter que la grille proposée dans cet article permet d'observer des comportements

d'équipe et non uniquement des comportements individuels. L'approche socioconstructiviste et la théorie de l'activité offrent ici des principes au cœur du déroulement d'une activité en collaboration en contexte réel. En somme, la grille d'observation de la RCP que nous proposons est qualitative et interprétative. Cela fait en sorte de la rendre utilisable dans une variété de contextes, quels que soient les outils utilisés et le contexte dans lequel se déroule l'activité de résolution de problème en collaboration à l'aide du numérique. Or, l'usage du numérique n'est pas une condition *sine qua non* à l'application de la grille. En effet, si l'on fait fi des exemples proposés, les indicateurs et les descripteurs permettent d'observer une résolution collaborative de problèmes en contexte débranché. Les principaux utilisateurs anticipés de cette grille sont les chercheurs en sciences de l'éducation, mais nous considérons également que cette grille est destinée aux praticiens qui souhaitent orienter leurs pratiques en se fondant sur des données probantes issues de la recherche. C'est d'ailleurs pour ces raisons qu'une version « prête à être utilisée » est présentée à l'Annexe 3 de ce manuscrit.

5.6 Limites

Bien que cette grille ait été mise à l'essai dans un contexte réel et que son caractère empirique ait pu être confirmé, il demeure que ces résultats sont difficilement généralisables. Ce n'était d'ailleurs pas l'objectif fixé. En effet, les recherches fondées sur l'observation de cas sont très utiles pour comprendre un contexte particulier, mais permettent difficilement d'étendre les conclusions à grande échelle, considérant les particularités inhérentes à chaque milieu et les nombres de variables qu'il est difficile d'isoler (Karsenti et Demers, 2018). La diversification de notre échantillon a toutefois permis d'effectuer des observations dans trois milieux très différents les uns des autres, atténuant ainsi cette limite. Notons également que les indicateurs *Échanger* et *Solliciter les points de vue* ont posé problème à différentes étapes du processus de conception de la grille. En effet, la frontière entre ces deux éléments était parfois difficile à déterminer pour les codeurs. Malgré les nombreuses modifications et précisions apportées à ces deux indicateurs, de même que les exemples qui permettent d'illustrer les comportements ciblés, il est possible que d'autres utilisateurs éprouvent un problème similaire. Les travaux subséquents en lien avec cette grille permettront probablement de proposer des exemples d'applications

supplémentaires, dans une plus grande variété de contextes, par exemple où un robot humanoïde n'est pas utilisé.

5.7 Conclusion

Cet article a présenté le processus de conception d'une grille d'observation empirique de la mobilisation de la résolution collaborative de problèmes (RCP). Les trois phases du modèle de Strayer et Gauthier (1982), de même que les travaux de Guikas, Morin et Bigras (2015) qui s'en sont inspirés, ont offert une structure qui a guidé la création de cet outil. En combinant la littérature scientifique, la consultation d'experts, la collaboration d'étudiants de premier cycle, le codage du corpus de données et la validation de ce codage par le calcul d'un accord intercodeur, nous considérons avoir réuni les conditions nécessaires à la création d'une grille dont la pertinence scientifique et pratique a été démontrée. Notre grille pourrait donc être fort utile pour la réalisation de recherches qualitatives interprétatives futures dans le champ de la RCP. Elle pourrait également être utilisée dans les milieux de pratiques, par les enseignants ou d'autres intervenants, pour orienter des interventions ou des pratiques pédagogiques adaptées au contexte de la classe et au profil des élèves.

Avec l'intégration de plus en plus marquée de la programmation dans le contexte scolaire, ce que l'on peut notamment constater par la parution de documents ministériels comme le *Cadre de référence de la compétence numérique* (Ministère de l'Éducation et de l'Enseignement supérieur, 2019), de même que du guide intitulé *L'usage pédagogique de la programmation informatique* (Ministère de l'Éducation, 2020a), il s'avère nécessaire d'étudier et d'observer les impacts que produit l'usage de la programmation en tant qu'activité mobilisant la résolution de problèmes en collaboration. En effet, tel que nous l'avons évoqué dans notre cadre théorique, l'adéquation entre la programmation et la RCP est indéniable. Cette grille est un outil dont l'utilité est considérable à cet égard. Nous croyons qu'il serait pertinent, dans le cadre de futurs travaux, d'étendre la portée de cette grille en vérifiant son application auprès d'élèves plus âgés, mais aussi auprès d'étudiants du postsecondaire, que ce soit dans un contexte directement lié à l'utilisation de la programmation, ou encore dans un contexte tout à fait distinct, sans l'utilisation d'appareils numériques pour médiatiser la collaboration des apprenants

Références

Les références de cet article sont présentées à la section *Références bibliographiques* de cette thèse.

Chapitre 6 - Comprendre le processus de résolution collaborative de problèmes (RCP) lors d'une activité de programmation informatique avec des élèves du primaire

ARTICLE 3

RÉSUMÉ

Cet article présente les résultats d'une étude de cas multiple visant à comprendre la mobilisation d'habiletés de résolution collaborative de problèmes (RCP) à l'aide de la programmation. Nous avons observé, à l'aide d'une grille d'observation inédite, le déroulement d'un scénario pédagogique ayant pour objectif d'introduire des élèves du primaire à la programmation visuelle et tangible. Une centaine d'élèves du primaire de trois écoles du Québec ont participé à ces activités. Les résultats démontrent que la RCP est bel et bien mobilisée lors d'une activité de programmation en contexte scolaire. Cette mobilisation n'est toutefois pas uniforme, dans la mesure où certaines habiletés se démarquent, comme les échanges entre les membres d'une équipe en vue de comprendre le problème, la mise en œuvre des actions visant à résoudre le problème, la vérification des solutions ainsi que l'attribution des rôles. Considérant la démocratisation de la programmation informatique dans le contexte scolaire, cette étude permet de contribuer aux réflexions portant sur les pratiques pédagogiques à favoriser et sur les principaux aspects auxquels porter attention lorsque l'on souhaite développer la RCP d'apprenants à l'aide de la programmation informatique.

Mots-clés : programmation informatique, résolution collaborative de problèmes (RCP), compétences du 21^e siècle, compétences transversales.

ABSTRACT

This article presents the results of a multiple case study to observe collaborative problem solving (CPS). Using a novel observation grid, we observed the unfolding of a pedagogical scenario aimed at introducing primary school students to visual and tangible programming. Approximately one hundred elementary students from three Quebec schools participated in these activities. The results show that CPS is indeed mobilized during a programming activity in a school context. However, this mobilization is not uniform, as certain skills stand out, such as exchanges between team members to understand the problem, implementation of actions to solve the problem, verification of solutions, and role attribution. This study makes it possible to contribute to reflections on the pedagogical practices to be promoted and on the main aspects to pay attention to when one wishes to develop learners' CPS with the help of computer programming.

Keywords: computer programming, collaborative problem solving (CPS), 21st century skills, transversal skills.

6.1 Contexte

L'émergence du numérique comme outil d'apprentissage est bien loin d'être un phénomène nouveau. Ce sont les usages, découlant des avancées technologiques novatrices, qui mettent en évidence le potentiel renouvelé du numérique à l'école. Ces usages répondent aux nouveaux besoins des apprenants du 21^e siècle, qui se distinguent à plusieurs égards des besoins de la conjoncture précédente ; l'expression *compétences du 21^e siècle* illustre bien cette réalité (Ananiadou et Claro, 2009; Fadel et Trilling, 2009; Romero, 2017; van Laar et al., 2017). Or, l'évolution des technologies et de leurs usages est continue, ce qui explique l'apparition de concepts plus récents comme les *compétences futures* (Gouvernement du Canada, 2020), qui sont ancrées profondément dans les réalités changeantes du marché du travail. Ce dernier, au gré des innovations technologiques, change rapidement et exige des qualifications de plus en plus développées quant à l'utilisation de certains outils numériques (Lee et al., 2018), un phénomène que le Forum économique mondial qualifie de quatrième Révolution industrielle (Forum économique mondial, 2016).

Or l'employabilité et l'économie ne sauraient être les seules considérations associées à l'usage du numérique. Certains outils numériques ont de nombreuses applications potentielles en éducation et leur démocratisation nous apparaît souhaitable. Nous avons choisi de nous intéresser à la programmation, qui est à la fois un objet et un outil d'apprentissage. Cet outil ne doit pas être considéré comme une fin en soi, mais plutôt comme un levier vers de nombreux apprentissages et la mobilisation de compétences transversales. Dans son rapport présentant les résultats d'une étude de cas multiples, Barma (2018) recommande d'ajouter la pensée informatique comme compétence transversale dans le Programme de formation de l'école québécoise (PFÉQ) au primaire et au secondaire. L'ajout de la programmation au cursus scolaire est d'ailleurs une décision prise par plusieurs pays à travers le monde, notamment en Finlande (Toikkanen, 2015), au Royaume-Uni (Department for Education, 2013) et plus récemment en Australie (Australian Government, 2020). Au Canada, il revient aux provinces de décider de leurs programmes de formation, ce qui explique l'absence de consensus à l'échelle nationale.

Au cours des dernières années au Québec, il a été possible d'observer années une succession de mesures gouvernementales misant sur les retombées positives du numérique pour l'apprentissage, notamment avec la programmation informatique. Ce mouvement a débuté concrètement en 2018 avec le lancement du Plan d'action numérique en éducation et en enseignement supérieur, qui visait « une intégration efficace et d'une exploitation optimale du numérique au service de la réussite de toutes les personnes » (Gouvernement du Québec, 2018, p. 9). Parmi les mesures phares de ce plan d'action se trouvait l'élaboration d'un Cadre de référence de la compétence numérique, qui a été publié l'année suivante (Ministère de l'Éducation et de l'Enseignement supérieur, 2019). Ce document a non seulement permis d'illustrer de quelle façon les différentes dimensions de la compétence numérique pouvaient s'arrimer à un contexte scolaire, mais a surtout permis d'indiquer comment la programmation pouvait être considérée dans cet ensemble de dimensions. Bien que n'étant pas prescriptif quant à l'usage de la programmation, ce cadre de référence jetait néanmoins les fondations sur lesquelles allaient se construire d'autres mesures. Ainsi, dans une optique beaucoup plus concrète et pratique, le ministère de l'Éducation du Québec a publié un guide permettant d'accompagner les enseignantes et les enseignants dans leur appropriation de la programmation en tant qu'outil pédagogique (Ministère de l'Éducation, 2020a). Enfin, en décembre 2020, le nouveau référentiel de compétences professionnelles de la profession enseignante a été publié (Ministère de l'Éducation, 2020b). Établissant de nombreux ponts entre les compétences professionnelles et le Cadre de référence de la compétence numérique, ce référentiel a établi de façon claire la volonté (politique, à tout le moins) d'exploiter le potentiel des outils numériques à l'école, notamment la programmation.

Cet ensemble de mesures successives et complémentaires démontre un mouvement, une volonté de rendre officielle l'utilisation de la programmation et du numérique en général à l'école. Il semble y avoir une démocratisation de la programmation et un consensus mondial quant à sa pertinence dans le cursus scolaire, bien que les justifications de son implantation puissent varier entre l'impératif économique (les futurs emplois) et le potentiel cognitif quant au développement et à l'apprentissage. Les apports cognitifs de la programmation ont d'ailleurs été abordés dans un rapport du Conseil supérieur de l'éducation du Québec (Couture, 2020), où la résolution de

problèmes, la collaboration et la pensée critique ont notamment été évoquées, bien que de façon nuancée³⁸.

6.2 Cadre théorique

Dans cette section nous présentons les principaux éléments, tels qu'énoncés par Dansereau et al. (1997), cités par Gohier (2018), composant notre cadre théorique, c'est-à-dire les théories et modèles qui ont orienté la recherche, les principaux concepts mobilisés, de même que les autres recherches semblables présentes dans la littérature. L'intérêt d'observer et de *comprendre* la collaboration en contexte authentique renvoie à l'approche socioconstructiviste (Lave et Wenger, 1991; Vygotsky, 1934), dont le postulat principal repose sur la prééminence des facteurs socioculturels dans l'apprentissage. Parmi les théories phares de cette approche, on compte notamment la théorie de l'activité de Leontiev (1978), qui est au cœur du modèle de l'activité d'Engeström (1987). Ce modèle décline l'activité en trois pôles : le sujet (l'individu), l'objet (l'objet de l'apprentissage) et la communauté (les pairs). La nature des interactions entre chacun de ces éléments est désignée par l'outil (numérique), la séparation du travail ainsi que les règles qui régissent la collaboration entre l'individu et les pairs (figure 14).

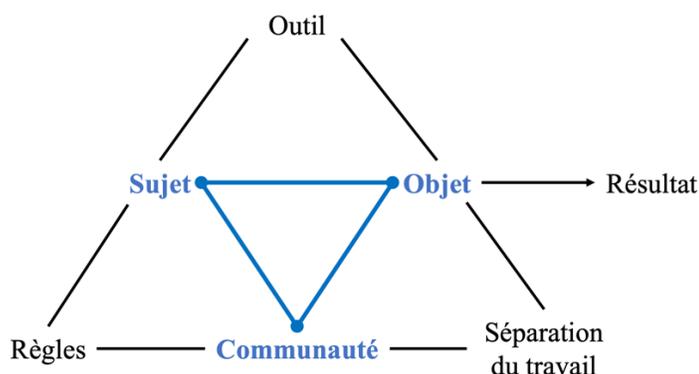


Figure 14. – Modèle de l'activité d'Engeström (1987)

³⁸ Les nuances sont apportées par la considération d'autres facteurs (outre le simple usage de la programmation) pouvant expliquer le développement de certaines habiletés, par exemple les stratégies pédagogiques des enseignants ou l'aménagement de la classe.

La théorie de l'activité offre un point de vue intéressant pour observer la collaboration entre individus médiatisée par le numérique³⁹ (Avouris et al., 2004; Hodgson et al., 2012; Hurme et Järvelä, 2005). Dans notre recherche, nous avons observé la mobilisation des compétences de résolution collaborative de problèmes (RCP), concept que nous définirons plus bas, dans un contexte d'usage pédagogique de la programmation à l'enseignement primaire. Pour définir le concept de RCP, nous débutons par la présentation de ses composantes, qui représentent des concepts existant de façon distincte. Nous faisons ici référence d'une part à la résolution de problèmes (RP), et d'autre part à la collaboration (C).

La résolution de problèmes est un processus complexe au cours duquel se succèdent plusieurs étapes. Le nombre d'étapes varie selon les auteurs, mais tous s'entendent sur les principaux fondements. Jonnassen (2014) a fondé ses travaux sur ceux de George Polya (1957), éminent chercheur dans le champ de la RP, et proposé un processus de RP en sept étapes, soit la définition du problème, l'analyse du problème, la collecte de données, l'élaboration de pistes de solution, l'évaluation de solutions alternatives, l'application des solutions et la vérification de celles-ci. Cette série d'étapes implique la mobilisation de multiples processus cognitifs, notamment à l'égard du traitement des informations présentes et de la sélection des ressources en vue de résoudre le problème, ce que Greeno et Simon (1984) appellent l'analyse moyens-finalités⁴⁰. Par ailleurs, d'autres écrits dans la littérature font référence à la résolution de problèmes complexes, qui se déroule dans un contexte authentique de la vie de tous les jours (Funke, 2010). Cette résolution complexe renvoie, selon Marquardt et Yeo (2012), à une compréhension plus approfondie du problème et à l'anticipation des problèmes subséquents potentiels induits par l'application d'une solution. Dans le cadre de notre recherche, nous mobilisons le concept de résolution de problèmes, tout en faisant appel sans la nommer à la notion de complexité, telle que la conçoivent Marquardt et Yeo. En ce qui a trait à la collaboration, il s'agit du travail collectif de deux ou plusieurs personnes en vue de réaliser une tâche (Henri et Lundgren-Cayrol, 2001).

³⁹ Nous présentons une adaptation de ce modèle en fonction de notre recherche à la section *Méthode* de ce manuscrit.

⁴⁰ Traduction libre (*means-ends analysis*).

Dans un contexte où les travaux portant sur les compétences du 21^e siècle suggèrent la juxtaposition de plusieurs compétences (Chalkiadaki, 2018; van Laar et al., 2017), on a vu apparaître un concept associant la résolution de problèmes à la collaboration, c'est-à-dire la résolution collaborative de problèmes (RCP). Comme la compétence est un savoir-agir, une démarche qui ne peut être automatisée et qui se déroule dans une situation relativement complexe (Lasnier, 2014), les compétences de résolution de problèmes et de collaboration désignent l'ensemble des habiletés mobilisées dans ce *savoir-agir* par les apprenants. À l'instar de Kamga, Romero, Komis et Misirli (2017), nous avons choisi d'étudier conjointement la résolution de problèmes et la collaboration, c'est-à-dire la résolution collaborative de problèmes (RCP). L'OCDE (2017) a défini la RCP comme étant une démarche où l'élève partage la compréhension du problème avec d'autres élèves en vue de le résoudre, et ce, en mettant en commun leurs connaissances, leurs compétences et leurs efforts. Dans le processus de RP à l'aide de pairs, les apprenants sont en mesure d'échanger leurs interprétations du problème à différents moments du processus (Hesse et al., 2015; Roschelle et Teasley, 1995), ce qui fait de chaque individu de l'équipe une ressource pour parvenir à la formulation d'une solution (DeBlois et al., 2018).

