## Original Paper

# Robotics Learning at Elementary School: Constructing

## Abstractions Using Multiple Instruments

Sandra Nogry<sup>1</sup> & Michel Spach<sup>2</sup>

<sup>1</sup> Paragraphe Lab, Cergy-Paris University, Site universitaire de Gennevilliers, Gennevilliers, France
<sup>2</sup> Laboratoire EDA, universit éde Paris, Paris, France

Received: February 11, 2022	Accepted: March 23, 2022	Online Published: April 22, 2022
doi:10.22158/fet.v5n2p36	URL: http://dx.doi.org/10.22158/fet.v5n2p36	

#### Abstract

This paper explores the robotics learning process at elementary school using a sociocultural approach. Its focus is on the activity of students aged 6-8 years during robotics learning sessions designed and implemented by two teachers. Drawing on two case studies, this research examines the functions of artefacts and semiotic representations during the learning process. While numerous artefacts are used, this question, well documented in the field of mathematics, is still little studied in the field of educational robotics. Microgenetic analyses highlight the multiple conversions triggered by the artefacts used in constructing the symbolic representations that make up the formal language to be acquired by students when learning how to program a sequence of instruction. The study highlights the embeddedness of the process of abstraction in this field.

#### Keywords

learning process, instruments, robotics, programming, elementary education

#### 1. Introduction

Computational thinking and programming skills have become critical for understanding the world around us, and there is clear evidence that developing these skills from the earliest age has significant benefits. As a result, computer science and programming have come to form an integral part of elementary school curricula in many countries. In this context, research on teaching and learning practices related to the development of programming skills at elementary school is crucial for understanding the specific features of these conversions and for designing appropriate scenarios.

Robotics is one of the most common tools used to introduce students to computer science, especially at elementary level (for a review, see Jung & Won, 2018). Popularized by Seymour Papert (1980) and his

Logo Turtle, educational robots with which learners are able to interact are known to be of particular value in capturing children's interest and curiosity. Using these robots, learners can observe the results of their actions and engage in creative activity, thereby promoting the development of a wide spectrum of skills (Benitti, 2012; Tho et al., 2016).

The range of programmable toys and educational robots available in the market has developed considerably in recent years. With the widespread availability of robots specifically designed for children comes a need for a specifically tailored pedagogy (Alimisis, 2012; Fluckiger, 2019). Many different pedagogical scenarios and teaching resources have been proposed in this area. More recently, there has been growing interest in integrating robots into elementary school curricula (Sullivan & Bers, 2016; Elkin, Sullivan, & Bers, 2018; see Jung & Won, 2018 for a review). In the different scenarios proposed by researchers or designed by teachers themselves, a wide range of artefacts and inscriptions are often used to help students learn how to program robots. Current studies often consider the robot as a source of feedback, and other artefacts as information to process (Chevalier et al., 2022).

In most studies, the role played by the bodily interactions with artifacts in the learning process is often neglected. If their role is well documented in the field of mathematics (e.g., Lave, 1988; Hershkowitz, Schwarz, & Dreyfus, 2001; Noss, Hoyles, & Pozzi, 2002; Radford, 2009, 2014), it is still little studied in educational robotics. Sociocultural approaches and embodied cognition offer theorical and methodological framework to study this question in this field (Jung & Won, 2018; Kopcha, Ocak, & Qian, 2021). Kopcha, et al. (2021) studied the embodiment of computational thinking analyzing educational robotics activities. However, few studies have examined the functions of the artefacts used during educational robotic learning process. These roles remain to be investigated.

This research project aims to examine the introduction and use of educational robots in class by teachers using resources made available to them (textbooks, scenarios published online, etc.). The purpose of this study is to analyze the instruments that contribute to conceptual development: how are these instruments formed and appropriated by students? What roles do they perform? How do they contribute to the process of conceptualization?

To investigate these questions, two case studies conducted in elementary schools (first and second grades) are presented. In each scenario, a simple programmable robot and other artefacts were used to introduce the same computational concept. The analysis highlights the roles played by these artefacts in the robotic learning process.

