



Étude des processus de construction d'algorithmes et de programmes par les petits enfants à l'aide de jouets programmables

Vassilis Komis, Anastasia Misirli

► To cite this version:

Vassilis Komis, Anastasia Misirli. Étude des processus de construction d'algorithmes et de programmes par les petits enfants à l'aide de jouets programmables. Drot-Delange, B. ; Baron, G-L. & Bruillard, E. Sciences et technologies de l'information et de la communication (STIC) en milieu éducatif, 2013, Clermont-Ferrand, France. 2013. <edutice-00875628>

HAL Id: edutice-00875628

<https://edutice.archives-ouvertes.fr/edutice-00875628>

Submitted on 22 Oct 2013

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

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

Étude des processus de construction d'algorithmes et de programmes par les petits enfants à l'aide de jouets programmables

V. Komis, A. Misirli
komis@upatras.gr, amisirli@upatras.gr

Université de Patras-Grèce

Résumé L'objectif du travail est l'étude des processus de construction d'algorithmes et de programmes par les enfants de l'école maternelle en utilisant des jouets programmables. Dans ce cadre, nous avons conçu et implémenté des scénarios pédagogiques dont l'objectif est le développement des concepts préliminaires d'algorithmique et de programmation par les enfants de cet âge. Par une analyse détaillée de la manière avec laquelle une centaine d'enfants de 4 à 6 ans conçoivent, verbalisent et exécutent des algorithmes en utilisant le jouet programmable Bee-Bot nous essayons d'éclaircir certaines questions par rapport à la construction des concepts préliminaires de programmation. Nos résultats montrent que les enfants dans leur majorité construisent des algorithmes séquentiels avec plusieurs commandes (de six à douze). Le débogage devient un processus habituel car le tiers des enfants procède à des corrections effectives de leur programme. Enfin, les enfants mettent en route plusieurs stratégies de programmation.

Mots-clés: Robotique, Jouets programmables, Algorithmique, Programmation, Stratégies cognitives

Introduction

Depuis plus de quarante ans, la robotique trouve des usages intéressants en éducation, qui sont désignés par un courant éducatif que l'on appelle la robotique pédagogique (Denis & Baron, 1994). Ce courant s'inscrit dans l'approche de l'apprentissage par la découverte, où le dispositif robotique est associé à un langage de programmation (de type LOGO) à l'aide duquel les apprenants développent des compétences transversales. La robotique pédagogique s'adresse à des différents types d'apprenants (de l'école maternelle à la formation d'adultes) dans un objectif d'initiation à la démarche scientifique et de développement des compétences techniques et informatiques. Ce contexte se situe au carrefour de deux approches pédagogiques très fertiles au plan cognitif : les activités de manipulation et de construction des objets tangibles à la base des dispositifs de type Lego ou autres, et les micromondes programmables (Depover, Karsenti & Komis, 2007). Ces dernières années, la robotique pédagogique s'est étendue vers des publics plus jeunes, dès l'école maternelle, grâce à des nouveaux outils robotiques, dont les aspects ludiques et les interfaces tangibles favorisent un usage précoce des concepts de robotique et de programmation. Il s'agit des jouets programmables se plaçant dans l'approche épistémologique et psychopédagogique du langage LOGO (Robert, 1985), c'est-à-dire dans un contexte de résolution de problèmes et de développement de la pensée algorithmique.

Dans la présente recherche nous utilisons des jouets programmables qui sont très appropriés au niveau développemental pour les enfants de l'école maternelle et ils permettent d'exercer la programmation de manière tangible et concrète. L'usage d'outils tangibles favorise la motivation des enfants envers l'apprentissage et leur implication dans des activités inscrites dans un contexte ludique et plein de signification. De plus, dans ce contexte de l'apprentissage par le jeu, la construction des notions abstraites et des habiletés sociales, très importantes pour cette classe d'âge pourrait être renforcée (Bers & Horn, 2010 ; Yelland, 2007).

Le jouet programmable que nous avons utilisé est le Bee-Bot (une « abeille programmable »), qui constitue une sorte d'équivalent contemporain de la tortue Logo avec commandes à cartes utilisée dans les années 1980. C'est un robot de sol ayant une interface tangible pour le programmer directement sans faire appel à un logiciel informatique. Les enfants peuvent, par conséquent le

programmer pour effectuer des trajets sur le sol. Le contrôle des actions du Bee-Bot se fait à partir d'une interface se trouvant sur la partie supérieure du jouet et comportant un ensemble de boutons de différentes couleurs (figure 1).