En tenant compte de la littérature, nous avons estimé que la programmation était un contexte des plus appropriés pour l'observation d'un phénomène fondé sur la résolution de problèmes et la collaboration. En effet, considérant le caractère collaboratif inhérent à l'utilisation du numérique (Depover et al., 2007), de même que le fort lien entre la programmation et la résolution de problèmes (Modeste, 2012; Wing, 2006), la programmation s'est imposée comme un contexte singulièrement pertinent à l'enseignement primaire. Son potentiel pédagogique a été explicité à de nombreuses reprises dans la littérature, notamment quant à la résolution de problèmes (Clements et Gullo, 1984; Komis et Misirli, 2011; Lai et Yang, 2011), à la collaboration (Mauch, 2001; Nugent et al., 2009; Pinto-Llorente et al., 2017), à la pensée logique (Sáez-López et al., 2016) et à la motivation (Bugmann et Karsenti, 2018a; Serafini, 2011). La complexité inhérente à l'activité de programmation, notamment lors de la manifestation de bogues dans un programme, favorise la collaboration entre pairs qui souhaitent le résoudre : les élèves doivent mettre en œuvre différentes compétences comme l'observation, l'estimation et la manipulation

(Sullivan, 2008), ou encore l'interaction sociale, c'est-à-dire l'établissement de liens avec les pairs (Mitnik et al., 2008). L'objectif de l'étude présentée dans ce manuscrit est de comprendre la mobilisation des compétences de résolution collaborative de problèmes (RCP) dans un contexte d'usage pédagogique de la programmation à l'enseignement primaire.

6.3 Méthodologie

Afin de répondre à notre objectif de recherche, nous avons mis en place une étude collective de cas (R. Stake, 1995) qui avait pour but d'observer un phénomène en contexte authentique. Les six cas retenus sont abordés dans leur complémentarité. En effet, l'analyse des résultats de ces six cas permet d'étudier en profondeur le phénomène ciblé (Pires, 1997). Nous présentons dans cette section les aspects méthodologiques ayant guidé la collecte et l'analyse des données, soient l'échantillon, la collecte des données, les instruments utilisés et les méthodes d'analyse.

6.3.1 Échantillon

Nous avons procédé à un échantillonnage intentionnel (Fortin et Gagnon, 2016) qui n'était pas guidé par des critères d'inclusion ou d'exclusion. Les six cas ont mené à la constitution de six échantillons, c'est-à-dire des groupes-classes ; LeCompte et Preissle (1993) désignent ceux-ci comme des *groupes naturels* puisqu'ils existent à l'extérieur du contexte de la recherche.

Les six cas représentent donc six classes de trois écoles d'enseignement primaire situées au Québec : deux d'entre elles sont à Montréal, dans des zones urbaines, et la troisième se situe dans la grande région de Québec, dans une zone rurale.

École	Groupe	Année	Élèves (<i>n</i>)
École A	A1	Multi	29
École B	B1	4 ^e année	14
	B2	5 ^e année	17
	B3	5 ^e année	20
	B4	6 ^e année	19
École C	C1	5 ^e année	10
			109

Tableau 14. – Échantillon de l'étude

Les participants de cette recherche (n=109) sont des apprenants des classes identifiées dans l'échantillon et dont les parents ou tuteurs ont accepté la participation.

6.3.2 Collecte des données

Tel que nous l'avons mentionné, nous avons souhaité observer la mobilisation des compétences de résolution collaborative de problèmes dans un contexte d'utilisation pédagogique de la programmation. Pour ce faire, nous avons mis en place un scénario pédagogique intitulé *Deviens un maître NAO* (Karsenti et al., 2019a). Dans ce scénario composé de 20 niveaux divisés en différentes tâches, les élèves sont appelés à connaître puis comprendre le fonctionnement d'un robot humanoïde appelé NAO. Ce dispositif robotique, mesurant environ 58 centimètres, possède des capteurs tactiles, des caméras pour la reconnaissance visuelle, de même que des micros pour la synthèse vocale. L'ensemble de ces fonctions, et bien plus, peuvent être déclenchées et contrôlées à l'aide de la programmation sur un logiciel nommé *Choregraphe*. La programmation permettant d'animer un dispositif robotique est désignée par le terme *programmation concrète* ou *tangible* (Henry et al., 2018; Komis et Misirli, 2013) et représente l'une des deux principales applications pédagogiques de la programmation à l'enseignement primaire. La seconde application pédagogique, sur laquelle repose d'ailleurs le logiciel *Choregraphe*, est la programmation visuelle (Green et Petre, 1996), c'est-à-dire une programmation qui s'effectue à l'aide de boîtes reliées par des fils plutôt que des codes textuels. La programmation visuelle et tangible est particulièrement appropriée avec des élèves du primaire, notamment pour son potentiel de développement de compétences et d'habiletés (Lai et Yang, 2011; Sáez-López et al., 2016), mais aussi pour parce qu'elle répond mieux aux besoins de cette clientèle (Xu et al., 2019).

Dans ce scénario pédagogique, chaque niveau devient en quelque sorte un espace problème (*problem space*) (Simon et Newell, 1971). En équipes de 2 à 5 individus, les élèves devaient tenter de réaliser les niveaux du scénario en utilisant différentes ressources, notamment un guide d'accompagnement sur les aspects techniques relatifs au fonctionnement du robot (Karsenti et al., 2019b). La figure 15 illustre, à l'aide du modèle adapté de l'activité d'Engeström, la dynamique inhérente aux éléments théoriques, conceptuels et méthodologiques de cette recherche.

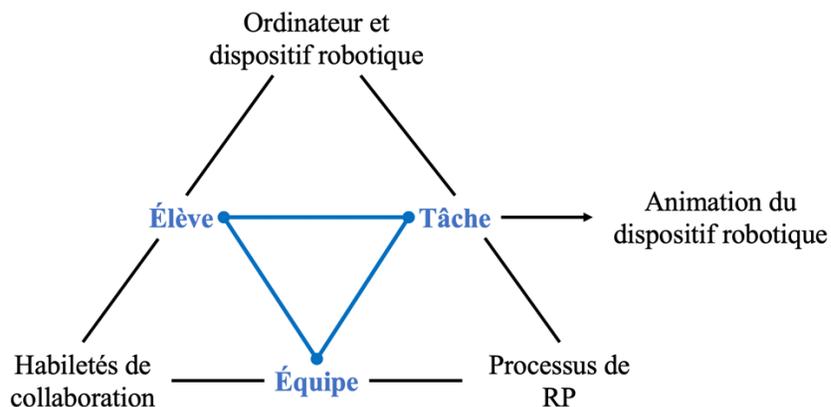


Figure 15. – Modèle de l'activité en contexte de résolution collaborative de problèmes à l'aide de la programmation tangible (adapté du modèle de l'activité d'Engeström, 1987).

6.3.3 Instruments

Afin de mener à bien notre objectif, nous avons utilisé l'observation vidéographiée (Fortin et Gagnon, 2016; Holmbeck et Shapera, 1999). Ce procédé a été guidé par notre grille d'observation de la résolution collaborative de problèmes (annexe 3). La conception de cette grille a fait l'objet d'un processus rigoureux, notamment encadré par le modèle de Strayer et Gauthier (1982). Fondée sur la littérature (Hesse et al., 2015; Kamga et al., 2017; OCDE, 2017), la grille a été mise à l'essai dans un contexte authentique de classe dans des écoles du Québec. À la suite de cette première utilisation et de calculs d'accords intercodeur satisfaisants, nous avons été en mesure de proposer une version finale de la grille, que nous utilisons ici dans le cadre de cette recherche. Présentant 15 indicateurs distribués au sein de trois dimensions, cette grille permet de déterminer la fréquence de certains comportements, de même que l'intensité (de façon qualitative).

6.3.4 Analyse des données

À l'issue des observations vidéographiées, nous avons constitué un corpus de 49 fichiers vidéo totalisant un peu plus de 900 minutes d'enregistrement. Nous avons retenu une portion de ce corpus pour l'analyse de données qui comportait des séquences filmées dans les groupes A, B1, B2, B4 et C. Cette sélection a été faite de façon à retenir un groupe de chaque école et dans le cas des groupes de l'école B, de retenir un groupe de chaque année scolaire (4^e, 5^e et 6^e année). Le

groupe B3 (5^e année) n'a donc pas fait l'objet d'une analyse considérant la quantité de données collectées et la saturation des données (Pires, 1997).

Ces fichiers ont été importés dans le logiciel NVivo (QSR International, 2020) en vue du codage des fichiers. La structure de codage utilisée provient d'une déclinaison de la grille d'observation utilisée, notamment par la codification (Paillé et Mucchielli, 2005) de chacun des 15 indicateurs ; dix indicateurs relèvent l'occurrence des éléments (pour le calcul d'une fréquence) et cinq indicateurs relèvent une interprétation empirique générale pour la séance ciblée⁴¹ (échelle à 3 niveaux, décrivant tantôt un niveau d'intensité [faible, moyen, élevé], tantôt une fréquence relative [rarement, à l'occasion, souvent]). Les indicateurs de séance relèvent d'une analyse raisonnée des données empiriques constituant notre corpus. En nous appuyant sur la grille d'observation de la RCP et sur les échelles proposées, nous avons accordé une cote d'intensité ou de fréquence à cinq indicateurs : la participation active, la persévérance, la communication, l'action en tant qu'équipe et le respect des rôles attribués. Le codage a été effectué avec la collaboration de deux étudiants de 1^{er} cycle. Avant de débiter le codage individuel, un accord intercodeur a été calculé et déterminé satisfaisant, c'est-à-dire supérieur à 75% (Cohen, 1960).

6.4 Résultats

Afin d'illustrer les données analysées, nous avons utilisé divers modes de représentation mettant en évidence les réflexions à l'origine des interprétations présentées dans la discussion des résultats (van der Maren, 2004). D'emblée, il est à noter que les résultats sont déclinés par dimension, ou encore par école et par groupe. Bien que l'analyse ait été effectuée au niveau des équipes, dans chaque groupe, nous avons choisi de présenter les résultats à un niveau de granularité moins élevé, c'est-à-dire au niveau des groupes. Dans cette section, nous présentons d'abord les résultats relatifs aux dix indicateurs d'occurrence, puis ceux rattachés aux cinq indicateurs de séance.

⁴¹ Le terme *séance* désigne la période d'intervention, sur une journée, où l'équipe de recherche était présente sur place. La durée d'une séance varie entre 60 et 120 minutes, selon l'école et le groupe.

6.4.1 Les indicateurs de fréquence

Dans un premier temps, les dix indicateurs de fréquence (ou d'occurrence) nous informent sur la manifestation de chacun des indicateurs tout en permettant d'offrir un portrait du poids de chacune des dimensions de la grille d'observation de la RCP. En moyenne, la fréquence des manifestations empiriques associées à l'établissement et au maintien d'une compréhension partagée représente 22,7% des actions analysées. Quant à la fréquence des manifestations associées à la réalisation des actions appropriées à la résolution du problème, elle représente en moyenne 56,2% des actions analysées. Puis, en ce qui concerne les manifestations relatives à l'établissement et au maintien de l'organisation de l'équipe, leur fréquence représente en moyenne 21,1% des actions analysées.

Dimension	Indicateur	Fréquence relative (par groupe)				
		A N=358	B1 N=150	B2 N=167	B4 N=261	C N=378
D1. Établissement et maintien d'une compréhension partagée	Échanger	14,2%	2,0%	21,6%	18,4%	17,5%
	Solliciter les points de vue	6,7%	3,3%	3,0%	10,3%	6,9%
	Adapter les interventions	0,8%	0%	0%	0%	0,5%
D2. Réalisation des actions appropriées à la résolution du problème	Mettre en œuvre les actions	44,1%	44%	32,3%	27,6%	38,9%
	Vérifier les actions	1,1%	0,7%	1,2%	0,4%	1,3%
	Vérifier la solution	10,9%	15,3%	13,2%	16,1%	14%
	Adapter les solutions inadéquates	3,1%	4%	2,4%	3,4%	2,9%
D3. Établissement et maintien de l'organisation de l'équipe	Attribuer des rôles	18,2%	30,7%	25,1%	21,5%	17,5%
	Identifier les forces et faiblesses	0,6%	0%	0,0%	0,8%	0,3%
	Résoudre des différends	0,3%	0%	1,2%	1,5%	0,3%
<i>Total</i>		<i>100%</i>	<i>100%</i>	<i>100%</i>	<i>100%</i>	<i>100%</i>

Tableau 15. – Fréquence relative des comportements, par groupe

Nous avons également cherché à interpréter la fréquence des indicateurs en fonction des groupes de chacune des trois écoles. Cette représentation des données offre une perspective différente par laquelle l'âge, les caractéristiques des apprenants ou le milieu pourraient être considérés davantage. Rappelons que le groupe A est composé de 29 élèves de plusieurs niveaux dans une école alternative. Le groupe B1 est quant à lui composé de 14 élèves de 4^e année. Le groupe B2,

quant à lui, est composé de 17 élèves de 5e année. Le groupe B4 est composé de 19 élèves de 6^e année et enfin, le groupe C est composé de 10 élèves de 5e année en difficultés d'apprentissage. Les données sont illustrées dans la figure 16 afin d'offrir une vue d'ensemble et de faciliter la comparaison entre les cas.

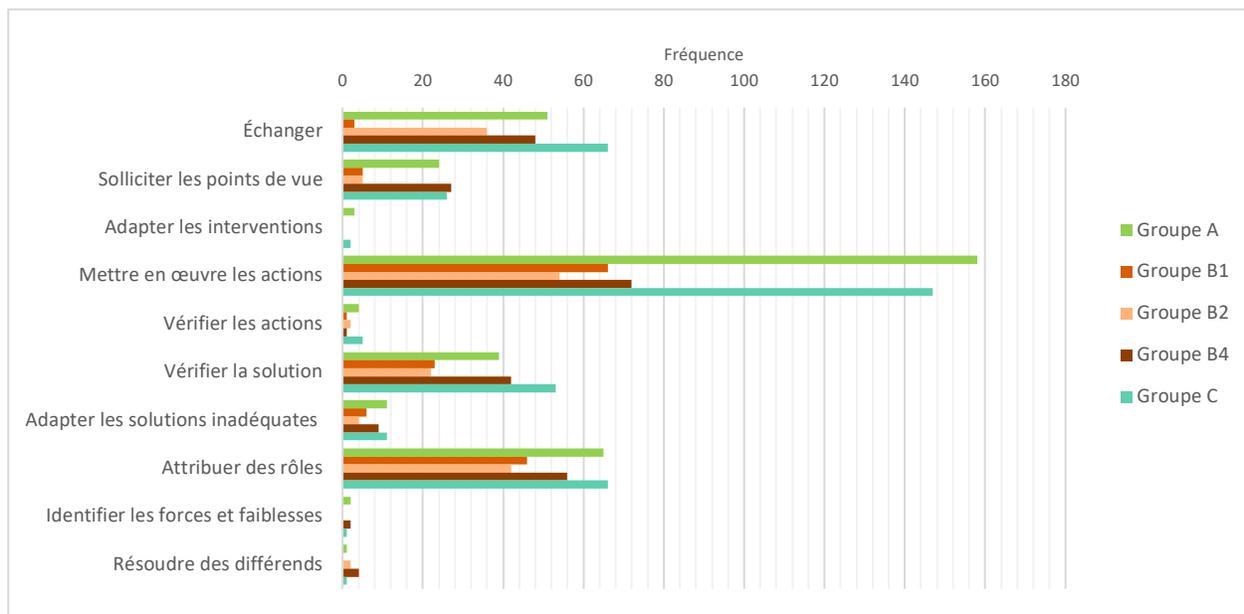


Figure 16. – Fréquence des comportements pour les trois séances, par groupe

Ces données illustrent une disparité notable entre, d'une part, les groupes A et C, et d'autre part, les groupes B1, B2 et B4 quant à l'adaptation des interventions et à la mise en œuvre des actions. Il est à noter que les indicateurs de séance ne figurent pas dans le tableau 15 et la figure 16 : ils sont présentés ci-après.

6.4.2 Les indicateurs de séance

Dans le tableau suivant, nous présentons les cotes moyennes pour chaque groupe, c'est-à-dire la moyenne des cotes attribuées aux équipes composant le groupe, au cours des trois séances d'activité.

Dimension	Indicateur de séance	Cote moyenne*				
		Groupe A	Groupe B1	Groupe B2	Groupe B4	Groupe C
D1. Compréhension partagée	N/A	-	-	-	-	-
D2. Réalisation des actions appropriées	Participer activement	2,4	2,7	2,4	2,4	2,6
	Persévérer	2,5	3,0	3,0	2,8	2,5
D3. Organisation de l'équipe	Communiquer clairement	**2,4	**2,2	**2,2	**2,5	**2,2
	Agir en tant qu'équipe	2,2	2,3	2,2	2,0	2,6
	Respecter les rôles attribués	2,4	2,5	2,2	2,5	2,5

Note. * 1 : faible, 2 : moyen, 3 : élevé (excepté l'indicateur *Communiquer clairement*)

** 1 : rarement, 2 : à l'occasion, 3 : souvent

Tableau 16. – Indicateurs de séance pour les trois séances, par groupe

Nous constatons ici des cotes relativement élevées pour la persévérance, alors que la communication claire et l'action en tant qu'équipe ont des cotes relativement basses dans la plupart des groupes. Bien que ces données ne permettent pas de tirer des conclusions définitives, elles permettent néanmoins de mieux comprendre la mobilisation de la RCP et de proposer des hypothèses appuyées sur la littérature.

6.5 Discussion

Dans l'étude de la mobilisation des compétences de RCP, il semble nécessaire d'en observer les composantes de façon à comprendre l'articulation de ces dernières. À cet égard, les données provenant des indicateurs de fréquence offrent un portrait clair : les fréquences observées à travers les équipes et les groupes sont similaires et permettent de tirer certaines conclusions.

6.5.1 L'analyse des indicateurs de fréquence

La mobilisation des habiletés associées à la dimension 2 (*réalisation des actions appropriées à la résolution du problème*) représente le cœur de l'activité selon les données. En effet, représentant en moyenne plus de 56% des habiletés observées à l'aide de la grille d'observation utilisée, cette dimension se distingue des deux autres. Ces actions associées au problème sont vraisemblablement soutenues par des habiletés liées à la compréhension partagée (dimension 1) et à l'organisation de l'équipe (dimension 3). Rappelons que les dimensions ne sont pas individuellement rattachées à l'une ou l'autre des compétences de RCP (RP ou C). Dans ce cas-ci,

la réalisation des actions appropriées (dimension 2) ne saurait être associée uniquement à la RP, puisque ces actions sont réalisées de façon collaborative. Cela démontre le caractère hautement complémentaire et interdépendant des trois dimensions de la RCP (Care et Griffin, 2017), ce qui revient en quelque sorte à dire que le tout (RCP) est plus grand que la somme des parties (RP et C). Or, certaines habiletés sont considérablement plus mobilisées que d'autres. C'est le cas d'*échanger* (dimension 1), de *mettre en œuvre les actions et vérifier la solution* (dimension 2), ainsi que d'*attribuer les rôles* (dimension 3). Ces habiletés sont discutées ci-après.