#### 1.1 Educational Robotics at Elementary School

Programming was first introduced at elementary level in the 1980s following research by Papert (1980) that led to the introduction of LOGO Language (for example, Michayluk, 1986; Hoc, Green, Samur çay & Geelmore, 1990), a project that was subsequently abandoned. Having been recognized as key requirements of modern society, computer science and programming have been reintroduced into elementary curricula in the last decade, the aim being to provide children with the necessary tools for

understanding the processing operations performed by the systems they use (for an overview, see Baron & Drot-Delange, 2016). The point is to introduce children to basic computational concepts while also promoting the development of computational thinking (Wing, 2006). Brennan and Resnick (2012) distinguished three aspects of computational thinking: computational concepts (sequences, loops, parallelism, conditional structures, operators, data, etc.), computational practices (incremental and iterative, testing and debugging, reusing and remixing, and abstracting and modularizing, etc.), and the way in which programmers describe themselves and how they perceive their relationship to others and to the technological world around them. In this perspective, robotics requires students to decompose a larger task, develop potential solutions, apply mathematical concepts as they program the robot, and debug problems as they negotiate the challenge at hand (Bers et al., 2014; Chen et al., 2017; Kopcha et al., 2017).

Bers (2019) proposed another approach of robot programming learning, viewed not simply as a means of developing problem-solving skills but also as a way of learning a new language—in other words, a system of symbolic representations with its own grammar and syntax capable of being used to express ideas. The evolution of languages and the development of programming environments tailored to children (scratch, scratchJR, programming kits, robots, etc.) promote such learning from the youngest age. Even at preschool level, teachers can resort to pedagogical scenarios involving the use of simple programmable toys, allowing students to learn how to control their actions using a specific language (Bers et al., 2014; Komis & Misirli, 2015; Komis, Romero, & Misirli, 2017) and to develop their own projects (Bers et al., 2014). Through this learning process, students develop a basic understanding of the concept of sequence and learn how to plan, to manage several programming commands, to perform an algorithm, and to debug. Scenarios such as these also help to promote the development of social skills necessary for peer collaboration.

Beyond research conducted at preschool level, studies have also shown that educational robotics supports the development of cognitive skills, including spatial skills, and promotes greater flexibility in applying abstract rules. A degree of abstraction is needed to program a robot, requiring abstract representations, temporal references, and an understanding of causal mechanisms—all of which young children are in the process of developing (Crahay, 1987; Gaudiello & Zibetti, 2013). Educational robotics also contributes to the development of a wide range of concepts (Benitti, 2012; Strawhacker & Bers, 2019) and language and social skills (Tho et al., 2016). Studies in this area have also emphasized the crucial role of support from teachers and parents, which is known to be a key factor in learning (Tho et al., 2016; Benitti, 2012).

#### 1.2 The Learning Processes Involved in Programming Using Robots

How do children develop from first handling programmable toys—i.e. devices with which they are not familiar—to mastering the principles of programming? What are the processes at work in such learning?

A small number of studies have focused specifically on robotics learning processes (for a summary, see Jung & Won, 2018) and have demonstrated that early conceptions of IT tend to be anthropomorphic (Duboulay, 1986, cited by Rogalski, 2015; Levy & Mioduser, 2008). In other words, from a young observer's point of view, a machine considers intentions, semantically processes the operations at work, and appears to be endowed with human skills and capabilities. Understanding how a robot actually works is, therefore, a gradual incremental process. For example, Mioduser et al. (2009) examined the processes involved in abstracting the rules underlying a robot's emergent behaviors among young children. Their study found that children initially focus on the robot's action sequences before becoming aware of repeated sequences triggered by occasional events (script). Once the robot's actions and the environmental conditions are viewed as being of similar importance, the children are then able to observe their covariations, a process necessary for abstracting temporal rules. The embeddedness of abstract rules in concrete objects and events ("concrete abstraction"), mediated by the teacher, supports children's reasoning. During this learning process, their representations become increasingly more general and complex, moving from temporal to atemporal descriptions, with their attentional focus on the robot gradually expanding to include the immediate environment.

Beginners also experience difficulties with the deferred nature of program execution (Rogalski, 2015). It is difficult for them to make the transition from "doing" to "getting to do", in other words to adopt the point of view of the robot, which is required to follow the program's instructions. A strategy typically used by children involves manipulating robots directly. In doing so, children are able to imagine how the robot will move before anticipating the control commands (Mioduser & Levy, 2010).

Kalas, Blaho and Moravcik (2018) identified different levels of control: direct manipulation, which involves moving the robot manually, indirect manipulation, characterized by the identification of the goal to be achieved, direct control, in which each movement is programmed and immediately executed ('one button-one step'), and computational control, in which all the steps of the itinerary are programmed before executing the program. The authors proposed a learning environment designed to enable students to progress from one level of control to another by gradually increasing the distance between the object to be controlled and the person controlling it (whether physically or symbolically).