Figure 1 : Programmation tangible du jouet programmable et programme en cartes de commandes

Méthodologie

L'étude de cas présentée ici a eu lieu dans sept classes de l'école maternelle de la ville de Patras en Grèce et elle concerne 108 enfants (55 garçons et 53 filles dont 45 ont entre 4 et 5 ans et 63 entre 5 et 6 ans). Nous avons utilisé un scénario en robotique pédagogique qui a été conçu dans le cadre du projet européen Fibonacci (Komis & Misirli, 2011, Komis & Misirli, 2012a). Les enseignantes des écoles ont été formées dans le cadre de ce projet pour appliquer ce scénario basé sur les jouets programmables Bee-Bot (Komis & Misirli, 2012b). L'objectif du scénario est d'entraîner les enfants à la programmation concernant le jouet programmable Bee-Bot et plus précisément aux notions de commandes de gestion et de fonctionnement (CLEAR (EFFACER) et GO (EXECUTER)), à la construction des séquences de commandes et à la réalisation des trajets dans l'espace.

Dans ce scénario les enfants doivent apprendre à construire des programmes, c'est-à-dire à piloter le Bee-Bot au moyen de commandes de direction et d'orientation. Les commandes peuvent être représentées graphiquement par des cartes associées aux boutons de commandes sur le dos du jouet programmable (Figure 1). Dans ce cadre, un programme concrétise un algorithme en tant que processus de résolution d'un problème. Un programme complet en Bee-Bot est un ensemble commençant par la commande CLEAR (pour vider la mémoire du jouet), comportant une séquence de commandes de direction et d'orientation (AVANCER, RECULER, TOURNER à DROITE, TOURNER à GAUCHE) et terminant par la commande GO (pour exécuter le programme).

Les enfants apprennent à contrôler, à manipuler et à programmer le Bee-Bot de manière organisée et construisent les concepts de direction et d'orientation (Avancer, Reculer, Tourner à droite, Tourner à gauche) durant l'estimation, la comparaison et la construction des trajets précis en se basant sur un système de référence qui est différent de leur corps. Cette approche se différencie des travaux de Greff (Greff, 1998) et de Demo (Demo et al., 2008), où le système de référence est le corps de l'enfant. Il s'agit, dans ce cas, d'une fonction cognitive de haut niveau.

La construction d'un algorithme comporte initialement la construction d'un modèle mental concernant un trajet de Bee-Bot dans l'espace. Plus précisément, les enfants doivent résoudre un problème dans

l'espace quadrillé de deux dimensions : ce problème consiste à mener le Bee-Bot vers un objet précis, c'est-à-dire à planifier le trajet convenable, à le verbaliser, à rédiger un programme par cartes de commandes et à l'appliquer avec le Bee-Bot, et, enfin, à corriger ce programme, si l'objectif n'est pas atteint.

L'objectif de notre travail a consisté à à étudier les processus de construction d'algorithmes et de programmes par les enfants de la maternelle et plus précisément les questions de construction des concepts préliminaires de programmation tels que la séquence et le débogage (debugging) ainsi que les modèles (patterns) de la programmation utilisés par les enfants pendant la construction de leurs algorithmes.

Description du processus

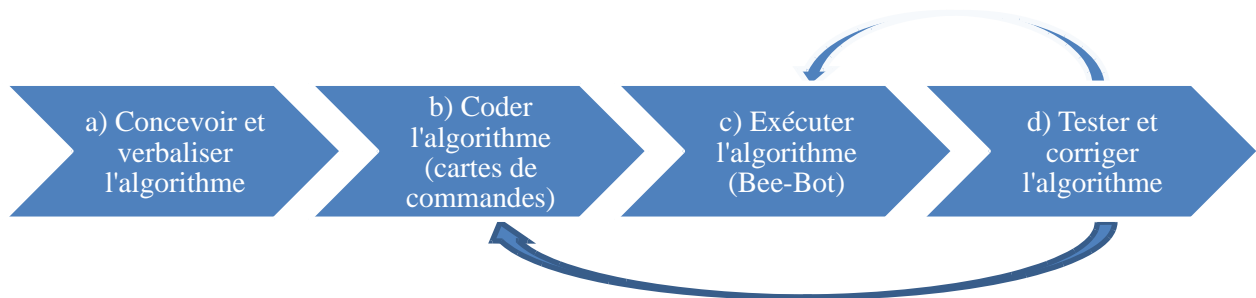


Figure 2 : Les étapes suivies par les enfants pour programmer le Bee-Bot

Les étapes suivies par les enfants pour programmer le Bee-Bot sont prévues dans le scénario éducatif (Komis et Misirli, 2012b). L'objectif du scénario consiste à construire un parcours dans le sol selon des consignes décrites explicitement par l'enseignante. Le contrat didactique du déroulement du scénario prévoit :