6.5.1.1 Échanger

Cet indicateur désigne les échanges à propos des caractéristiques inhérentes au problème, et ce, dans le but d'en avoir une compréhension partagée dans l'équipe. En tant que première étape du processus de résolution de problème (Jonassen, 2014; Polya, 1957) et établissant en quelque sorte la fondation de l'espace problème (Simon et Newell, 1971) au sein de l'activité de RCP, cette habileté représente une partie particulièrement importante du processus. Bien que tout aussi importante dans un processus de résolution de problème individuel, cette habileté revêt un caractère particulièrement important lorsque plusieurs individus doivent coordonner leurs actions en vue d'atteindre un but commun. L'établissement d'une compréhension partagée du problème devient alors primordial, et c'est d'ailleurs ce qui semble expliquer pourquoi cet indicateur a une fréquence d'occurrence relativement élevée.

Cela dit, lorsque l'on compare avec les deux autres indicateurs de la dimension 1, on remarque que la sollicitation des points de vue n'a pas été relevée aussi souvent. Nos observations ont permis de mettre en évidence un déséquilibre entre la capacité des apprenants à énoncer leur propre point de vue et leur capacité relativement limitée à solliciter le point de vue des autres. En effet, alors qu'en général les élèves participent aux échanges visant à établir une compréhension partagée, ils ne semblent pas percevoir l'utilité ou ressentir la nécessité (dans plusieurs cas) de questionner les autres membres quant à leur point de vue. Il en va de même pour l'indicateur *adapter les interventions*, qui n'a été que très peu observé. Ce troisième indicateur et le second, que nous avons abordé avant, relèvent d'opérations plus complexes et portées sur les autres. Comme ces élèves du niveau primaire en sont à développer leurs compétences, notamment quant à la collaboration, il nous semble normal que ces indicateurs

n'aient pas été observés davantage. Nous posons l'hypothèse que dans une situation nouvelle comme celle que nous avons induite avec l'activité de programmation, les élèves ont perçu l'utilité d'échanger et d'offrir leur point de vue, mais pour la plupart, n'ont pas pensé de façon proactive pour solliciter le point de vue des autres membres de l'équipe.

6.5.1.2 Mettre en œuvre les actions

Cet indicateur désigne les discussions entre les élèves d'une équipe lorsqu'ils cherchent à savoir comment mettre en œuvre les actions pour résoudre le problème. Contrairement à *échanger*, il s'agit ici de déterminer la mise en œuvre des actions (et non la détermination des actions en soi). En plus de représenter une part importante du processus de résolution de problème en collaboration, cet indicateur indique dans notre cas la concrétisation des actions postulées lors des échanges en une série de procédures propres à l'activité de programmation. Ce processus d'abstraction (Hromkovič et al., 2016b) consiste à comprendre comment le robot pourra être animé de façon à répondre aux critères énoncés dans le problème, tout en tenant compte des outils disponibles pour y parvenir, c'est-à-dire l'interface de programmation et les codes qu'elle est en mesure de traiter. Cette abstraction est fondamentale dans l'activité de programmation en général et démontre le caractère complexe inhérent à celle-ci : elle renvoie notamment à la définition de régularités (Wing, 2010), à la modélisation ou simulation (Voogt et al., 2015) et à la systématisation (Tatar et al., 2017). Par exemple, bien que les élèves aient compris qu'il fallait faire avancer le robot d'un mètre, ils doivent néanmoins trouver une façon de traduire cette action en une série de consignes objectives qui pourra être intégrée dans l'interface de programmation, puis interprétée par le robot.

La très faible mobilisation de l'habileté *vérifier les actions* (1,03%⁴²), qui consiste à vérifier si les actions ont été correctement effectuées, soulève un questionnement lorsque comparée à la mobilisation de l'habileté *vérifier la solution*, qui est considérablement plus présente (13,33%⁴³). D'emblée, précisons que la vérification dans notre contexte est l'action de lancer le programme pour en analyser le résultat ; ici, comme il s'agit de programmation tangible et que l'animation

⁴² Donnée moyenne pour trois séances, dans les trois écoles.

⁴³ *Ibid.*

d'un robot est le résultat escompté, les élèves peuvent faire la vérification en observant le comportement du robot et en déterminant si celui-ci correspond ou non aux critères énoncés dans l'espace problème. Nous observons que les élèves étaient davantage portés à vérifier la solution entière (*vérifier la solution*) plutôt qu'à effectuer plusieurs vérifications intermédiaires (*vérifier les actions*) avant la vérification de la solution.

6.5.1.3 Vérifier la solution

Cet indicateur désigne les actions visant à confirmer la validité du programme en le lançant sur l'interface de programmation, puis en observant le comportement du robot. La mise en œuvre de cette habileté permet aux élèves d'analyser de façon objective la validité de leur solution en se référant aux conditions et critères énoncés dans le scénario pédagogique. Quelques éléments sont à considérer dans l'interprétation des résultats présentés. En effet, nos observations ont démontré que très souvent les équipes procèdent à plusieurs vérifications pour une même solution. Deux principales raisons expliquent ce comportement : la première est fonctionnelle, c'est-à-dire que lorsque plusieurs critères ou conditions doivent être respectés, les élèves peuvent procéder à plusieurs vérifications successives en validant certains critères chaque fois. La seconde est liée au caractère ludique de l'exercice, lorsque le résultat est, par exemple, de faire danser le robot NAO. Dans ce cas, les élèves répètent la vérification par plaisir, pour la montrer à d'autres équipes, ou encore pour montrer l'animation du robot à leur enseignante. Le poids de cet indicateur dans la grille doit donc être interprété en fonction de cette réalité.

Lors de la vérification de la solution, la compréhension partagée du problème s'impose comme norme de référence commune à tous les membres de l'équipe. S'amorce alors un processus décisionnel à l'issue duquel deux avenues sont possibles : dans le premier cas, l'animation du robot démontre que la solution proposée répond aux attentes du problème, mettant ainsi fin au processus de RCP dans cet espace problème (pour ce défi). L'équipe peut alors passer au problème (défis) suivant. Dans le second cas, l'animation du robot ne rencontre pas les attentes imposées dans l'espace problème. Dans ce cas, les élèves mettent en place un processus de débogage (correspondant à l'indicateur *adapter les solutions inadéquates*), qui est en quelque sorte un second processus de résolution collaborative de problèmes subordonné au processus général de RCP associé au défi ciblé. Le débogage, qui sollicite la RCP (Nugent et al., 2009), amène

les membres de l'équipe à convenir ensemble des actions à poser pour corriger le ou les problèmes ayant émergé de la vérification de la solution. De façon itérative, ils apporteront des modifications, puis effectueront une vérification jusqu'à ce que la solution proposée réponde pas aux attentes.

Par ailleurs, pour les indicateurs de la grille d'observation, nous n'avons jamais fait allusion à la durée puisque ce n'était pas un facteur pertinent pour l'interprétation des résultats. En revanche, pour l'indicateur *adapter les solutions inadéquates* en particulier, la durée doit être considérée. Bien que mobilisée de façon moins fréquente, cette habileté représente une période définie qui débute lorsque la première vérification de la solution n'est pas concluante et qui se termine, après une ou plusieurs itérations, lorsque la solution mise en œuvre est adéquate. Ainsi, la durée de cette période est fonction de plusieurs facteurs, notamment le nombre de tentatives de débogage. La comparaison entre la fréquence de cet indicateur et les autres n'est donc pas pertinente : une analyse plus approfondie des enregistrements vidéo est nécessaire à la compréhension de cette habileté en particulier.

6.5.1.4 Attribuer les rôles

Cet indicateur désigne le mécanisme par lequel les élèves s'attribuent des rôles dans l'équipe afin de favoriser le bon déroulement du processus de résolution collaborative de problèmes. Cette attribution peut avoir lieu en amont, mais aussi pendant l'activité et peut faire l'objet de changements à différents moments. Elle peut être à la fois explicite et implicite : certains élèves peuvent décider de prendre la responsabilité et occuper une fonction précise sans que cela ait fait l'objet d'une discussion. Il peut aussi y avoir une attribution explicite, pendant laquelle les membres de l'équipe discutent et choisissent ce que chacun fera. Nos observations montrent que l'attribution est plus implicite, ou encore qu'elle est explicite mais qu'elle ne fait pas l'objet d'un consensus explicite. Par exemple, un élève affirme qu'il s'attribue un rôle et les autres élèves n'offrent aucune réponse, évitant ainsi le processus d'attribution. C'est d'ailleurs la raison pour laquelle l'indicateur *identifier les forces et les faiblesses* n'a été que très peu relevé dans les observations : ignorant souvent le processus explicite et consensuel d'attribution de rôles, les élèves n'ont pas été appelés à mobiliser cette habileté pour justifier l'attribution d'un rôle, lorsque cela est pertinent.

6.5.2 L'analyse des indicateurs de séance

Alors que les indicateurs de fréquence avaient pour fonction méthodologique de consigner un nombre d'occurrences pour chacune des habiletés observées (indicateurs), les indicateurs de séance ont quant à eux fait l'objet d'une évaluation qualitative et interprétative globale. Cette évaluation, menée lors du codage des données, avait pour but d'indiquer une interprétation appuyée d'une part sur les critères empiriques formulés dans la grille d'observation, et d'autre part, sur les échelles de Likert à 3 niveaux. À la suite de l'analyse de ces données, nous avons constaté que certains indicateurs se démarquent, c'est-à-dire *persévérer* et *agir en tant qu'équipe* (échelle d'intensité), ainsi que *communiquer clairement* (échelle de fréquence).

6.5.2.1 Communication et esprit d'équipe

Les habiletés désignées par les indicateurs *communiquer clairement* et *agir en tant qu'équipe* semblent avoir été moins mobilisées, ou de façon moins intense, que les autres. La communication claire était observée notamment par l'emploi (par les élèves) de termes précis, spécifiques ou techniques. Avec une cote moyenne de 2,33 (où 2 désigne « à l'occasion » et 3 « souvent »), il est possible de constater qu'à plusieurs moments, les élèves n'ont pas été en mesure ou n'ont pas perçu l'utilité d'utiliser des termes précis. Par ailleurs, en ce qui concerne l'indicateur *agir en tant qu'équipe*, il désigne les comportements des élèves qui démontrent, tant de façon verbale que non verbale, une cohésion et un intérêt continu au sein de l'équipe. Avec une cote moyenne de 2,28 (où 2 désigne « moyen » et 3 « élevé »), cet indicateur souligne un manque de cohésion pour plusieurs équipes. Les observations ont permis de démontrer que dans plusieurs situations, certains élèves d'une équipe se retrouvent à réaliser l'activité seuls, alors qu'un ou deux autres élèves sont désintéressés. Ce désintérêt a été observé tant dans le comportement verbal, c'est-à-dire des élèves qui affirment qu'ils n'ont pas envie de faire l'activité, que dans le comportement non verbal, notamment lorsque des élèves regardent ailleurs dans la pièce, ou en faisant dos à l'équipe.

Nous avons également observé, dans certains cas, des équipes entières qui n'étaient pas actives dans la résolution des problèmes, ce qui explique les quelques cotes « 1 » (faible) ayant été attribuées. Considérés conjointement, ces deux indicateurs appartenant à la même dimension

(organisation de l'équipe) démontrent que le travail en collaboration a représenté un défi pour plusieurs équipes.

6.5.2.2 Persévérance

Avec la cote moyenne la plus élevée (2,69), la persévérance se démarque des autres indicateurs de séance. Cet indicateur désigne la capacité de l'équipe à faire face aux obstacles et à compléter le processus de résolution collaborative de problèmes, et ce, de façon constante dans leur complétion du scénario pédagogique *Deviens un maître NAO*. Comme nous l'avons constaté dans le cadre d'une autre étude, l'utilisation du robot NAO peut s'avérer être un facteur important de motivation (Bugmann et Karsenti, 2018a). La littérature a d'ailleurs exprimé de façon assez soutenue le lien entre l'utilisation de dispositifs technologiques variés et l'augmentation ou le maintien de la motivation, plus particulièrement quant à l'utilisation de la programmation en contexte scolaire (Giroux et al., 2018; Sáez-López et al., 2016; Serafini, 2011). Nos observations ont permis de constater la curiosité et l'intérêt des élèves dès la première séance, qui se sont ensuite maintenus au fil des autres séances pour plusieurs équipes.

6.5.3 Facteurs pouvant influencer la mobilisation des compétences de RCP

Les données collectées indiquent que les compétences de RCP ont bel et bien été mobilisées par les élèves de notre échantillon. Or, notre objectif de recherche exigeait d'approfondir l'analyse de la RCP pour en comprendre la mobilisation. Ainsi, nous avons cherché à observer quels facteurs peuvent entrer en ligne de compte dans ce phénomène. L'analyse des données par groupe indique que la mobilisation de la RCP pourrait dépendre de certaines caractéristiques des apprenants. En effet, les groupes A et C se trouvent dans des conditions particulières : le groupe A se trouve dans l'école A, une école alternative où chaque groupe était composé d'élèves de différents niveaux scolaires. Le groupe C de l'école C, qui se trouve en milieu rural, est composé de très peu d'élèves en raison de difficultés d'apprentissage variées⁴⁴. À certains égards, les données de ces deux groupes se distinguent des deux autres, notamment quant à **la mise en**

⁴⁴ Au Québec, le nombre d'élèves dans un groupe-classe dépend de quelques facteurs, notamment les cotes attribuées aux élèves en situation de handicap ou ayant des difficultés d'apprentissage. Ainsi, en fonction des besoins plus ou moins substantiels, le groupe peut être réduit.

œuvre des actions et l'adaptation des interventions. Bien qu'il ne soit pas possible de tirer des conclusions définitives sur les raisons qui expliquent cette distinction, nous proposons néanmoins deux explications raisonnées.

La première est l'approche pédagogique à laquelle les élèves sont habitués. En effet, à l'école alternative A, la pédagogie par projet (PPP) occupe une place centrale dans le curriculum scolaire de l'ensemble des élèves. Ils sont donc habitués à la collaboration avec d'autres élèves et ont déjà expérimenté le processus de réalisation d'un projet, qui peut induire un certain nombre de situations où la résolution d'un problème complexe⁴⁵ est de mise. Il s'agit d'ailleurs d'une approche fortement présente dans les écoles alternatives (St-Amand Tellier, 2018). Dans le cas du groupe C, bien qu'il ne s'agisse pas d'une école alternative, l'enseignante opte pour une approche pédagogique se rapprochant de la PPP, dans la mesure où elle doit adapter le contenu et son enseignement aux besoins multiples et variés de ses élèves, et ce, en les mettant en action autour de tâches structurantes. Cette conjoncture pourrait à notre avis expliquer la disparité entre ces deux groupes et les groupes B1, B2 et B4. Notons qu'il n'est pas insinué ici que les groupes de l'école B n'ont jamais été exposés à cette pédagogie ou qu'ils ont moins développé cette habileté. Nous mettons plutôt l'accent sur le fait que les élèves des deux groupes (A et C) pourraient avoir perçu davantage l'utilité de mobiliser ces habiletés et connaissances antérieures relatives à la réalisation d'un projet.

La seconde hypothèse concerne la diversité interne des groupes. Dans le groupe A, plusieurs années séparaient parfois les élèves d'une même équipe. Il est alors inévitable que les plus vieux aient parfois à adapter les mots utilisés ou à préciser certains détails afin que les plus jeunes élèves puissent comprendre. Quant au groupe C, comme les élèves ont des besoins variés, il peut arriver qu'un élève doive adapter son intervention ou son comportement afin favoriser la participation de l'autre élève. Par exemple, nous avons observé une situation où un élève avait de la facilité avec la programmation et utilisait des termes très précis et techniques. Constatant qu'une autre élève ayant des difficultés associées au langage n'était pas en mesure de comprendre, il a utilisé des synonymes et a tenté d'adapter son discours afin qu'elle puisse

⁴⁵ Le qualificatif *complexe* est utilisé ici dans le sens proposé par Funke (2010), pour indiquer que le problème se trouve dans un contexte authentique du quotidien.

comprendre. Dans les deux cas, les élèves de ces groupes ont l'habitude d'interagir avec des pairs dont les connaissances et les besoins sont différents des leurs. Ils ont donc été appelés à mobiliser, voire à développer, cette habileté à plusieurs reprises auparavant. C'est ce qui pourrait expliquer que seuls ces deux groupes ont manifesté des comportements associés à l'indicateur *adapter les interventions*. Il s'agit là d'hypothèses que nous aurions souhaité pouvoir vérifier à l'aide d'entrevues, ce qui n'a pas été possible en raison de la pandémie de COVID-19.

6.6 Forces et limites

L'une des principales forces de cette recherche est l'échantillonnage, qui nous a permis d'observer la mobilisation des compétences de RCP dans des contextes variés. Nous avons donc pu tracer un portrait authentique et reflétant des réalités transférables dans plusieurs milieux. De plus, le corpus de données collectées s'avère aussi être une force importante de cette étude de cas multiples. En effet, considérant l'important appareillage instrumental utilisé lors des observations, nous avons été en mesure de consigner une quantité importante de données vidéo, sans toutefois que la qualité soit compromise. Il est aussi important de mentionner que ce manuscrit présente des résultats qui sont directement concernés par des débats et des réflexions dans le milieu scientifique (Stadler et al., 2020) à l'égard de la validité des tâches de RCP utilisées dans l'enquête PISA, qui sont réalisées entre un élève et un agent virtuel (H-A), plutôt qu'entre humains (H-H), comme c'est le cas dans notre étude. Nos travaux s'inscrivent donc dans la suite de ceux qui font l'objet de ces réflexions et pourront y contribuer.