Other strategies have been observed in the implementation of a pedagogical scenario among children aged 4-6 years (Komis & Misirli, 2015). In this case, the students used different strategies to program the Bee-Bot's movements, including a "step-by-step" strategy (direct control) and a "total" strategy (implementation of the complete sequence, i.e., computational control), but also a "subprogram" strategy (segmentation of the itinerary into program sub-sets) and mixed strategies, observed mainly

during debugging sessions. It should be noted that in this pedagogical scenario, the 'total' strategy was the most commonly used approach. Its implementation was supported by the teachers' interventions and by coding command cards used to represent the programmable instructions. Through these cards, different symbolic representations are introduced to represent the instructions, enabling learners to develop a new language. What role do these cards play in the learning process during programming? More broadly, what roles play artefacts and inscriptions used by children during the programming learning process?

#### 1.3 Learning and Abstraction Based on Sociocultural Approaches

Recent trends in the computational thinking literature suggest that computational thinking consist in a complex, dynamic interaction among person, technology and environment (Berland & Wilensky, 2015; Yasar, 2018). In this field, as in others fields of sciences learning, an embodied perspective has been viewed as a way to improve teaching and learning because it engages all aspects of the learner when understanding abstract concepts (Abrahamson & Lindgren, 2014; Black et al., 2012; Hall & Nemirovsky, 2012; Sung et al., 2017; Wu & Cheng, 2021). Designing learning environments that effectively create the conditions for embodied learning has become an area of research and design (Lindgren et al., 2016; Marougkas et al., 2021; Li & Zhang, 2021).

In school context, there has been an increasing interest in exploring how children might embody computational thinking (Brennan & Resnick, 2012; Grover & Pea, 2013; Kopcha et al., 2021; Odgaard, 2022). Some of these studies analyse the way that children coordinate their physical activity with the robot and structures in the environment during educational robotics learning process. For instance, in Kopcha et al. (2021) study, the participants anticipated robot movement by acting as if they, themselves, were the robot when interacting with each other, the map, and the computer program. In this context, *"the robot became an object that helped ground and align the participant's knowledge and thinking with their own bodily experience*" (Kopcha et al., 2021, p. 1996)

In these studies, in-depth analyses of the functions of gestures and body movements have been realised. However, the functions ensured by the manipulated artefacts, whether semiotic or material, have been poorly investigated. Considering computational thinking as an embodied and embedded phenomenon, our question is: what functions perform these artefacts during the learning process?

As noted by Jung and Won (2018), sociocultural approaches offer useful theoretical and methodological frameworks for examining the embeddedness of robotics learning process. Most of the researches on embeddedness and learning have been conducted in the area of mathematics (e.g., Lave, 1988; Noss, Hoyles, & Pozzi, 2002; Radford, 2009, 2014). As pointed out by Radford, "concepts are continuously transformed by the activity through which they appear in their sensuous and material form (...) they are mediated entities and their mediation is concrete human practice" (2014, p. 353). In this perspective, artefacts used as mediation during activity are constitutive of conceptual development.

The mediation functions provided by the artefacts (whether semiotic or material) have been well studied by cultural and historical activity theories (Kaptelinin & Nardi, 2006). Their Shape, matter and structure reflect the experience of people who produced and used them; Created and transformed during activity, they are an accumulation and a transmission of social knowledge. Associated with gestures they become instruments, mediators of purposeful human actions.

The instrumental approach (Rabardel, 1995; Rabardel, 2001; Rabardel & Bourmaud, 2003) aims to examine the process of instrumental genesis, in other words, how, during activity, an artefact becomes an instrument. The instrument consists of an artefact component (an artefact, a fraction of an artefact or a group of artefacts) and a scheme component (defined as an unvarying structure of the action performed to achieve a specific objective). Activity is mediated by a set of instruments that gradually form a system comprising a structure, redundancies, etc. (Rabardel & Bourmaud, 2003). This approach has been adopted to study the children activity in school context in order to design environments supporting child development (Decortis, 2015; Gourlet & Decortis, 2018; Forcisi & Decortis, 2018). In this perspective, better understanding the functions of artefacts during the learning process requires an instrumental genesis analysis; it consists in highlighting the schemes associated to the artefacts used during the activity, their change, and then analyzing the relationships between the multiple instruments organized in system during the activity.

Besides, semiotic perspective has highlighted the role of the transformations operated on the multiple semiotic representations used as mediations during the learning process (Saenz-Ludlow & Presmeg, 20106). In this view, mathematical objects can only be accessed through these semiotic representations (equation systems, graphic representations, etc.). Learning new mathematical concepts involves the coordination of different semiotic registers. They constitute a set of resources, constraints, and indications for performing the required task. The conceptualization process is underpinned by conversions from one register to another that enable the construction of new meanings (Duval, 2017). The body, physical gestures, and material artefacts contribute to this process; they play a key role in the development of abstract representations when learning arithmetic (Radford, 2009).