- a) une extériorisation du modèle mental à l'aide d'une verbalisation produite par l'enfant, qui décrit à haute voix le trajet à parcourir,
- b) une représentation graphique de l'algorithme à l'aide d'un ensemble de cartes qui représentent les commandes de programmation du Bee-Bot. Dans ce cas, un programme représenté par cartes de commandes est une structure séquentielle de type : VIDER LA MEMOIRE – INTRODUIRE COMMANDES – EXECUTION,
- c) l'introduction du programme déjà décrit de manière verbale et iconique dans le jouet programmable,
- d) l'exécution du programme et sa correction (debugging), si nécessaire, pour atteindre l'objectif du problème (figure 2).

Les données collectées sont de trois types : a) des entretiens individuels avant et après le déroulement du scénario concernant les représentations des enfants sur le Bee-Bot et ses commandes, b) des grilles d'évaluation sur lesquelles l'enseignante note le programme verbalisé, la construction du programme par des cartes et des commentaires concernant la programmation directe de Bee-Bot par chaque enfant, c) des vidéos du processus de programmation.

Présentation des résultats

La construction d'algorithmes séquentiels

Pour l'étude de la construction d'algorithmes séquentiels nous avons utilisé une série de variables qui nous permettent d'évaluer a) la capacité des enfants à planifier et à verbaliser un algorithme, b) la capacité des enfants à construire correctement des algorithmes séquentiels pour des problèmes qui ne

sont pas triviaux par rapport à leur âge et c) la capacité des enfants à gérer plusieurs commandes de programmation.

La capacité des enfants à concevoir et à verbaliser un algorithme est d'abord évaluée dans le cadre de notre étude. Nos résultats montrent que cette capacité est développée en grande partie. Plus de la moitié des enfants arrivent à verbaliser de manière correcte l'algorithme demandé par le problème.

Cette verbalisation se fait soit avec spécification (42%), c'est-à-dire avec une explicitation exacte et complète du processus de résolution, soit sans spécification (13%), c'est-à-dire sans expliciter de manière détaillée les différentes étapes de résolution. Dans ce cas, ils ne dénombrent pas par exemple les pas à effectuer par le Bee-Bot pour accomplir le trajet demandé ou ils ne précisent pas l'orientation du jouet. 31% des enfants verbalisent l'algorithme de manière semi-correcte, c'est-à-dire ils expriment la manière à construire le trajet demandé par le problème mais ils ne donnent pas de détails et ils font des erreurs de dénombrement de pas ou des erreurs d'orientation. Ces erreurs arrivent d'habitude dans la partie finale de l'algorithme. Il est à noter que seulement 6% des enfants verbalisent leur algorithme de manière non correcte. L'exécution de l'algorithme à l'aide de Bee-Bot aide par la suite certains de ces enfants à corriger leurs erreurs. La capacité des enfants à construire leurs algorithmes est également évaluée par l'examen des programmes construits à l'aide des cartes de programmation (figure 1). Dans ce cas, le 36% des enfants construit un programme correct en cartes de programmation. La différence par rapport à la verbalisation est due à l'absence de commandes de fonctionnement et de gestion (CLEAR et GO). Quand on verbalise on n'a pas vraiment besoin d'utiliser ces commandes tandis que quand on programme il faut les mettre.