Par ailleurs, deux principales limites peuvent être soulevées. Nous avons formulé à quelques reprises dans ce manuscrit des hypothèses quant à la façon d'interpréter certaines données. Il avait été prévu, lors de la planification de ce projet de recherche, de mener une série d'entrevues en aval des collectes de données sur le terrain. En recherche qualitative, le fait d'opter pour différents instruments (comme les observations et les entrevues) permet d'éviter certains biais propres à chaque instrument utilisé (Savoie-Zajc, 2018). Ces entrevues auraient été l'occasion de présenter le corpus de données vidéos aux participants et de les inviter à expliquer certaines situations ou certains comportements (Fortin et Gagnon, 2016). Ces entretiens d'autoconfrontation (Mollo et Falzon, 2004), ou *video-recall technique* (Holmbeck et Shapera,

1999; Powers et al., 1994), auraient fait émergé la perception des participants vis-à-vis de leurs comportements et aurait permis de mieux comprendre comment sont mobilisées les compétences de RCP. Nous tenons toutefois à souligner que notre méthode d'observation (qui se rapproche grandement de l'observation participante) a permis aux chercheurs d'observer *in situ* et par eux-mêmes la dynamique sur le terrain, avec les élèves. Ils ont interagi avec eux et sont en mesure de dégager des interprétations basées à la fois sur les données et sur l'expérience vécue lors des observations.

Une seconde limite de l'étude présentée dans ce manuscrit relève de la nature même de l'approche, c'est-à-dire l'étude de cas multiples. En effet, comme les observations sont effectuées dans des contextes particuliers dont les caractéristiques inhérentes sont propres aux différents cas, il peut être difficile de justifier comment les résultats présentés pourraient être appliqués à d'autres contextes. Les répercussions de cette limite ont été partiellement mitigées⁴⁶ par la variété des échantillons : une école alternative, des groupes de différents niveaux scolaires (incluant un groupe multiniveau), des établissements de milieux urbains et de milieu rural, ainsi que des élèves en situation de handicap ou aux prises avec des difficultés d'apprentissage.

6.7 Conclusion

Cette étude de cas multiples avait pour principal objectif de comprendre comment sont mobilisées les compétences de résolution collaborative de problèmes (RCP) lors d'activités de programmation en contexte scolaire, à l'enseignement primaire. À l'aide d'une grille d'observation empirique déclinant les trois grandes dimensions de la RCP, nous avons observé les comportements d'élèves de six groupes dont les caractéristiques diffèrent à différents égards (niveau, âge, besoins spéciaux, approches pédagogiques, etc.). La grande quantité de données issue de ces observations a permis de répondre de façon générale à notre objectif.

En effet, l'activité de programmation a permis aux élèves de mobiliser chacune des trois dimensions de la RCP. Cette mobilisation n'était toutefois pas uniforme à travers les dimensions. Quatre indicateurs se sont distingués : les échanges entre les membres d'une équipe en vue de

⁴⁶ Cette variété des échantillons n'a pas fait l'objet d'une analyse particulière.

comprendre le problème, la mise en œuvre des actions visant à résoudre le problème, la vérification des solutions ainsi que l'attribution des rôles. En analysant de façon comparative ces quatre indicateurs avec les autres, nous remarquons que les élèves ont démontré moins d'habiletés proactives visant à solliciter le point de vue de leurs pairs. Nous avons aussi constaté que les élèves préfèrent la vérification de la solution aux vérifications intermédiaires : ils doivent donc attendre de vérifier l'entièreté du programme avant de constater un bogue, si ce dernier n'est pas forcément apparent dans l'interface de programmation. Par ailleurs, les élèves optent souvent pour l'adoption de rôles de façon implicite, c'est-à-dire sans que l'attribution des rôles ait fait l'objet de discussions dans l'équipe et d'un consensus. Cette situation met donc en évidence l'émergence naturelle de leaders au sein de plusieurs équipes, c'est-à-dire des élèves qui s'assigneront des rôles alors que les autres membres du groupe acceptent passivement par l'absence d'opposition. Enfin, différents facteurs semblent avoir une influence sur la mobilisation de la RCP, notamment leurs expériences antérieures quant à la réalisation de projets ou à la résolution de problèmes complexes ainsi que les dynamiques propres à la classe, où les caractéristiques des autres apprenants, du groupe ou de l'équipe, modulent ou suscitent la mobilisation de certaines habiletés de RCP. Ces résultats permettent d'offrir des données empiriques intéressantes aux intervenants des milieux de pratique qui souhaiteraient développer les habiletés de RCP de leurs apprenants en utilisant la programmation informatique ou d'autres outils.

Références

Les références de cet article sont présentées à la section *Références bibliographiques* de cette thèse.

Chapitre 7 - Conclusion générale

Alors que l'utilisation pédagogique de la programmation informatique s'est démocratisée au cours des dernières années à l'échelle internationale, plusieurs plateformes et outils ont fait leur apparition dans le milieu scolaire. Au Québec, le gouvernement a démontré un intérêt certain envers l'usage de la programmation et du numérique en général à l'école; les politiques et mesures mises en place en témoignent. Or, il ne s'agit toujours pas d'un contenu officiel, c'est-à-dire qu'il ne figure pas dans le *Programme de formation de l'école québécoise* (PFÉQ). Ainsi, il revient aux enseignants de décider s'ils veulent utiliser la programmation dans leur enseignement. Or, comme la tendance l'indique, la programmation est appelée à occuper une place de plus en plus importante dans l'éducation au cours des prochaines années. Il nous est donc apparu nécessaire de mieux comprendre ce phénomène en vue d'en dégager des constats éclairants tant pour la recherche que la pratique. Dans cette conclusion, nous présenterons successivement les principaux résultats de l'étude, les pistes d'applications pratiques, ainsi que les pistes de recherches futures.

7.1 Principaux résultats

Dans cette étude de cas multiples, nous avons l'ambition de comprendre ce phénomène qu'est l'utilisation pédagogique de la programmation informatique, dont la pertinence semble indéniable. Pour ce faire, nous avons mis en place dans trois écoles du primaire au Québec un scénario pédagogique intitulé *Deviens un maître NAO* : en complétant une série de défis, les élèves étaient appelés à s'initier à la programmation en apprenant ses différents principes fondamentaux, en équipes de 2 à 5 personnes. Ces défis furent autant d'occasions de mobiliser différentes compétences, notamment la résolution de problèmes et la collaboration. À l'aide d'une méthodologie articulée autour d'un appareillage d'observation complexe, nous avons constitué un important corpus de données vidéos auprès de six groupes aux caractéristiques variées. L'analyse des données, sous la lunette de nos différentes perspectives conceptuelles et théoriques, a permis de tirer de nombreux constats.

D'abord, le premier article visait à identifier les principales pratiques effectives de programmation d'élèves du primaire en contexte scolaire (objectif spécifique 1). Il s'avère que les pratiques d'assemblage fondamentales, qui sont au cœur de l'activité de programmation, ont été les plus observées en termes de fréquence. Il est question ici de la recherche, la sélection, la liaison et le paramétrage de codes dans l'interface de programmation. Par ailleurs, lorsqu'il est question du temps imparti à chacune de ces pratiques, nous avons remarqué une prédominance du paramétrage (dont la récurrence a aussi été soulignée), de la vérification et du débogage. Il s'agit là de pratiques dans lesquelles repose la complexité inhérente à l'activité de programmation, que ce soit pour la manipulation de multiples variables (paramétrage) ou pour la résolution de problèmes dans le programme (débogage). Cette dernière pratique est d'ailleurs beaucoup plus longue qu'elle n'est fréquente : elle ne se produit pas souvent, mais lorsqu'elle est mobilisée, cette pratique nécessite beaucoup de temps aux élèves. Nos travaux ont ainsi débouché sur une typologie adaptée à l'enseignement primaire tant au niveau des pratiques que des descripteurs, qui sont basés sur nos observations (empiriques et authentiques).

Puis, en nous appuyant sur la méthode de Strayer et Gauthier (1982), nous avons répondu à notre second objectif spécifique en développant une grille d'observation de la RCP. Notre intérêt d'observer ces compétences est notamment justifié par une forte présence des compétences de résolution de problèmes et de collaboration dans la littérature associée à la programmation, et plus particulièrement la programmation tangible. La grille d'observation que nous avons proposée dans cet article s'articule autour de trois dimensions permettant de décrire l'ensemble des aspects relatifs à la RCP, notamment selon l'OCDE (2017) et Kamga et al. (2017) : l'établissement et le maintien d'une compréhension partagée, la réalisation des actions appropriées à la résolution du problème, ainsi que l'établissement et le maintien de l'organisation de l'équipe. Chaque indicateur associé à l'une de ces trois dimensions s'est vu attribuer un descripteur ainsi que des exemples (et contre-exemples) empiriques. Nous souhaitons que cette grille, à la fois ancrée dans la littérature scientifique et basée sur des observations en contexte authentique, puisse participer aux réflexions entourant la compréhension et l'observation de la RCP, qui fait l'objet de débats dans la littérature au moment d'écrire ces lignes (Stadler et al., 2020).

Enfin, en utilisant cette grille d'observation inédite, nous avons offert un portrait détaillé et compréhensif de la mobilisation de la RCP chez des élèves du primaire dans le cadre d'une activité de programmation (objectif spécifique 3). S'éloignant de l'approche quantitative employée dans l'étude PISA 2015, notre approche s'inscrivait véritablement dans un processus interprétatif. Les résultats ont démontré que la mobilisation de la RCP s'effectue dans les trois dimensions de la grille, soulignant toutefois une prédominance de la seconde, c'est-à-dire la réalisation d'actions en vue de résoudre le problème. Les comportements des élèves montrent qu'ils sont souvent capables de formuler leur propre point de vue afin d'établir une compréhension partagée du problème, mais qu'il est moins fréquent de les voir solliciter le point de vue des autres membres de l'équipe. Nous avons également remarqué que l'assignation des rôles au sein de l'équipe se fait surtout de façon implicite : des élèves prennent donc un rôle de leadership, alors que les autres ne signifient ni leur accord ni leur désaccord. L'attribution des rôles ne fait que très rarement l'objet de discussions et d'un consensus. De façon globale, dans les séances, nous avons observé une communication et un esprit d'équipe relativement faibles comparés à d'autres indicateurs généraux. Par ailleurs, nous avons également remarqué que les élèves n'effectuent pratiquement pas de vérifications intermédiaires lors de la programmation, attendant ainsi d'avoir terminé le programme avant de vérifier si toutes les conditions ont été respectées. Finalement, la persévérance des élèves a été globalement observée, ce que nous expliquons notamment par l'activité de programmation en soi et l'utilisation d'un robot humanoïde.

Nous proposons ci-après un digramme synthétisant l'ensemble des éléments composant cette recherche, organisé en fonction de nos trois objectifs spécifiques de recherche (figure 17).

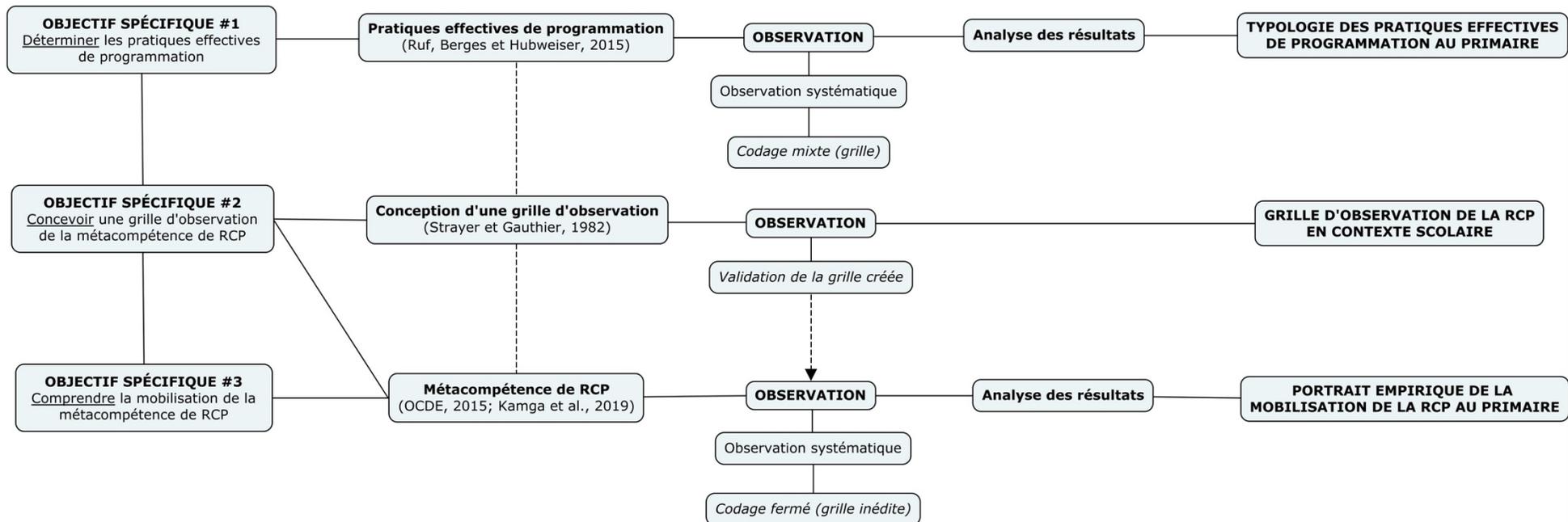


Figure 17. – Diagramme synthèse de la recherche

7.2 Pistes d'applications pratiques et de recherches futures

Les décisions en faveur de l'usage de la programmation à l'école de nombreux gouvernements à travers le monde montrent qu'il s'agit d'un phénomène qui dépasse largement les frontières. Au Québec, les politiques et mesures annoncées démontrent une volonté politique de développer l'utilisation de la programmation à l'école à des fins pédagogiques. Il semble donc que la conjoncture actuelle soit des plus propices à l'intégration officielle de la programmation au curriculum du primaire et du secondaire. Cela aurait pour effet de favoriser le développement d'une multitude d'activités directement dans les milieux de pratique par les différents intervenants (enseignants, conseillers pédagogiques, etc.), et se faisant, aurait aussi pour effet de générer des contextes et des phénomènes qui pourront être observés et analysés dans le cadre d'études scientifiques.

7.2.1 Dans les milieux de pratique

Comme l'utilisation de la programmation n'est toujours pas encadrée par le PFÉQ, l'ajout de cet outil aux pratiques pédagogiques relève de l'autonomie professionnelle des enseignants, qui peuvent choisir ou non d'en faire profiter leurs élèves. En publiant la plus récente version du Référentiel des compétences professionnelles de la profession enseignante, le ministère de l'Éducation (2020) a pris le soin d'ajouter la compétence numérique à l'ensemble des compétences essentielles à l'enseignement. Il nous semble clair que les enseignants auront à acquérir ou développer, dès la formation initiale et tout au long de leur carrière, certaines habiletés ou connaissances associées à la programmation informatique, que ce soit à la formation initiale ou dans leur formation continue. Or, outre l'impératif professionnel, il est important de rappeler que l'attitude des enseignants à l'égard de la programmation a un effet direct sur l'utilisation pédagogique qu'ils en feront (Farjon et al., 2018; Julio et al., 2018). Il s'avère donc important qu'ils puissent juger du potentiel de cet outil d'apprentissage au-delà du caractère ludique de plusieurs applications ou dispositifs.

La typologie des pratiques effectives de programmation visuelle peut avoir un rôle d'importance dans le milieu de l'éducation. En effet, son utilisation par les enseignants ou les conseillers pédagogiques pourrait permettre de créer des activités ciblant différentes pratiques de

programmation visuelle, de façon à diversifier l'activité et les façons d'apprendre les concepts fondamentaux de la programmation. Cela permettrait également aux élèves de mobiliser, voir développer, un ensemble de pratiques de programmation visuelle pouvant être réinvesties dans d'autres contextes, voire même ultérieurement lors d'activités de programmation textuelle. La typologie est donc un outil structurant pour le développement et l'animation de situations d'apprentissage au primaire, ou même au secondaire.

De plus, afin de favoriser la collaboration entre les chercheurs et les praticiens, nous soulignons la pertinence d'adapter la grille d'observation empirique de la RCP (termes utilisés, structure) aux programmes de formation ciblés du système éducatif souhaité. Au Québec, par exemple, l'établissement d'une adéquation entre les termes employés dans le PFÉQ et dans la grille constitue une étape préliminaire à l'utilisation de la grille dans les milieux de pratique.

De façon globale, les résultats de cette recherche contribuent aux réflexions entourant l'implantation officielle de la programmation dans le curriculum scolaire québécois. Tel que nous l'indiquent les politiques et mesures mises en place au cours des deux dernières années, tout porte à croire que la programmation sera tôt ou tard intégrée au PFÉQ. D'ailleurs, la toute nouvelle version du *Référentiel de compétences professionnelles de la profession enseignante* (Ministère de l'Éducation, 2020b) met de l'avant l'importance du numérique et, par le fait même, de la programmation; la prise en compte du *Cadre de référence de la compétence numérique* (Ministère de l'Éducation et de l'Enseignement supérieur, 2019) et du guide *L'usage pédagogique de la programmation informatique* (Ministère de l'Éducation, 2020a) offre une vue d'ensemble évocatrice à cet égard.