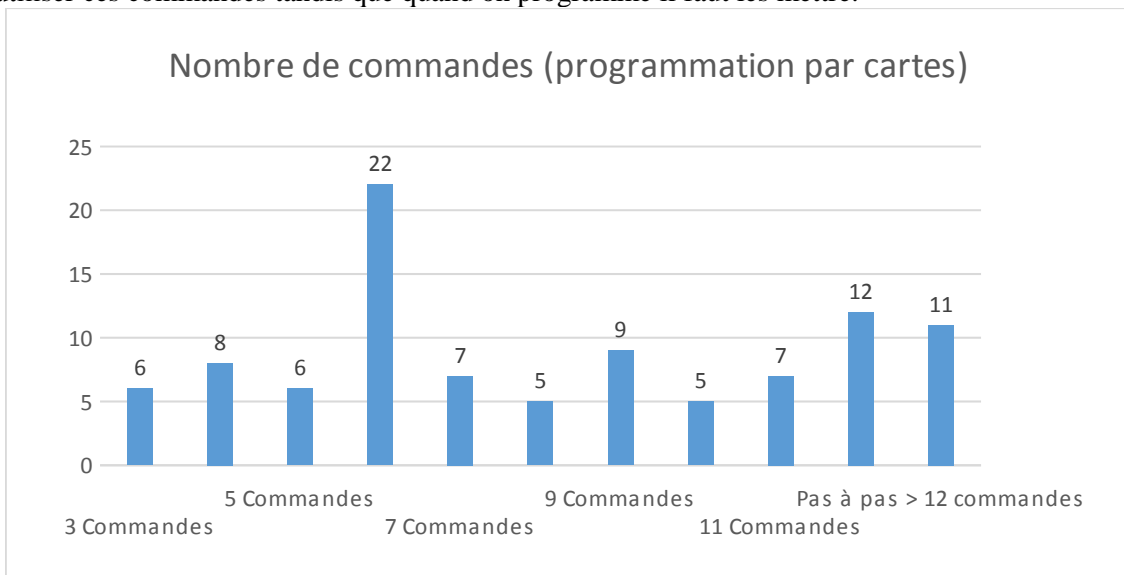


Figure 3 : Distribution du nombre de commandes pendant la programmation par cartes

La capacité des enfants à gérer plusieurs commandes de programmation est évaluée par le nombre de commandes (les cartes de programmation) dans un algorithme séquentiel. Dans ce cas, toutes les commandes utilisées sont comptées (commandes de gestion et de fonctionnement du programme et commandes de direction et d'orientation). Etant donné que les problèmes à résoudre sont ouverts, les solutions possibles sont différentes. Dans ce cas, nous tenons compte du dernier essai effectué par l'enfant. Dans la figure 3 nous trouvons la distribution du nombre de commandes. Dans le dernier cas (programmation pas à pas) les enfants utilisent plus de douze commandes car ils utilisent à plusieurs reprises des commandes CLEAR et GO.

Le débogage (debugging)

La variable concernant le débogage (*debugging*) mérite d'être étudiée de manière plus approfondie. Nous avons par conséquent identifié les différentes approches de correction des erreurs dans les programmes des enfants et nous les avons caractérisées en cinq catégories : débogage effectif (31%),

débogage non effectif (1%), programme correct non débogage (41%), programme non correct non débogage (18%) et enfants non exprimés (9%). Ces résultats montrent donc que une majorité (41%) d'enfants conçoit un programme correct du premier essai tandis que presque 31% des enfants procèdent à un débogage effectif après la rétroaction offerte par le Bee-Bot. Un seul enfant n'arrive pas à corriger son programme. Dix enfants n'ont pas pris part à la séance de l'évaluation et 19 enfants n'essayent pas à corriger leur programme bien qu'il ne fonctionne pas correctement.

Les stratégies de programmation

Une autre variable qui nous paraît intéressante concerne les stratégies ou modèles (patterns) de programmation. Cette variable décrit les stratégies utilisées par les enfants quand ils rédigent par cartes et exécutent leurs programmes (figure 4).

Le contrat didactique du scénario prévoit deux stratégies alternatives :

- a) la **stratégie Totale**, c'est-à-dire la rédaction du programme complet selon la règle *CLEAR* – ensemble des *commandes d'orientation et de direction (AVANCER, TOURNER, RECULER) – GO* et
- b) la **stratégie Pas à pas** selon la règle *CLEAR - une commande de direction ou d'orientation – GO, CLEAR - une commande de direction ou d'orientation - GO.*

Dans le premier cas, l'enfant rédige et introduit la totalité du programme dans le Bee-Bot et ensuite essaye son algorithme. C'est la stratégie que l'on mise à construire à l'aide du scénario. Dans le deuxième cas, l'enfant applique la plus simple stratégie de programmation : il introduit une commande à la fois et il teste le comportement du Bee-Bot. Cette stratégie correspond à la première stratégie d'apprentissage de programmation à l'aide du Bee-Bot.

L'analyse des données fait apparaître cinq autres stratégies :

- a) la **stratégie de Sous-Programme**, qui segmente le programme en sous-programmes constituant des sous-ensembles du programme reflétant une hiérarchie fonctionnelle : chaque sous-programme comprend un programme complet en programmation de Bee-Bot, c'est-à-dire un ensemble de commandes de direction et d'orientation entre la commande *CLEAR* et la commande *GO*.

Les trois stratégies suivantes combinent les précédentes :

- b) la **stratégie Totale et Pas à pas** (l'enfant commence par la stratégie Totale et il termine par la stratégie Pas à pas),
- c) la **stratégie Totale et Sous-Programme** (l'enfant commence par la stratégie Totale et il termine par la stratégie Sous-Programme) et
- d) la **stratégie Sous-Programme et Pas à pas** (l'enfant commence par la stratégie Sous-Programme et il termine par la stratégie Pas à pas).

Ces stratégies apparaissent durant le processus de la programmation en étant suite d'un débogage du programme.

Une dernière stratégie a été détectée lors de l'analyse des données : c'est

- e) la **stratégie par Essai et Erreur**. Dans ce cas, l'enfant n'a pas verbalisé l'algorithme et il n'a pas rédigé son programme par cartes de commandes. Il ne dispose donc pas une représentation mentale de l'algorithme et par conséquent il approche par essai et erreur pour trouver une solution.

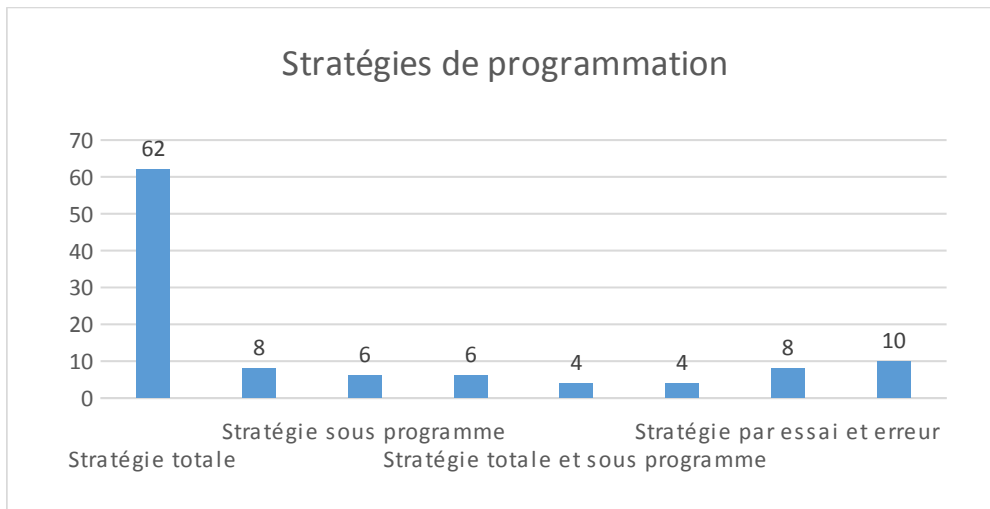


Figure 4 : Distribution des différentes stratégies de programmation

Dans la figure 4 nous trouvons la distribution des fréquences des différentes stratégies utilisées par les enfants. La stratégie Totale a la plus grande fréquence d'apparition (62 enfants). Toutes les autres stratégies apparaissent entre quatre et huit fois.

Discussion

Dans cet article nous avons analysé le processus avec lequel une centaine d'enfants de 4 à 6 ans conçoivent, planifient, verbalisent, exécutent et corrigent des algorithmes en utilisant le jouet programmable Bee-Bot. L'analyse effectuée a mis l'accent sur une série de questions concernant la construction des concepts préliminaires de programmation par les jeunes enfants et plus précisément aux questions de séquentialité, de débogage et de stratégies de programmation.

La question de séquentialité : les enfants de cet âge sont aptes à construire des algorithmes séquentiels. Après un enseignement structuré, la majorité arrive à concevoir, planifier, à verbaliser, à rédiger par cartes de commandes et à exécuter le programme à l'aide du jouet programmable. Les enfants construisent des algorithmes structurés disposant un relativement grand nombre de commandes (de six à douze).