7.2.2 Dans le milieu de la recherche

Dans l'ensemble, nos travaux ouvrent la voie à de nombreuses recherches ultérieures relatives à l'usage pédagogique de la programmation informatique. D'emblée, les observations que nous avons menées gagneraient à être complétées par des entretiens d'autoconfrontation (Mollo et Falzon, 2004) avec les participants. Le fait de mener ces entretiens pourrait rendre compte des perceptions des élèves et ainsi apporter une perspective nouvelle aux données d'observation. Il serait également pertinent d'utiliser la typologie des pratiques de programmation ainsi que la

grille d'observation de la RCP que nous avons proposées dans différents contextes. La première itération de la grille devrait être réutilisée et bonifiée à la lumière des contextes et des milieux dans lesquels elle pourrait être mise en œuvre, et ce, dans le cadre d'études empiriques dans les écoles du primaire ou du secondaire.

7.2.3 Recommandation

À la lumière des résultats présentés dans cette thèse, nous formulons une recommandation en lien avec l'usage pédagogique de la programmation informatique. Comme ce sont les enseignants qui sont appelés à trouver diverses façons d'en faire usage, il est important de considérer certains aspects importants. Nous en ciblons deux en particulier : la sollicitation de nombreuses compétences ou connaissances et la mise en place de problèmes complexes et authentiques. D'abord, il est important de favoriser l'interdisciplinarité lors des activités de programmation. En effet, en ayant en tête la double finalité de la programmation (à la fois objet et outil d'apprentissage), il est important de ne pas limiter son potentiel à une simple activité visant le développement d'habiletés technologiques. La mobilisation de concepts plus ou moins complexes et appartenant à diverses disciplines fera en sorte de favoriser le développement de compétences variées. Puis, il est également nécessaire d'inscrire la programmation dans des contextes authentiques de résolution de problèmes complexes. La complexité se trouve notamment dans la prise en compte de nombreuses variables, ce qui n'est pas possible lorsque le problème proposé est trop fermé et laisse peu de place à la créativité des élèves. La complexité inhérente aux problèmes proposés aura pour effet de favoriser la collaboration au sein des équipes, qui devront mettre en commun les capacités individuelles de chaque membre pour compléter l'activité.

Références bibliographiques

- Académie des sciences. (2013, mai). *L'enseignement de l'informatique en France: il est urgent de ne plus attendre*. http://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf
- Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832-835. <https://doi.org/10.1093/comjnl/bxs074>
- Aldebaran Robotics. (2014). Choregraphe (version 2.1.4) [logiciel]. Softbank Group.
- Alexandrova, S., Tatlock, Z. et Cakmak, M. (2015, mars). Visual Robot Programming for Generalizable Mobile Manipulation Tasks. Dans. à 10th Annual ACM/IEEE International Conference on Human-Robot Interaction, Portland, États-Unis.
- Allaire, S. (2017). L'interdiction avant l'éducation. *LeQuotidien*. <https://www.lequotidien.com/chroniques/linterdiction-avant-leducation-26843340a213726f033f795f7899f666>
- Ananiadou, K. et Claro, M. (2009, décembre). 21st century skills and competences for new millennium learners in OECD countries. *OECD Education Working Papers*, (41). <https://doi.org/http://dx.doi.org/10.1787/218525261154>
- Armoni, M., Meerbaum-Salant, O. et Ben-Ari, M. (2015). From Scratch to «Real» Programming. *ACM Transactions on Computing Education*, 14(4). <https://doi.org/10.1145/2677087>
- Arts, J. A., Gijssels, W. H. et Segers, M. S. (2006). Enhancing problem-solving expertise by means of an authentic, collaborative, computer supported and problem-based course. *European Journal of Psychology of Education*, 21(1), 71-90.
- Atmatzidou, S. et Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670. <https://doi.org/10.1016/j.robot.2015.10.008>
- Australian Government. (2020). *Support for Science, Technology, Engineering and Mathematics (STEM)*. <https://www.education.gov.au/support-science-technology-engineering-and-mathematics>
- Avouris, N., Komis, V., Fiotakis, G., Dimitracopoulou, A. et Margaritis, M. (2004, 2-3 septembre). Method and Tools for analysis of collaborative problem-solving activities. Dans. à First International Workshop on Activity Theory Based Practical Methods for IT-Design, Copenhagen, Danemark.
- Balanskat, A. et Engelhardt, K. (2014). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. European Schoolnet. http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887
- Barma, S. (2018). *Réaliser une étude de cas multiple qui vise à affiner les connaissances sur l'usage pédagogique ou didactique de la programmation dans les écoles du Québec* (publication n° 118982). C. d. r. e. d. i. s. l. r. scolaire. <https://lel.crires.ulaval.ca/oeuvre/rapport-final-realiser-une-etude-de-cas-multiple-qui-vise-affiner-les-connaissances-sur>
- Baron, G.-L. et Bruillard, E. (2001). Une didactique de l'informatique? *Revue française de pédagogie*, (135), 163-172.

- Baron, G.-L. et Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française? Mise en perspective historique. *Revue française de pédagogie. Recherches en éducation*, (195), 51-62. <https://doi.org/https://doi.org/10.4000/rfp.5032>
- Barr, V. et Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads*, 2(1), 48-54.
- Bélangier, M.-F. (2016, 18 septembre). La programmation informatique à l'école: un nouvel outil pédagogique en vogue. *Radio-Canada.ca*. <http://ici.radio-canada.ca/nouvelle/803708/programmation-informatique-codage-ecole-outil-pedagogique>
- Bers, M. U., Flannery, L., Kazakoff, E. R. et Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Béziat, J. (2012). *Les TIC à l'école primaire en France: informatique et programmation*. Colloque scientifique international sur les TIC en éducation: bilan, enjeux et perspectives futures, Montréal.
- Blackwell, A. F. (2002). What is Programming? Dans. au 14th Workshop of the Psychology of Programming Interest Group, Londres, Angleterre. <http://www.ppig.org/workshops/ppig-2002-14th-annual-workshop>
- Blanquer, J.-M. (2018). *Interdiction du téléphone portable dans les écoles et les collèges*. <http://www.education.gouv.fr/cid133479/interdiction-telephone-portable-dans-les-ecoles-les-colleges.html>
- Bossuet, G. (1982). *L'ordinateur à l'école*. Presses Universitaires de France.
- Bossuet, G. (1983). *L'ordinateur à l'école: le système LOGO* (2^e éd.). Presses Universitaires de France.
- Boutin, G. (2004). L'approche par compétences en éducation: un amalgame paradigmatique. *Connexions*, 1(81), 25-41. <https://doi.org/10.3917/cnx.081.0025>
- Bower, M. (2008, juin). A taxonomy of task types in computing. Dans. à 13th annual conference on Innovation and technology in computer science education, Madrid, Espagne.
- Boyer, P. et Martineau, S. (2018). La problématique. Dans T. Karsenti et L. Savoie-Zajc (dir.), *La recherche en éducation: étapes et approches* (4^e éd.). Presses de l'Université de Montréal.
- Brennan, K. et Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Dans. à American Educational Research Association meeting, Vancouver, Canada. https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
- Brigaudeau, C. (2018). Interdire le portable à l'école? Bon courage! *Le Parisien*. <http://www.leparisien.fr/societe/interdire-le-portable-a-l-ecole-bon-courage-06-06-2018-7757118.php>
- Bruner, J. S. (1967). On Cognitive Growth. Dans J. S. Bruner, R. R. Olver et P. M. Greenfield (dir.), *Studies in Cognitive Growth* (p. 1-29). John Wiley & Sons, Inc.
- Bugmann, J. et Karsenti, T. (2018a). Apprendre à programmer un robot humanoïde: impacts sur des élèves de l'adaptation scolaire. *Formation et Profession*, 26(1), 26-42. <https://doi.org/10.18162/fp.2018.460>
- Bugmann, J. et Karsenti, T. (2018b). Quand les robots entrent en classe. *Formation et Profession*, 26(1), 142-145. <https://doi.org/10.18162/fp.2018.a141>

- Bugmann, J., Karsenti, T. et Parent, S. (2018, mai). Développer l'apprentissage de la programmation par l'usage d'un jeu vidéo : intégration de Scratch à Minecraft. Dans. Colloque international en éducation, Montréal, Canada.
- Cahour, B. et Licoppe, C. (2010). Confrontations with traces of one's own activity. Understanding, development and regulation of action in an increasingly reflexive world. *Revue d'anthropologie des connaissances*, 4(2), A-K. <https://doi.org/https://doi.org/10.3917/rac.010.000a>
- Calmet, C., Hirtzig, M. et Wilgenbus, D. (2016). *1, 2, 3... codez! : enseigner l'informatique à l'école et au collègue (cycles 1, 2 et 3)*. Paris, France : Éditions Le Pommier.
- Canada Learning Code. (2020). *Apprendre pour un monde numérique: un cadre de référence pancanadien pour l'enseignement de l'informatique*. https://k12csframework.ca/wp-content/uploads/apprendre_pour_un_monde_numerique_final.pdf
- Care, E. et Griffin, P. (2017). Assessment of collaborative problem-solving processes. Dans B. Csapó et J. Funke (dir.), *The nature of Problem Solving. Using Research to Inspire 21st Century Learning* (p. 227-243). OECD Publishing. <https://doi.org/10.1787/9789264273955-16-en>
- Caspersen, M. E. et Nowack, P. (2013). Computational thinking and practice: A generic approach to computing in Danish high schools. Dans. à 15th Australasian Computing Education Conference, Adelaide, Australie.
- Centre facilitant la recherche et l'innovation dans les organisations. (2020). *Portrait numérique des foyers québécois*. https://cefrio.qc.ca/media/2288/netendances-2019_fascicule-4_portrait-numerique-des-foyers-quebecois_final.pdf
- Cesar, M., Perret-Clermont, A.-N. et Benavente, A. (2000). Modalités de travail en dyades et conduites à des tâches d'algèbre chez des élèves portugais. *Revue suisse des sciences de l'éducation*, 3, 443-466.
- Chalkiadaki, A. (2018). A Systematic Literature Review of 21st Century Skills and Competencies in Primary Education. *International Journal of Instruction*, 11(3), 1-16.
- Champoux, L., Couture, C. et Royer, E. (1992). *École et comportement. L'observation systématique du comportement*. Ministère de l'Éducation.
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X. et Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175. <https://doi.org/https://doi.org/10.1016/j.compedu.2017.03.001>
- Ching, Y.-H., Hsu, Y.-C. et Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends*, 62(6), 563-573.
- Clements, D. H. et Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051.
- Code.org. (2016). *All Hour of Code events - Canada*. <https://hourofcode.com/us/events/all/ca>
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1), 37-46.
- Collin, S. et Brotcorne, P. (2019). Capturing digital (in) equity in teaching and learning: A sociocritical approach. *The International Journal of Information Learning Technology*, 36(2). <https://doi.org/https://doi.org/10.1108/IJILT-05-2018-0059>
- Computer Science Education Research Group. (2018). *CSUnplugged*. <https://csunplugged.org/en/>

- Conde, M. Á., Fernández-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellán-Olivera, V. et García-Peñalvo, F. J. (2017, octobre). Promoting Computational Thinking in K-12 students by applying unplugged methods and robotics. Dans. à 5th International Conference on Technological Ecosystems for Enhancing Multiculturality, Cadiz, Espagne.
- Conseil des compétences futures. (2020). *Le Canada – Une nation axée sur l'apprentissage : Une main-d'œuvre qualifiée et souple, prête à définir l'avenir*. (publication n° Em16-20/2020E-PDF).
- Couture, H. (2020). *État des connaissances sur l'apprentissage et la pratique de la programmation informatique en contexte scolaire*.
- Crahay, M. (1987). Logo, un environnement propice à la pensée procédurale. *Revue française de pédagogie*, 80(1), 37-56. <https://doi.org/https://doi.org/10.3406/rfp.1987.1473>
- Curzon, P. et McOwan, P. W. (2017). *The power of computational thinking : games, magic and puzzles to help you become a computational thinker*. World Scientific.
- De-Sola Gutiérrez, J., Rodríguez de Fonseca, F. et Rubio, G. (2016, 2016-October-24). Cell-Phone Addiction: A Review [Review]. *Frontiers in Psychiatry*, 7(175). <https://doi.org/10.3389/fpsyt.2016.00175>
- DeBlois, L., Romero, M. et Abajyan, P. (2018). Créacube, comparaison de la résolution créative de problèmes, chez des enfants et des adultes, par le biais d'une tâche de robotique modulaire. *MathémaTICE*, (61). <http://revue.sesamath.net/spip.php?article1104>
- Delcker, J. et Ifenthaler, D. (2017). Computational Thinking as an Interdisciplinary Approach to Computer Science School Curricula: A German Perspective. Dans P. J. Rich et C. B. Hodges (dir.), *Emerging Research, Practice and Policy on Computational Thinking* (p. 49-62). Springer.
- Departement for Education. (2013). *National curriculum in England: computing programmes of study*. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>
- Depover, C., Karsenti, T. et Komis, V. (2007). *Enseigner avec les technologies: Favoriser les apprentissages, développer des compétences*. Presses de l'Université du Québec.
- Depover, C. et Noël, B. (2003). *Les TIC peuvent-elles favoriser le développement de compétences de haut niveau?*
- Deslauriers, J.-P. et Kérisit, M. (1997). Le devis de recherche qualitative. Dans J. Poupart, L.-H. Groulx, J.-P. Deslauriers, A. Laperrière, R. Mayer et A. P. Pires (dir.), *La recherche qualitative: enjeux épistémologiques et méthodologiques* (p. 85-109). Gaëtan Morin.
- Dillenbourg, P., Järvelä, S. et Fischer, F. (2009). The Evolution of Research on Computer-Supported Collaborative Learning: From Design to Orchestration. Dans N. Balacheff, S. Ludvigsen, T. de Jong, A. Lazonder et S. Barnes (dir.), *Technology-Enhanced Learning: Principles and Products* (p. 3-19). Springer.
- Dillenbourg, P. et Traum, D. (2006). Sharing Solutions: Persistence and Grounding in Multimodal Collaborative Problem Solving. *Journal of the Learning Sciences*, 15(1), 121-151. https://doi.org/http://dx.doi.org/10.1207/s15327809jls1501_9
- Dionne, L. (2018). L'analyse qualitative des données. Dans T. Karsenti et L. Savoie-Zajc (dir.), *La recherche en éducation: étapes et approches*. (4^e éd., p. 317-342). Les Presses de l'Université de Montréal.

- Duchâteau, C. (1989). *Quand le savoir faire ne suffit plus*.
- Engeström, Y. (1987). *Learning by expanding*. Orienta-konsultit.
- Fadel, C. et Trilling, B. (2009). *21st Century Skills: Learning for Life in Our Times*. Jossey-Bass.
- Farjon, D., Smits, A. et Voogt, J. (2018, 2018/11/30/). Technology integration of pre-service teachers explained by attitudes and beliefs, competency, access, and experience. *Computers & Education*. [https://doi.org/https://doi.org/10.1016/j.compedu.2018.11.010](https://doi.org/10.1016/j.compedu.2018.11.010)
- Feltovich, P. J., Spiro, R. J., Coulson, R. L. et Feltovich, J. (1996). Collaboration Within and Among Minds: Mastering Complexity, Individually and in Groups. Dans T. Koschmann (dir.), *CSCL: Theory and practice of an emerging paradigm* (p. 25-44). Lawrence Erlbaum Associates Inc.
- Fessakis, G., Gouli, E. et Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97.
- Feurzeig, W., Papert, S., Bloom, M., Grant, R. et Solomon, C. (1969). *Programming-languages as a conceptual framework for teaching mathematics*. B. a. N. I. Bolt.
- Fondation de la Commission scolaire de Montréal. (2020). *Code MTL*. Fondation de la Commission scolaire de Montréal,. <https://codemtl.org>
- Fortin, F. et Gagnon, J. (2016). *Fondements et étapes du processus de recherche : méthodes quantitatives et qualitatives* (3^e éd.). Chenelière éducation.
- Forum économique mondial. (2015). *New Vision for Education. Unlocking the Potential of Technology*. Forum économique mondial.
- Forum économique mondial. (2016, 14 janvier). The Fourth Industrial Revolution: what it means, how to respond. *Forum économique mondial*. <https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>
- Forum économique mondial. (2018). *Towards a Reskilling Revolution: A Future of Jobs for All*. http://www3.weforum.org/docs/WEF_FOW_Reskilling_Revolution.pdf
- Fourez, G. (2005). Controverses autour de diverses conceptualisations (modélisations) des compétences transversales. *Canadian Journal of Math, Science & Technology Education*, 5(3), 401-412.
- Funke, J. (2010, May 01). Complex problem solving: a case for complex cognition? [journal article]. *Cognitive Processing*, 11(2), 133-142. <https://doi.org/10.1007/s10339-009-0345-0>
- Futschek, G. et Moschitz, J. (2011, octobre). Learning algorithmic thinking with tangible objects eases transition to computer programming. Dans. International Conference on Informatics in Schools: Contributing to 21st Century Education, Bratislava, Slovaquie.
- Gardner, F. (2000). Methodological issues in the direct observation of parent–child interaction: Do observational findings reflect the natural behavior of participants? *Clinical Child and Family Psychology Review*, 3(3), 185-198.
- Gaspar, M., Julião, J. et Cruz, M. (2018, Juin). Organizational Strategies Induced by the Fourth Industrial Revolution: Workforce Awareness and Realignment. Dans. à International Conference on Innovation, Engineering and Entrepreneurship, Guimarães, Portugal.
- George, S. (2001). *Apprentissage collectif à distance, SPLACH: un environnement informatique support d'une pédagogie de projet* [Université du Maine]. <https://tel.archives-ouvertes.fr/edutice-00000207/document>

- Giroux, P., Girard, J.-F. et Gagnon, M. (2018). La robotique pour motiver ses élèves. *Revue hybride de l'éducation*, 2(2), 68-77.
- Gohier, C. (2018). Le cadre théorique. Dans T. Karsenti et L. Savoie-Zajc (dir.), *La recherche en éducation: étapes et approches*. (4^e éd.). Les Presses de l'Université de Montréal.
- Goldman, S. V. (1996). Mediating Microworlds: Collaboration on High School Science Activities. Dans T. Koschmann (dir.), *CSCL: Theory and practice of an emerging paradigm* (p. 45-81). Lawrence Erlbaum Associates Inc.
- Gorman, H. et Bourne, L. E. (1983, 1983/03/01). Learning to think by learning LOGO: Rule learning in third-grade computer programmers. *Bulletin of the Psychonomic Society*, 21(3), 165-167. <https://doi.org/10.3758/BF03334676>
- Gouvernement de l'Ontario. (2020). *Nouveau programme-cadre de mathématiques de la 1re à la 8e année*. Gouvernement de l'Ontario. <https://www.ontario.ca/fr/page/nouveau-programme-cadre-de-mathematiques-de-la-1re-la-8e-annee>
- Gouvernement de la Colombie-Britannique. (2016). *Introduction au programme Conception, compétences pratiques et technologies*. <https://curriculum.gov.bc.ca/fr/curriculum/adst/introduction>
- Gouvernement du Canada. (2020). *Conseil des compétences futures*. Gouvernement du Canada. <https://www.canada.ca/fr/emploi-developpement-social/programmes/competences-futures/conseil.html>
- Gouvernement du Québec. (2006). *Programme de formation de l'école québécoise - Éducation préscolaire et enseignement primaire*.
- Gouvernement du Québec. (2018). *Plan d'action numérique en éducation et en enseignement supérieur*. http://www.education.gouv.qc.ca/fileadmin/site_web/documents/ministere/PAN_Plan_action_VF.pdf
- Government of the United Kingdom. (2017). *The national curriculum*. <https://www.gov.uk/national-curriculum/overview>
- Green, T. R. G. et Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing*, 7(2), 131-174.
- Greeno, J. G. et Simon, H. A. (1984). *Problem Solving and Reasoning*. <http://www.dtic.mil/dtic/tr/fulltext/u2/a138889.pdf>
- Greiff, S., Holt, D. et Funke, J. (2013). Perspectives on problem solving in cognitive research and educational assessment: analytical, interactive, and collaborative problem solving. *Journal of Problem Solving*, 5(2), 71-91. <http://publications.uni.lu/bitstream/10993/3177/1/Greiff%20Holt%20Funke%202013%20JoPS.pdf>
- Griffin, P. et Care, E. (2015). The ATC21S Method. Dans P. Griffin et E. Care (dir.), *Assessment and Teaching of 21st Century Skills: Methods and Approach* (p. 3-33). Springer.
- Grover, S. et Pea, R. (2013). Computational Thinking in K-12 A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43.
- Guikas, I., Morin, D. et Bigras, M. (2016). Développement d'une grille d'observation: considérations théoriques et méthodologiques. *Revue francophone de la déficience intellectuelle*, 27, 163-178. <https://doi.org/10.7202/1043131.ar>

- Guttag, J. V. (2017). *Introduction to Computation and Programming Using Python: With Application to Understanding Data* (2^e éd.). The MIT Press.
- Hanko, G. (2016). *Increasing competence through collaborative problem solving* (vol.). Routledge.
- Hao, J., Liu, L., von Davier, A. et Kyllonen, P. (2015). Assessing collaborative problem solving with simulation based tasks. Dans. à *Exploring the Material Conditions of Learning: The Computer Supported Collaborative Learning (CSCL)*, Gothenburg, Suède.
- Hardouin-Mercier, G. (1974). *Technique de la programmation*. Masson.
- Hargittai, E. (2010). Digital natives? Variation in internet skills and uses among members of the “net generation”. *Sociological inquiry*, 80(1), 92-113.
- Harris, F. C. et Lahey, B. B. (1982). Recording system bias in direct observational methodology: A review and critical analysis of factors causing inaccurate coding behavior. *Clinical Psychology Review*, 2(4), 539-556.
- Heitink, M., Voogt, J., Verplanken, L., van Braak, J. et Fisser, P. (2016, 2016/10/01/). Teachers' professional reasoning about their pedagogical use of technology. *Computers & Education*, 101, 70-83. <https://doi.org/https://doi.org/10.1016/j.compedu.2016.05.009>
- Hendrix, R. et Weeks, M. (2018). *First Programming Language for High School Students*. Society for Information Technology & Teacher Education International Conference 2018, Washington, D.C., États-Unis. <https://www.learntechlib.org/p/182788>
- Henri, F. (2014). *Les bases de la programmation*. JFD éditions.
- Henri, F. et Lundgren-Cayrol, K. (2001). *Apprentissage collaboratif à distance: Pour comprendre et concevoir les environnements d'apprentissage virtuels*. Presses de l'Université du Québec.
- Henry, J., Dumas, B. et Bodart, A. (2018, octobre). Programmation tangible pour les enfants: analyse de l'existant, classification et opportunités. Dans. à la 30e conférence francophone sur l'interaction homme-machine, Brest, France.
- Hesse, F., Care, E., Buder, J., Sassenberg, K. et Griffin, P. (2015). A Framework for Teachable Collaborative Problem Solving Skills. Dans P. Griffin et E. Care (dir.), *Assessment and Teaching of 21st Century Skills: Methods and Approach* (p. 37-56). Springer. https://doi.org/10.1007/978-94-017-9395-7_2
- HITSA. (2017). *ProgeTiigri Kogumik*. [http://progetiiger.ee/?q=&domain\[\]=1](http://progetiiger.ee/?q=&domain[]=1)
- Hodgson, V., McConnell, D. et Dirckinck-Holmfeld, L. (2012). The Theory, Practice and Pedagogy of Networked Learning. Dans L. Dirckinck-Holmfeld, V. Hodgson et D. McConnell (dir.), *Exploring the Theory, Pedagogy and Practice of Networked Learning* (p. 291-305). Springer New York. https://doi.org/10.1007/978-1-4614-0496-5_17
- Holmbeck, G. N. et Shapera, W. E. (1999). Research methods with adolescents. Dans P. C. Kendall, J. N. Butcher et G. N. Holmbeck (dir.), *Handbook of research methods in clinical psychology* (2^e éd., p. 634-661). John Wiley & Sons, Inc.
- Holzman, T. G. et Glaser, R. (1977). Developing Computer Literacy in Children: Some Observations and Suggestions. *Educational Technology*, 17(8), 5-11. www.jstor.org/stable/44421200
- Hoos, H. H. (2012). Programming by optimization. *Communications of the ACM*, 55(2), 70–80. <https://doi.org/10.1145/2076450.2076469>

- Horn, M. S., Solovey, E. T., Crouser, R. J. et Jacob, R. J. K. (2009). *Comparing the use of tangible and graphical programming languages for informal science education*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, États-Unis.
- Hromkovic, J. (2006, novembre). Contributing to general education by teaching informatics. Dans. à International Conference on Informatics in Secondary Schools-Evolution and Perspectives (ISSEP), Vilnius, Lithuanie.
- Hromkovič, J., Kohn, T., Komm, D. et Serafini, G. (2016a). Combining the Power of Python with the Simplicity of Logo for a Sustainable Computer Science Education. Dans A. Brodnik et F. Tort (dir.), *Informatics in Schools: Improvement of Informatics Knowledge and Perception: 9th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2016, Münster, Germany, October 13-15, 2016, Proceedings* (p. 155-166). Springer International Publishing. https://doi.org/10.1007/978-3-319-46747-4_13
- Hromkovič, J., Kohn, T., Komm, D. et Serafini, G. (2016b). Examples of algorithmic thinking in programming education. *Olympiads in Informatics*, 10(1-2), 111-124. <https://doi.org/10.15388/ioi.2016.08>
- Hu, C. (2011, Juin). Computational thinking: what it might mean and what we might do about it. Dans. à 16th annual joint conference on Innovation and technology in computer science education, Darmstadt, Allemagne.
- Hurme, T.-R. et Järvelä, S. (2005). Students' Activity in Computer-Supported Collaborative Problem Solving in Mathematics. *International Journal of Computers for Mathematical Learning*, 10(1), 49-73. <https://doi.org/10.1007/s10758-005-4579-3>
- Information Technology Foundation for Education. (2020). *ProgeTiger Programme*. Hariduse Infotehnoloogia Sihtasutus. <http://www.hitsa.ee/it-education/educational-programmes/progetiger>
- ISDE. (2019). *CodeCan*. Gouvernement du Canada. <https://www.ic.gc.ca/eic/site/121.nsf/fra/accueil>
- Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S. et Tapia, T. (2015). Empowering K–12 students with disabilities to learn computational thinking and computer programming. *TEACHING Exceptional Children*, 48(1), 45-53.
- Jaccoud, M. et Mayer, R. (1997). L'observation en situation et la recherche qualitative. Dans J. Poupart, L.-H. Groulx, J.-P. Deslauriers, A. Laperrière, R. Mayer et A. P. Pires (dir.), *La recherche qualitative: enjeux épistémologiques et méthodologiques* (p. 85-109). Gaëtan Morin.
- Johannessen, J.-A. (2018). *Automation, Innovation and Economic Crisis*. Routledge. <https://www.taylorfrancis.com/books/9781351039857>
- Jonassen, D. H. (2011). *Learning to Solve Problems: A Handbook for Designing Problem-Solving Learning Environments*. Routledge.
- Jonassen, D. H. (2014). Assessing Problem Solving. Dans J. M. Spector, D. M. Merrill, J. Elen et M. J. Bishop (dir.), *Handbook of Research on Educational Communications and Technology* (4^e éd., p. 269-288). Springer. https://doi.org/10.1007/978-1-4614-3185-5_22
- Jonassen, D. H. et Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning environments. *Educational technology research and development*, 47(1), 61-79.

- Julio, L., Kormos, E., Morgan, A. et Dopuch, T. (2018). *Latin American International School Faculty Uses and Gratifications of Technology*. EdMedia + Innovate Learning 2018, Amsterdam, Netherlands. <https://www.learntechlib.org/p/184253>
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in human behavior*, 52, 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>
- Kamga, R., Romero, M., Komis, V. et Misirli, A. (2017, 18-19 mai). *Identification des difficultés des futur(e)s enseignant(e)s du primaire et du préscolaire en lien avec la compétence de résolution collaborative de problèmes (RCP)*. au 5e Sommet du iPad et du numérique en éducation, Montréal, Canada.
- Kaptelinin, V. et Nardi, B. A. (2006). *Acting with Technology: Activity Theory and Interaction Design*. The MIT Press. <https://books.google.ca/books?hl=fr&lr=&id=qYj6AQAAQBAJ&oi=fnd&pg=PP8&dq=activity+theory&ots=cKO2KLLMmO&sig=sG0j5UrmHXQ6Wjy2I6rGvUIDxrM#v=onepage&q=activity%20theory&f=false>
- Karsenti, T. (2018). *Débat entourant l'utilisation du téléphone intelligent en classe*. FM93. <http://fm93.com/extraits-audios/opinions/72146/thierry-karsenti-de-la-faculte-des-sciences-de-leducation-de-ludem-est-avec-nous-un-jeune-de-15-ans-met-en-demeure-sa-commission-scolaire-parce-quil-juge-inconstitutionnel-certaines-sanctions-contre-lutilisation-du-cellulaire-en-classe>
- Karsenti, T. et Bugmann, J. (2017). Faut-il apprendre à coder à l'école? *École branchée*, 19(3), 32-35.
- Karsenti, T., Bugmann, J. et Parent, S. (2018). Usage d'un robot humanoïde en contexte éducatif : impacts sur la résolution collaborative de problèmes et le développement de la pensée computationnelle. Dans. Colloque international en éducation, Montréal, Canada.
- Karsenti, T. et Demers, S. (2018). L'étude de cas. Dans T. Karsenti et L. Savoie-Zajc (dir.), *La recherche en éducation : étapes et approches* (4^e éd., p. 289-316). Les Presses de l'Université de Montréal.
- Karsenti, T., Komis, V., Depover, C., Collin, S. et Bugmann, J. (2018). Les technologies. Dans T. Karsenti et L. Savoie-Zajc (dir.), *La recherche en éducation: étapes et approches* (4^e éd., p. 369-406). Les presses de l'Université de Montréal.
- Karsenti, T., Parent, S., Kerbrat, N. et Bugmann, J. (2019a). *Le robot NAO en éducation. Deviens un maître NAO* (2^e éd.). CRIFPE.
- Karsenti, T., Parent, S., Kerbrat, N. et Bugmann, J. (2019b). *Le robot NAO en éducation. Guide de l'élève*. (3^e éd.). CRIFPE.
- Kelleher, C. et Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83-137.
- Khayrallah, M. A. et Van den Meiraker, M. (1987). LOGO programming and the acquisition of cognitive skills. *Journal of Computer-Based Instruction*, 14(4), 133-137.
- Komis, V. et Misirli, A. (2011, octobre). Robotique pédagogique et concepts préliminaires de la programmation à l'école maternelle: une étude de cas basée sur le jouet programmable Bee-Bot. Dans. à la Sciences et technologies de l'information et de la communication en milieu éducatif: Analyse de pratiques et enjeux didactiques., Patras, Grèce.

- Komis, V. et Misirli, A. (2013). Étude des processus de construction d'algorithmes et de programmes par les petits enfants à l'aide de jouets programmables. Dans à Sciences et technologies de l'information et de la communication (STIC) en milieu éducatif, Clermont-Ferrand, France.
- Koschman, T., Hall, R. et Miyake, N. (2002). *CSCL 2. Carrying forward the conversation*. Lawrence Erlbaum Associates.
- Koschmann, T. (1996). Paradigm Shifts and Instructional Technology: An Introduction. Dans T. Koschmann (dir.), *CSCL: theory and practice of an emerging paradigm* (p. 1-23). Lawrence Erlbaum Associates.
- Koschmann, T., Kelson, A. C., Feltovich, P. J. et Barrows, H. S. (1996). Computer-Supported Problem-Based Learning: A Principled Approach to the Use of Computers in Collaborative Learning. Dans T. Koschmann (dir.), *CSCL: Theory and practice of an emerging paradigm* (p. 83-124). Lawrence Erlbaum Associates Inc.
- Kumar, R. S. (2020, 17 mai). Computer skills are crucial for children - in lockdown and in life. *Forum économique mondial*. <https://www.weforum.org/agenda/2020/05/coronavirus-lockdown-children-computers-stem/>
- Kuutti, K. (1996). Activity Theory as a Potential Framework for Human-Computer Interaction Research. Dans B. A. Nardi (dir.), *Context and Consciousness: Activity Theory and Human-Computer Interaction* (p. 17-44). The MIT Press. <https://books.google.ca/books?hl=fr&lr=&id=JeqcgPIS2UAC&oi=fnd&pg=PA17&dq=activity+theory&ots=e-gdTABVLv&sig=OAF1sjz0MiPemGbJrQ3Zr7QkHO8#v=onepage&q=activity%20theory&f=false>
- Lai, A.-F. et Yang, S.-M. (2011, septembre). The learning effect of visualized programming learning on 6 th graders' problem solving and logical reasoning abilities. Dans à International Conference on Electrical and Control Engineering (ICECE).
- Lasnier, F. (2014). *Les compétences de l'apprentissage à l'évaluation*. Guérin universitaire, 3e millénaire.
- Lave, J. et Wenger, É. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press.
- Le Boterf, G. (2002, février). De quel concept de compétence avons-nous besoin? *Soins cadres*, 41, 1-3.
- LeCompte, M. D. et Preissle, J. (1993). *Ethnography and Qualitative Design in Educational Research*. Academic Press.
- Lee, M., Yun, J. J., Pyka, A., Won, D., Kodama, F., Schiuma, G., Park, H., Jeon, J., Park, K., Jung, K., Yan, M.-R., Lee, S. et Zhao, X. (2018). How to respond to the Fourth Industrial Revolution, or the Second Information Technology Revolution? Dynamic new combinations between technology, market, and society through open innovation. *Journal of Open Innovation: Technology, Market, and Complexity*, 4(21). <https://doi.org/10.3390/joitmc4030021>
- Legendre, R. (2005). Dans *Dictionnaire actuel de l'éducation*. Guérin.
- Leontiev, A. (1978). *Activity and Consciousness*. Prentice-Hall.
- LeRobert. (2016). Développement. Dans J. Rey-Debove et A. Rey (dir.), *Le Petit Robert. Les logiciels éducatifs*. (1984). Québec, Ministère de l'éducation, Direction générale du développement pédagogique.

- Liben, L. S. (1987). Approaches to Development and Learning: Conflict and Congruence. Dans L. S. Liben (dir.), *Development and learning : conflict or congruence?* (p. 225-235). L. Erlbaum Associates.
- Lin, J. M.-C., Yen, L.-Y., Yang, M.-C. et Chen, C.-F. (2005). Teaching computer programming in elementary schools: A pilot study. Dans à National educational computing conference.
- Lopez, J. (1986). *Des algorithmes aux langages: basic, LSE, logo*. Hachette.
- Lu, J. J. et Fletcher, G. H. L. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin*, 41(1), 260. <https://doi.org/10.1145/1539024.1508959>
- Lye, S. Y. et Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in human behavior*, 41, 51-61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Maisonneuve, V. et Lemieux, N. (2018). La Vérif: cellulaire et cerveau font-ils bon ménage? *Radio-Canada*. <https://ici.radio-canada.ca/nouvelle/1079805/verif-cellulaire-cerveau-comportement-anxiete-sommeil-concentration-technologie>
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. et Settle, A. (2014). Computational thinking in K-9 education. Dans à Innovation & technology in computer science education conference.
- Marquardt, M. J. et Yeo, R. K. (2012). *Problem Solving with Action Learning: Concepts and Cases*. Stanford University Press.
- Mauch, E. (2001, 2001/03/01). Using Technological Innovation to Improve the Problem-Solving Skills of Middle School Students: Educators' Experiences with the LEGO Mindstorms Robotic Invention System. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 74(4), 211-213. <https://doi.org/10.1080/00098650109599193>
- McConnell, D. (2000). *Implementing computer supported cooperative learning* (2^e éd.). Kogan Page.
- Merrill, M. D. et Gilbert, C. G. (2008). Effective peer interaction in a problem-centered instructional strategy. *Distance education*, 29(2), 199-207. <https://doi.org/10.1080/01587910802154996>
- Miles, M. B. et Huberman, A. M. (2003). *Analyse des données qualitatives*. De Boeck Supérieur.
- Miller, R. B., Kelly, G. N. et Kelly, J. T. (1988, 1988/10/01/). Effects of Logo computer programming experience on problem solving and spatial relations ability. *Contemporary Educational Psychology*, 13(4), 348-357. [https://doi.org/https://doi.org/10.1016/0361-476X\(88\)90034-3](https://doi.org/https://doi.org/10.1016/0361-476X(88)90034-3)
- Ministère de l'Éducation. (2020a). *L'usage pédagogique de la programmation informatique*. http://www.education.gouv.qc.ca/fileadmin/site_web/documents/ministere/Usage-pedagogique-programmation-informatique.pdf
- Ministère de l'Éducation. (2020b). *Référentiel de compétences professionnelles. Profession enseignante*.
- Ministère de l'Éducation et de l'Enseignement supérieur. (2019). *Cadre de référence de la compétence numérique*. http://www.education.gouv.qc.ca/fileadmin/site_web/documents/ministere/continuum-cadre-reference-num.pdf
- ministère de l'Éducation nationale. (2018). *Vademecum - Interdiction de l'utilisation du téléphone portable à l'école et au collège*.