La question de débogage (debugging) des programmes : le débogage, étant partie intégrante de la conception du scénario pédagogique, devient un processus habituel pendant la programmation du jouet programmable. La programmation tangible que le jouet programmable supporte, offre un cadre favorable pour tester et corriger les programmes. Nous avons remarqué à plusieurs reprises qu'après l'exécution du programme les enfants réalisent un retour au programme en cartes pour ne corriger que la commande erronée. De plus, tous les enfants qui ont effectué un débogage, sauf un, ont réussi à résoudre les erreurs détectées.

La question de stratégies de programmation développées par les enfants : les enfants mettent en route plusieurs stratégies de programmation. La plus fréquente est celle de la stratégie totale. C'est la stratégie incitée par le scénario pédagogique. La stratégie pas à pas, est celle qui permet de mener à fin la solution du problème à l'aide de la machine tangible. C'est également une stratégie enseignée dans le contexte du scénario. Les autres stratégies, bien que peu fréquentes, représentent un intérêt particulier étant donné qu'elles sont pratiquement inventées par les enfants dans leur effort de résoudre le problème. La plus intéressante est la stratégie de sous-programme qui permet de segmenter le problème en sous-problèmes et de donner ensuite une solution appropiée.

En conclusion, les enfants de quatre à six ans qui ont assisté dans la présente recherche sont capables à construire des programmes séquentiels à l'aide des commandes graphiques, à les transférer sur l'interface tangible du jouet programmable et à les corriger. Bien attendu, il faut tenir compte du

contexte de l'étude qui a eu lieu à l'aide d'un encadrement structuré par les enseignantes qui suivaient des approches didactiques appropriées.

Remerciements

Les auteurs tiennent à remercier Professeur Georges-Louis Baron pour ses précieux commentaires et son aide permanente. Ils remercient également Professeur Vassiliki Zogza, responsable scientifique du projet Fibonacci pour l'université de Patras pour son soutien continu. Ce projet a été supporté partiellement par le 7ème Programme-cadre de l'Union Européenne.

Références

- Bers, M. & Horn, M. (2010). Tangible programming in early childhood: Revisiting developmental assumptions through new technologies. In I. R. Berson & M. J. Berson (Eds.), *High-Tech Tots: Childhood in a Digital World* (p.49-69). North Carolina: Information Age Publishing.
- Demo, G.B., Marciano, G., Siega, S. (2008). Concrete Programming: Using Small Robots in Primary Schools. *Advanced Learning Technologies, ICALT '08*. Eighth IEEE International Conference on Advanced Learning Technologies, Santander - Cantabria , 1-5 July 2008, (p.301 – 302).
- Denis, B., Baron, G.L. (1994). Regards sur la robotique pédagogique. *Actes du quatrième colloque international sur la robotique pédagogique*. Paris, INRP : Technologies nouvelles et éducation.
- Depover, C., Karsenti, T., & Komis, V. (2007). *Enseigner avec les technologies*. Québec, Canada : Presses de l'Université du Québec.
- Greff, E. (1998), Le « jeu de l'enfant-robot » : une démarche et une réflexion en vue du développement de la pensée algorithmique chez les très jeunes enfants, *Revue Sciences et techniques éducatives* 5(1), 47-61.
- Komis, V. & Misirli A., (2011). Robotique pédagogique et concepts préliminaires de la programmation à l'école maternelle: une étude de cas basée sur le jouet programmable Bee-Bot. In *Sciences et technologies de l'information et de la communication en milieu éducatif: Analyse de pratiques et enjeux didactiques*. Actes du quatrième colloque international DIDAPRO 4, Dida&Stic, 24-26 octobre 2011, Université de Patras (p.271-284).
- Komis V., Misirli, A., (2012a). L'usage des jouets programmables à l'école maternelle : concevoir et utiliser des scénarios éducatifs de robotique pédagogique. *Skholê*, 17, 143-154.
- Komis, V., & Misirli, A. (2012b May 30). Jeux programmables de type Logo à l'école maternelle. *Portique adjectif.net*. Text. Retrieved June 25, 2013 from <http://www.adjectif.net/spip/spip.php?article140&lang=fr>
- Robert, F. (1985). L'utilisation de l'ordinateur dans l'enseignement primaire : l'exemple de la France. *Enfance*, 38(1), 19–30. doi:10.3406/enfan.1985.285.
- Yelland, N. (2007). *Shift to the future: rethinking learning with new technologies in education*. New York: Routledge.