- http://cache.media.eduscol.education.fr/file/Vie_des_ecoles_et_des_ets/60/8/Vademe_cum_inderdiction-portable-ecole-college_03092018_992608.pdf
- Ministère de l'Éducation de l'Ontario. (2016). *Compétences du 21e siècle: Document de réflexion*.
- Ministère de l'Éducation et de l'Enseignement supérieur. (2018). *Robot 360 : projet pilote en programmation informatique dans les milieux scolaires*.
<https://www.newswire.ca/fr/news-releases/robot-360--projet-pilote-en-programmation-informatique-dans-les-milieux-scolaires-670495003.html>
- Ministère de l'Éducation et de l'Enseignement supérieur. (2020). *Indices de défavorisation des écoles publiques*.
- Ministry of Education and Research. (2014). *The Estonian Lifelong Learning Strategy 2020*.
https://www.hm.ee/sites/default/files/estonian_lifelong_strategy.pdf
- Mitnik, R., Nussbaum, M. et Soto, A. (2008, November 01). An autonomous educational mobile robot mediator [journal article]. *Autonomous Robots*, 25(4), 367-382.
<https://doi.org/10.1007/s10514-008-9101-z>
- Modeste, S. (2012, février). La pensée algorithmique: apports d'un point de vue extérieur aux mathématiques. Dans. au Colloque Espace Mathématique Francophone, Genève, Suisse.
- Mollo, V. et Falzon, P. (2004). Auto-and allo-confrontation as tools for reflective activities. *Applied ergonomics*, 35(6), 531-540.
<https://doi.org/https://doi.org/10.1016/j.apergo.2004.06.003>
- Morasse, M.-È. (2018, 20 janvier). Téléphone à l'école: le débat est relancé. *LaPresse*.
<http://www.lapresse.ca/actualites/education/201801/19/01-5150800-telephone-a-lecole-le-debat-est-relance.php>
- Moreno-León, J. et Robles, G. (2015, mars). Computer programming as an educational tool in the English classroom. Dans. IEEE Global Engineering Education Conference, Tallinn, Estonie.
- Mosley, P. et Kline, R. (2006). Engaging Students: A Framework Using LEGO Robotics to Teach Problem Solving. *Information Technology, Learning, and Performance Journal*, 24(1), 39-45.
- Mubin, O., Stevens, C. J., Shahid, S., Al Mahmud, A. et Dong, J.-J. (2013). A review of the applicability of robots in education. *Technology for Education and Learning*, 1(1), 1-7.
<https://doi.org/10.2316/Journal.209.2013.1.209-0015>
- Mugny, G. et Doise, W. (1978). Socio-cognitive conflict and structure of individual and collective performances. *European journal of social psychology*, 8(2), 181-192.
- Neubert, J. C., Mainert, J., Kretzschmar, A. et Greiff, S. (2015). The assessment of 21st century skills in industrial and organizational psychology: Complex and collaborative problem solving. *Industrial Organizational Psychology*, 8(2), 238-268.
<https://doi.org/https://doi.org/10.1017/iop.2015.14>
- Noh, J. et Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational technology research and development*, 68(1), 463-484. <https://doi.org/https://doi.org/10.1007/s11423-019-09708-w>
- Nugent, G., Barker, B., Grandgenett, N. et Adamchuk, V. (2009, 18-21 octobre). *The use of digital manipulatives in k-12: robotics, GPS/GIS and programming*. à 39th ASEE/IEEE Frontiers in Education Conference, Texas, États-Unis.

- OCDE. (2015a). *Résultats du PISA 2012 - Trouver des solutions créatives (Volume 5): Compétences des élèves en résolution de problèmes de la vie réelle*. (vol. 5). Éditions OCDE. <http://www.oecd-ilibrary.org/docserver/download/9814012e.pdf?expires=1504712563&id=id&accname=guest&checksum=7F57C00001A83DB0C35EA4D2FA64E212>
- OCDE. (2015b). *Students, Computers and Learning: Making the Connection*.
- OCDE. (2017). *PISA 2015. Collaborative Problem Solving Framework*. OCDE. <https://www.oecd.org/pisa/pisaproducts/Draft%20PISA%202015%20Collaborative%20Problem%20Solving%20Framework%20.pdf>
- OCDE. (2020a). *Education Policy Outlook: Estonia*. OCDE. <http://www.oecd.org/education/policy-outlook/country-profile-Estonia-2020.pdf>
- OCDE. (2020b). *The potential of online learning for adults: Early lessons from the COVID-19 crisis*. OCDE. <https://www.oecd.org/coronavirus>
- Özüorçun, N. Ç. et Bicen, H. (2017). Does the Inclusion of Robots Affect Engineering Students' Achievement in Computer Programming Courses? *EURASIA Journal of Mathematics, Science and Technology Education*, 13(8), 4779-4787. <https://doi.org/10.12973/eurasia.2017.00964a>
- Paillé, P. (2007). La méthodologie de recherche dans un contexte de recherche professionnalisante: douze devis méthodologiques exemplaires. *Recherches qualitatives*, 27(2), 133-151.
- Paillé, P. et Mucchielli, A. (2005). *L'analyse qualitative en sciences humaines et sociales*. Armand Colin.
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books
- Papert, S. (1981). *Jaillissement de l'esprit*. Flammarion.
- Papert, S. (1994). *L'enfant et la machine à connaître : repenser l'école à l'ère de l'ordinateur*. Dunod.
- Peelle, H. A. (1974). Computer Glass Boxes: Teaching Children Concepts with "A Programming Language (APL)". *Educational Technology*, 14(4), 9-16. www.jstor.org/stable/44420812
- Petre, M. et Price, B. (2004, June 01). Using Robotics to Motivate 'Back Door' Learning [journal article]. *Education and Information Technologies*, 9(2), 147-158. <https://doi.org/10.1023/b:eait.0000027927.78380.60>
- Pineau, E. (2018). Les « smartphones » interdits à l'école à la rentrée en France. *Reuters*. <https://fr.reuters.com/article/technologyNews/idFRKCN1J31X1-OFRIN>
- Pinet, N. (1982). *L'ordinateur, objet d'étude et outil pédagogique dans l'enseignement secondaire*.
- Pinto-Llorente, A. M., Casillas-Martín, S., Cabezas-González, M. et García-Peñalvo, F. J. (2017). Building, coding and programming 3D models via a visual programming environment. *International Journal of Methodology*, 1-14.
- Pires, A. P. (1997). Échantillonnage de recherche qualitative: essai théorique et méthodologique. Dans J. Poupart, L.-H. Groulx, J.-P. Deslauriers, A. Laperrière, R. Mayer et A. P. Pires (dir.), *La recherche qualitative: enjeux épistémologiques et méthodologiques* (p. 113-167). Gaëtan Morin.
- Polya, G. (1957). *How to solve it: A new aspects of mathematical methods*. Princeton University Press.

- Powers, S. I., Welsh, D. P. et Wright, V. (1994). Adolescents' affective experience of family behaviors: The role of subjective understanding. *Journal of Research on Adolescence*, 4(4), 585-600. https://doi.org/10.1207/s15327795jra0404_9
- Pullan, M. (2013). Using Robotics to Improve Retention and Increase Comprehension in Introductory Programming Courses. *Journal of Educational Technology Systems*, 42(2), 141-149. <https://doi.org/10.2190/ET.42.2.f>
- QSR International. (2020). NVivo 12 (version 12.6.0) [logiciel]. QSR International.
- RÉPAQ. (2020). *Portrait de l'école alternative*. Réseau des écoles publiques alternatives du Québec. <https://repaq.org/portrait/>
- Repenning, A., Webb, D. C., Brand, C., Gluck, F., Grover, R., Miller, S., Nickerson, H. et Song, M. (2014). Beyond minecraft: Facilitating computational thinking through modeling and programming in 3d. *IEEE Computer Graphics and Applications*, 34(3), 68-71.
- Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. Dans. 6th ACM SIGCHI conference on Creativity & cognition.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J. et Silverman, B. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.
- Rice, J. K. et Desautels, E. (1969). *Introduction to computer science; problems, algorithms, languages, information and computers*. Holt, Rinehart and Winston.
- Robertson, S. I. (2017). *Problem solving: Perspectives from Cognition and Neuroscience* (2^e éd.). Routledge.
- Romero, M. (2016a). De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative. *Formation et Profession*, 24(1), 87-89. <https://doi.org/10.18162/fp.2016.a92>
- Romero, M. (2016b, 30 août). *Faut-il enseigner l'art du code à nos jeunes?* Radio-Canada.
- Romero, M. (2017). Les compétences pour le XXI^e siècle. Dans M. Romero, B. Lille et A. Patiño (dir.), *Usages créatifs du numérique pour l'apprentissage au XXI^e siècle*. Presses de l'Université du Québec.
- Romero, M., Lepage, A. et Lille, B. (2017, December 12). Computational thinking development through creative programming in higher education [journal article]. *International Journal of Educational Technology in Higher Education*, 14(1), 42. <https://doi.org/10.1186/s41239-017-0080-z>
- Romero, M., Lille, B., Viéville, T., Dufлот-Kremer, M., de Smet, C. et Belhassein, D. (2018, Août). Analyse comparative d'une activité d'apprentissage de la programmation en mode branché et débranché. Dans. Conférence internationale sur l'enseignement au numérique et par le numérique, Bruxelles, Belgique.
- Romero, M. et Vallerand, V. (2016). *Guide d'activités technocréatives pour les enfants du 21^e siècle*. CoCreaTIC.
- Roschelle, J. et Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem solving. Dans. à Computer supported collaborative learning. NATO ASI Series (Series F: Computer and Systems Sciences), Berlin, Allemagne.
- Rosen, Y. et Foltz, P. W. (2014). Assessing collaborative problem solving through automated technologies. *Research & Practice in Technology Enhanced Learning*, 9(3), 389-410.

- Roy, N. et Garon, R. (2013). Étude comparative des logiciels d'aide à l'analyse de données qualitatives: de l'approche automatique à l'approche manuelle. *Recherches qualitatives*, 154(1).
- Ruf, A., Berges, M. et Hubwieser, P. (2015, septembre). Classification of Programming Tasks According to Required Skills and Knowledge Representation. Dans. à 8th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (ISSEP), Ljubljana, Slovénie.
- Saeli, M., Perrenet, J., Jochems, W. M. G. et Zwaneveld, B. (2011). Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective. *Informatics in Education* (1), 73-88.
- Sáez-López, J.-M., Román-González, M. et Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Savoie-Zajc, L. (2018). La recherche qualitative/interprétative. Dans T. Karsenti et L. Savoie-Zajc (dir.), *La recherche en éducation: étapes et approches* (4^e éd., p. 191-217). Les Presses de l'Université de Montréal.
- Savoie-Zajc, L. et Karsenti, T. (2018). La méthodologie. Dans L. Savoie-Zajc et T. Karsenti (dir.), *La recherche en éducation: étapes et approches* (4^e éd., p. 140-152). Les Presses de l'Université de Montréal.
- Scardamalia, M. et Bereiter, C. (1994). Computer support for knowledge-building communities. *The journal of the learning sciences*, 3(3), 265-283.
- Schleicher, A. (2019, 1er septembre). Should schools teach coding? *Global perspectives on education and skills*. <https://oecdeditoday.com/should-schools-teach-coding/>
- Sebire, K. B., Gregory, S. et Bannister-Tyrrell, M. (2018). *Learning in the Age of Distraction*. EdMedia + Innovate Learning 2018, Amsterdam, Netherlands. <https://www.learntechlib.org/p/184234>
- Sedgewick, R. et Wayne, K. (2011). *Algorithms* (4^e éd.). Addison-Wesley.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G. et Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2), 351-380. <https://doi.org/10.1007/s10639-012-9240-x>
- Serafini, G. (2011). Teaching programming at primary schools: visions, experiences, and long-term research prospects. Dans. International Conference on Informatics in Schools: Situation, Evolution, and Perspectives.
- Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C. et Wildeman, B. (2012, juillet). Infusing computational thinking into the middle-and high-school curriculum. Dans. à 17th ACM annual conference on Innovation and technology in computer science education, Haifa, Israel.
- Shim, J., Kwon, D. et Lee, W. (2016). The effects of a robot game environment on computer programming education for elementary school students. *IEEE Transactions on Education*, 60(2), 164-172. <https://doi.org/10.1109/TE.2016.2622227>
- Simon, H. A. et Newell, A. (1971). Human problem solving: The state of the theory in 1970. *American Psychologist*, 26(2), 145.

- Smart, S. (2016). *\$6 million to help connect students with coding, new curriculum and computers*. <https://news.gov.bc.ca/releases/2016PREM0065-000994>
- Société canadienne de pédiatrie. (2018). *Le temps d'écran et les jeunes enfants : promouvoir la santé et le développement dans un monde numérique*. <https://www.cps.ca/fr/documents/position/le-temps-d-ecran-et-les-jeunes-enfants>
- St-Amand Tellier, B. (2018). *Le développement des compétences transversales: étude exploratoire des perceptions d'acteurs dans une école alternative au Québec* [Université du Québec à Trois-Rivières]. <http://depot-e.uqtr.ca/id/eprint/8740/1/032180745.pdf>
- Stadler, M., Herborn, K., Mustafić, M. et Greiff, S. (2020). The assessment of collaborative problem solving in PISA 2015: An investigation of the validity of the PISA 2015 CPS tasks. *Computers & Education*, 157, 103964. <https://doi.org/https://doi.org/10.1016/j.compedu.2020.103964>
- Stahl, G. (2006). *Group Cognition: Computer Support for Building Collaborative Knowledge*. MIT Press.
- Stake, R. (1995). *The art of case study research*. Sage Publications.
- Stake, R. E. (1995). *The art of case study research*. Sage Publications.
- Stockless, A., Villeneuve, S. et Gingras, B. (2018). Maitrise d'outils technologiques : son influence sur la compétence TIC des enseignants et les usages pédagogiques | Mastery of Digital Tools: The Influence on Information and Communication Technologies Competency and Pedagogical Use. *Canadian Journal of Learning and Technology / La revue canadienne de l'apprentissage et de la technologie*, 44(2).
- Stoutemyer, D. R. (1979). Computer symbolic math & education: a radical proposal. *SIGSAM Bull.*, 13(3), 8–24. <https://doi.org/10.1145/1089170.1089173>
- Strayer, F. et Gauthier, R. (1982). L'approche éthologique de l'observation du comportement. *Apprentissage et socialisation*, 5(1).
- Sullivan, F. R. (2008). Robotics and science literacy: Thinking skills, science process skills and systems understanding. *Journal of Research in Science Teaching*, 45(3), 373-394. <https://doi.org/10.1002/tea.20238>
- Taggar, S. et Brown, T. (2001). Problem-solving team behaviors: Development and validation of BOS and a hierarchical factor structure. *Small Group Research*, 32(6), 698-726.
- Tardif, J. (2006). *L'évaluation des compétences : documenter le parcours de développement*. Chenelière Éducation.
- Tatar, D., Harrison, S., Stewart, M., Frisina, C. et Musaeus, P. (2017). Proto-computational Thinking: The Uncomfortable Underpinnings. Dans P. J. Rich et C. B. Hodges (dir.), *Emerging Research, Practice and Policy on Computational Thinking* (p. 49-62). Springer.
- Tausczik, Y. R., Kittur, A. et Kraut, R. E. (2014, 15 au 19 février). Collaborative problem solving: A study of MathOverflow. Dans CSCW, Baltimore, États-Unis.
- Tchounikine, P. (2016). *Initier les élèves à la pensée informatique et à la programmation avec Scratch*. <http://codebc.ca/wp-content/uploads/2017/02/PenseeInformatiqueEcole.pdf>
- Teasley, S. D. et Roschelle, J. (1993). Constructing a joint problem space: The computer as a tool for sharing knowledge. Dans S. LaJoie et S. Derry (dir.), *Computers as cognitive tools* (p. 229-258). Routledge. <https://books.google.ca/books?hl=fr&lr=&id=xGeasuCsN2MC&oi=fnd&pg=PA229&dq=Constructing+a+Joint+Problem+Space:+The+Computer+as+a+Tool+for+Sharing+Knowledg>

- e&ots=O5kvv-
mwMQ&sig=NE96Ra5tTxbMzVdjDMAdAtZY_Ck#v=onpage&q=Constructing%20a%20Joint%20Problem%20Space%3A%20The%20Computer%20as%20a%20Tool%20for%20Sharing%20Knowledge&f=false
- Toikkanen, T. (2015). *Coding in school: Finland takes lead in Europe*. <http://legroup.aalto.fi/2015/11/coding-in-school-finland-takes-lead-in-europe/>
- Tondeur, J., Van Braak, J. et Valcke, M. (2007). Towards a typology of computer use in primary education. *Journal of Computer Assisted Learning*, 23(3), 197-206.
- Touretzky, D. S., Marghitsu, D., Ludi, S., Bernstein, D. et Ni, L. (2013). Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. Dans. à 44th ACM technical symposium on Computer science education.
- Turski, W. M. (1978). *Computer programming methodology*. London.
- Université de Montréal. (2014). *Guide d'information sur le consentement libre, éclairé et continu*. Université de Montréal.
- van der Maren, J.-M. (2004). *Méthodes de recherche pour l'éducation* (2^e éd.). Presses de l'Université de Montréal.
- van Laar, E., van Deursen, A. J., van Dijk, J. A. et de Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. *Computers in human behavior*, 72, 577-588.
- Van Roy, P. et Haridi, S. (2004). *Concepts, techniques, and models of computer programming*. MIT Press.
- van Someren, M. W., Barnard, Y. F. et Sandberg, J. A. C. (1994). *The think aloud method: a practical guide to modelling cognitive processes*. Academic Press.
- Vigneault, A. (2018). Les ados minés par le cellulaire? *La Presse*. <https://www.lapresse.ca/vivre/societe/201802/22/01-5154897-les-ados-mines-par-le-cellulaire.php>
- Voogt, J., Fisser, P., Good, J., Mishra, P. et Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728. <https://doi.org/10.1007/s10639-015-9412-6>
- Vygotsky, L. S. (1934). *Thought and language*. M.I.T. Press.
- Vygotsky, L. S. (1997). *Pensée et langage*. La Dispute.
- Webb, N. M. et Lewis, S. (2008). The social context of learning computer programming. Dans R. E. Mayer (dir.), *Teaching and learning computer programming: Multiple research perspectives* (p. 179-206). Routledge.
- Wei, X., Lin, L., Meng, N., Tan, W. et Kong, S.-C. (2021). The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. *Computers & Education*, 160, 104023. <https://doi.org/https://doi.org/10.1016/j.compedu.2020.104023>
- Werner, L., Denner, J., Campe, S. et Kawamoto, D. C. (2012, mars). The fairy performance assessment: measuring computational thinking in middle school. Dans. à 43rd ACM technical symposium on Computer Science Education, New York, États-Unis.
- Wilson, A., Hainey, T. et Connolly, T. M. (2013). Using Scratch with Primary School Children. *International Journal of Game-Based Learning*, 3(1), 93-109. <https://doi.org/10.4018/ijgbl.2013010107>

- Wilson, A. et Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. Dans. Psychology of Programming Interest Group Workshop, Madrid, Espagne.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2010). *Computational Thinking: What and Why?* <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- Wood, D., Bruner, J. S. et Ross, G. (1976). The role of tutoring in problem solving. *Journal of child psychology psychiatry*, 17(2), 89-100.
- Xu, Z., Ritzhaupt, A. D., Tian, F. et Umaphy, K. (2019, 2019/07/03). Block-based versus text-based programming environments on novice student learning outcomes: a meta-analysis study. *Computer Science Education*, 29(2-3), 177-204. <https://doi.org/10.1080/08993408.2019.1565233>
- Yadav, A., Good, J., Voogt, J. et Fisser, P. (2017). Computational Thinking as an Emerging Competence Domain. Dans M. Mulder (dir.), *Competence-based Vocational and Professional Education: Bridging the Worlds of Work and Education* (p. 1051-1067). Springer International Publishing. https://doi.org/10.1007/978-3-319-41713-4_49
- Yin, K. Y. et Abdullah, A. G. K. (2013). The collaborative problem solving questionnaire: Validity and reliability test. *International Journal of Academic Research in Business and Social Sciences*, 3(1), 511.
- Zehner, F., Weis, M., Vogel, F., Leutner, D. et Reiss, K. (2019). Kollaboratives Problemlösen in PISA 2015: Deutschland im Fokus. *Zeitschrift für Erziehungswissenschaft*, 22(3), 617-646.
- Zumbach, J., Kumpf, D. et Koch, S. (2004). Using multimedia to enhance problem-based learning in elementary school. *Information technology in childhood education annual*, 1(1), 25-37.

Annexes

Annexe 1 : Scénario pédagogique *Deviens un maître NAO*.

Annexe 2 : Modèle de la compétence de RCP (Kamga et al., 2017)

Annexe 3 : Grille d'observation de la RCP (version synthétique)

Annexe 1 – Scénario pédagogique *Deviens un maître NAO*

Karsenti, T., Parent, S., Kerbrat, N. et Bugmann, J. (2019a). *Le robot NAO en éducation. Deviens un maître NAO*. (2e éd.). Montréal, Canada: CRIFPE.

Deviens un MAÎTRE NAO



Niveaux

Défis

Niveaux

Défis

1

- ◆ Vous devez doucement caresser le dessus de la tête de NAO pour qu'il réagisse.
- ◆◆ Vous devez dire « Salut » ou « Bonjour » à NAO jusqu'à ce qu'il vous comprenne et vous réponde. Demandez-lui, ensuite, « comment ça va ? ».
- ◆◆◆ Vous devez demander à NAO de se coucher sur le dos ou sur le ventre.

6

- ◆ Vous devez faire poser la question suivante à NAO : « Aimes-tu les desserts ? » en utilisant la boîte *Choice*. Il doit vous comprendre et vous répondre « Moi, j'adore ça ! »
- ◆◆ Vous devez faire poser la question suivante à NAO : « Quelle est la capitale du Canada ? ». Vous devrez proposer 3 choix de réponse, faire en sorte qu'il reconnaisse et félicite la bonne réponse ou demande de réessayer en cas de mauvaise réponse.
- ◆◆◆ Vous devez faire poser la question suivante à NAO : « Quelle est la capitale des États-Unis ? ». Veuillez à proposer 3 choix de réponse. NAO devra lever la main droite en cas de bonne réponse et lever la main gauche en cas de mauvaise réponse.

2

- ◆ Vous devez faire s'asseoir NAO en le programmant.
- ◆◆ Vous devez faire dire « Bonjour » à NAO en le programmant.
- ◆◆◆ Vous devez lui faire faire un *Bonjour animé* dans lequel il dira « Bonjour mon ami » en le programmant.

7

- ◆ Vous devez faire en sorte que NAO reconnaisse la balle rouge et la suive avec la tête.
- ◆◆ Vous devez faire en sorte que NAO reconnaisse la balle rouge, se dirige vers elle, et s'arrête à 0,2 mètre d'elle.
- ◆◆◆ Vous devez faire en sorte que NAO reconnaisse la balle rouge, se dirige vers elle en levant le bras droit, s'arrête à 0,4 mètre et dise la phrase : « Ma balle est ici, je la cherchais justement ! ».

BRONZE

3

- ◆ **Utiliser la fenêtre Robot View pour ce niveau**
Vous devez faire tourner la tête de NAO à gauche.
- ◆◆ Vous devez lui faire lever le bras droit.
- ◆◆◆ Vous devez lui faire bouger les deux bras en même temps.

8

- ◆ Vous devez faire apprendre à NAO deux visages.
- ◆◆ Vous devez faire en sorte que NAO reconnaisse votre visage et dise votre nom lorsque vous serez devant lui.
- ◆◆◆ Vous devez faire en sorte que NAO reconnaisse deux visages et dise un message personnalisé différent en identifiant chaque personne.

ARGENT

4

- ◆ Vous devez faire avancer NAO d'un mètre en le programmant.
- ◆◆ Vous devez faire reculer NAO d'un mètre en le programmant.
- ◆◆◆ Vous devez faire avancer NAO de 0,5 mètre vers la gauche et lui faire faire un « *Bonjour animé* » où il dit « Content de vous rencontrer ! » en le programmant.

9

- ◆ **Utiliser le mode animation pour ce niveau**
Vous devez faire jouer une musique de votre choix à NAO.
- ◆◆ Vous devez faire jouer cette musique à NAO et lui faire faire une chorégraphie avec les bras (*boîte Timeline*).
- ◆◆◆ Vous devez lui faire jouer la musique, puis lui faire faire une chorégraphie avec les bras, la tête et les jambes (*boîte Timeline*). Enregistrez la chorégraphie dans la librairie.

OR

5

- ◆ **Utiliser le mode animation pour ce niveau**
Vous devez faire lever le bras gauche de NAO, remplacer le nom de la boîte *Timeline* par « Bras gauche levé » et changer l'image en choisissant une proposée par le logiciel.
- ◆◆ Vous devez faire lever les deux bras de NAO.
- ◆◆◆ Vous devez donner une nouvelle position à NAO sans qu'il ne tombe.

10

- ◆ **Utiliser le mode animation pour ce niveau**
Vous devez faire demander à NAO : « Veux-tu que je te montre mon super parcours d'activité physique ? ». A la réponse « oui », une musique commencera et NAO débutera sa série d'exercices que vous aurez programmé :
 - NAO devra faire un trajet en forme de rectangle (1 mètre en avant, 0,3 m à droite, 1 m en arrière et 0,3 m à gauche) ;
 - Il devra faire un exercice de votre choix au début, au milieu et à la fin du parcours en *mode animation*. Vous avez la possibilité de reprendre la chorégraphie du niveau 9 pour un des exercices.
 - Quand il aura terminé il dira : « Ouf, ça fait du bien ! » et s'assiéra sur le

PLATINE

Deviens un MAÎTRE NAO



Niveaux

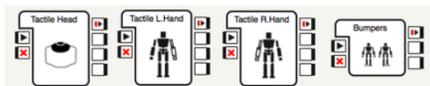
Défis

Niveaux

Défis

11

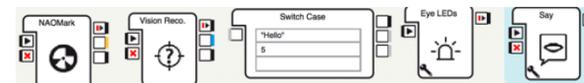
Vous devez programmer des actions différentes en fonction des capteurs sensoriels qui sont touchés. Voici certaines des boîtes à utiliser :



Vous avez le choix : Demander à NAO de dire quelque chose, de danser, de bouger, de poser une question, etc...

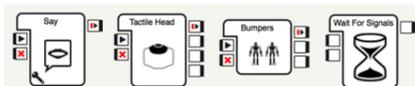
16

Vous devez demander à NAO de reconnaître visuellement un ballon bleu, rouge ou vert. Il doit répondre correctement en parlant et en indiquant la couleur avec ses yeux. Vous pouvez lui apprendre les couleurs ou alors utiliser les NaoMarks. Voici certaines des boîtes dont vous pourriez avoir besoin :



12

Vous devez programmer NAO pour lui faire dire quelque chose uniquement après avoir appuyé sur deux de ses capteurs, et cela l'un après l'autre. Voici certaines des boîtes à utiliser pour réaliser ce programme :



17

Vous devez faire dire à NAO « Bonjour jeunes humains, je suis content de vous voir après tout ce temps passé dans ma boîte », avec le mode *Python Script*. Voici certaines des boîtes à utiliser pour réaliser ce programme :



13

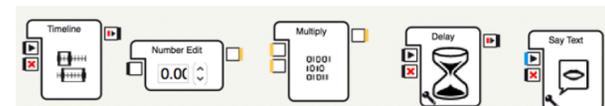
Vous devez faire un diagramme qui comporte 4 phrases que NAO devra dire au hasard. Voici certaines des boîtes à utiliser pour réaliser ce programme :



Ces 4 phrases doivent toutes être différentes et parler des robots.

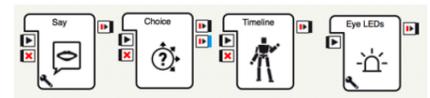
18

A l'aide d'une boîte *Timeline*, vous devez faire s'agenouiller NAO sans qu'il ne tombe. Ensuite, vous demanderez à NAO d'énoncer un problème mathématique que vous aurez créé, qui demande de multiplier deux nombres entiers entre eux. NAO devra y répondre. Voici certaines des boîtes à utiliser pour réaliser ce programme :



14

Vous devez faire faire à NAO des mouvements (Boîtes *Timeline*) qui représentent des émotions (la tristesse, la peur, la joie, la colère et l'amour) et lui faire demander à quelqu'un de quelle émotion il s'agit. Il félicite la bonne réponse et enchaîne avec une nouvelle émotion à deviner jusqu'à ce que toutes les émotions soient correctement identifiées. Voici certaines des boîtes à utiliser pour réaliser ce programme :



19

A l'aide d'une boîte *Dialog*, vous devez créer un programme qui permet à NAO de répondre à vos questions sur différentes capitales du monde. Voici certaines des boîtes à utiliser :



15

Vous devez construire une suite d'actions qui contient 10 boîtes différentes. Toutes les boîtes doivent avoir une utilité.

Attention, il n'est pas possible de retrouver deux fois la même boîte.



En utilisant les 3 types de boîtes *Dialog*, *Timeline* et *Python* vous devez donner des consignes de vie de classe à NAO en anglais (Assoie-toi, lève-toi, écoute, Va au tableau, lève la main, excuse-toi...). Il doit les exécuter.

Annexe 2 – Modèle de la compétence de RCP (Kamga et al., 2017)

	(1) Établir et maintenir une compréhension partagée	(2) Entreprendre des actions appropriées pour résoudre le problème	(3) Établir et maintenir l'organisation de l'équipe	(4) Co-régulation itérative des solutions intermédiaires	(5) Recherche et partage des ressources externes
(A) Explorer et comprendre	(A1) Découvrir les points de vue et les habiletés des membres de l'équipe.	(A2) Découvrir le type d'interaction collaborative pour résoudre le problème et atteindre les buts.	(A3) Comprendre les rôles nécessaires à la résolution du problème.	(A4) Développer la confiance, l'acceptation de l'échec et la résilience lors de l'évaluation des solutions intermédiaires au problème.	(A5) Rechercher les ressources externes qui peuvent aider à résoudre le problème.
(B) Représenter et formuler	(B1) Construire une représentation partagée du problème et en négocier le sens (espace partagé).	(B2) Identifier et décrire les tâches à réaliser.	(B3) Décrire les rôles et l'organisation d'équipe (protocoles de communication, règles d'engagement).	(B4) Être capable de concevoir la résolution de problèmes comme une approche itérative orientée vers des prototypes.	(B5) Extraire et partager les informations avec les autres membres de l'équipe.
(C) Planifier et exécuter	(C1) Communiquer avec les membres de l'équipe au sujet des actions entreprises ou à entreprendre.	(C2) Réalisation des plans.	(C3) Suivre les règles d'engagement (par exemple, inviter les autres membres à réaliser leurs tâches).	(C4) Co-réguler des efforts comme itérations de solutions intermédiaires qui conduisent vers une solution optimale.	(C5) Se mettre d'accord avec les autres membres de l'équipe sur la façon d'utiliser les ressources.
(D) Monitorer et réfléchir	(D1) Effectuer un suivi et rétablir la compréhension partagée.	(D2) Effectuer un suivi des résultats des actions et évaluer le succès de la résolution du problème.	(D3) Effectuer un suivi, offrir une rétroaction et adapter l'organisation d'équipe et les rôles.	(D4) Évaluer les avantages et les inconvénients des solutions intermédiaires et adapter les prochaines itérations en conséquence.	(D5) Évaluer avec les autres membres de l'équipe l'efficacité des ressources utilisées.

Annexe 3 – Grille d’observation de la RCP (version synthétique)

Indicateur	Descripteur
D1. Établissement et maintien d’une compréhension partagée	
1.1. Échanger	Au cours du processus de RP, les élèves échangent à propos des caractéristiques inhérentes au problème à résoudre et s’assurent d’en avoir une compréhension partagée.
1.2. Solliciter les points de vue	Les élèves cherchent à connaître le point de vue des autres membres de l’équipe quant au problème à résoudre.
1.3. Adapter les interventions	Les élèves adaptent leurs interventions en fonction d’un ou de plusieurs coéquipiers afin de maintenir la compréhension de la tâche.
D2. Réalisation des actions appropriées à la résolution du problème	
2.1. Mettre en œuvre les actions	Les élèves discutent de la façon d’effectuer une ou plusieurs actions (séquence) pour résoudre le problème, pendant l’activité (on ne cherche pas à savoir quelles actions doivent être faites [<i>Échanger</i>], on cherche à savoir comment les mettre en œuvre).
2.2. Vérifier les actions	Les élèves vérifient que les actions ont produit les effets escomptés. Il s’agit d’une vérification partielle de la solution.
2.3. Vérifier la solution	Les élèves vérifient que les actions ont permis de résoudre le problème. Il s’agit d’une vérification de la solution intégrale.
2.4. Adapter les solutions inadéquates	Le cas échéant, les élèves adaptent les solutions inadéquates dans le but de les rendre adéquates.
2.5. Participer activement*	De façon générale, les élèves participent activement aux tâches en vue de résoudre le problème.
2.6. Persévérer*	De façon générale, les élèves persévèrent dans l’activité de résolution de problème.
D3. Établissement et maintien de l’organisation de l’équipe	
3.1. Attribuer des rôles	Les élèves se sont attribués, entre eux, explicitement ou implicitement, des rôles. Cette attribution peut être unique, en amont de l’activité, ou itérative, selon les conditions mises en place par les membres de l’équipe. Il est possible, voire souhaitable, que des équipes effectuent des rotations constantes.
3.2. Identifier les forces et faiblesses	Les élèves identifient les forces et les faiblesses des membres de l’équipe.
3.3. Résoudre des différends	Les élèves sont capables de résoudre les différends ou parviennent à un consensus lorsqu’il y a des divergences d’opinions avec conflit (les divergences d’opinions sans conflit peuvent survenir lors des échanges et de la sollicitation des points de vue sans qu’il y ait de différend à résoudre).
3.4. Communiquer clairement*	De façon générale, les élèves utilisent des termes précis, spécifiques ou techniques pour établir une communication claire quant à la tâche à réaliser.
3.5. Agir en tant qu’équipe*	De façon générale, les élèves démontrent un intérêt pour la cohésion du groupe et le travail en collaboration.
3.6. Respecter les rôles attribués*	De façon générale, les élèves respectent les rôles qu’ils se sont attribués, explicitement ou implicitement. Cette attribution peut être unique, en amont de l’activité, ou itérative, selon les conditions mises en place par les membres de l’équipe.

* Indicateurs de séance (cote globale).