To summarize, in this view, studying the embeddedness of programming learning process involves understanding the mechanisms through which learners use, (re)-create and appropriate artefacts (whether semiotic or material) during the learning process, by analysing the instrumental genesis, role they play as mediation, and the multiples transformations performed with/on them by students to construct abstract representations of the concepts required for programming.

#### 2. Presentation of the Studies

The two studies were carried out as part of an ANR project devoted to understanding the design and implementation of pedagogical scenarios by teachers in France. Several teachers volunteered to take part in the project having opted to use robots with the aim of introducing their students to programming. This paper presents two case studies focusing on the analysis of two pedagogical scenarios designed by teachers using the same programmable toys (Bee-Bots) in their CP and CE1 classes (years 1 and 2 of elementary school).

The aim was to examine the activity of students in specific situations proposed by their teachers with a view to analyzing the learning processes involved in developing the concept of *sequence of instruction in* a socio-cultural perspective. In other words, the question raised was: What roles do the artefacts and semiotic representations given to students play during this learning process?

In both case studies, students' activity was analyzed through observation, recordings and interviews based on the principles of the instrumental approach (Rabardel, 1995; Rabardel, 2001; Rabardel & Bourmaud, 2003). In this approach, the unit of analysis is the situation of *instrumented activity*. The activity involves three elements: the acting *subject*, the *object* toward which the activity is directed, and the *instrument*, composed by an artefact associated with a scheme (defined as an invariant organization of action). The aim is to document the objectives of the activities undertaken by students, the sequence and development of their activity, and the mediating roles played by the instruments involved in achieving the objective as well as their genesis.

Since children interact with their environment through movements, gestures and drawings, and insofar as the robotics learning process is based on tangible interaction with technological environments, different types of data need to be collected during the robotics learning process (Jung & Won, 2018). Therefore, a decision was made to combine the analysis of student activity with a microgenetic analysis of conceptual development (Ackermann, 2013; Parnafes, 2013; Di Sessa, 2014). The microgenetic approach is a clinical approach designed to reveal knowledge as it develops at different temporal scales. The aim is to understand how children engaged in an activity explore a phenomenon, become aware of changes, think, move, "stumble", and come to revise their conceptions (Ackermann, 2013). Parnafes and Di Sessa used this approach to examine conceptual development in physics. According to them, the approach involves several key principles:

- (1) selection of episodes of activity leading to substantial changes in the subjects.
- (2) fine-grained observation of each episode.
- (3) intensive and opportunistic use of all available data to make detailed inferences about the changes performed and their mechanisms.

The microgenetic approach can be applied in ecological situations, thereby incorporating the inherent embeddedness of knowledge development.

2.1 In combining the two approaches, the aim is to show how the instrumental geneses associated with the artefacts used contribute to the conceptual geneses that occur during the learning situation. *Method* The two case studies focused on two pedagogical scenarios designed and implemented by two teachers in their respective classes (CP and CE1) located in the Paris region (France).

The programmable toys chosen by the teachers were Bee-Bots. In their scenarios, they opted for a method of investigation based on an inductive approach, which involved discovery of the robot by the students followed by a gradual structuring of the activity. The objective was for the students to learn how to program the robot to travel along a predefined route. To do so, the students had to develop a range of skills, including being able to use the controls of the programmable device, understanding what an instruction sequence is, designing and building an increasingly complex instruction sequence, and debugging. During the implementation of the scenario, the students' activity was observed and filmed following the protocol presented below.

#### 2.1.1 The Programmable Toys Used

The devices used, known as Bee-Bots, are small robots whose behavior can be programmed using a tangible interface located on top of the device. Figure 1 shows the console with the four buttons used to guide and direct the robot's movements, while three other buttons are used to control the execution of the program ("Go": execution of programmed instructions, "Pause": momentary interruption of the execution of the program, "Clear": deletion of the commands recorded in the memory).



Figure 1. Bee-Bot: A Programmable Device

#### 2.1.2 Observation Protocol

The two case studies focused on two pedagogical scenarios designed and implemented by two teachers in their respective classes (CP and CE1), in urban schools, located in the suburb of Paris (France). The teachers were highly experienced (having been in the profession for more than 10 years) and both were teacher trainers.

The first case study focused on a scenario implemented in a CP class with 18 students (10 girls, 8 boys) aged 6 to 7 years who were using educational robots for the first time.

The scenario involved six sessions, all of which were observed and filmed. Video recording was performed using a fixed camera (showing a wide shot of the room) and a mobile camera focused on the work of each group in turn. In total, four hours of activity were recorded and analyzed.

The second case study focused on a scenario implemented in a CE1 class with 23 students (12 girls, 11 boys) aged 7 to 8 years who were using educational robots for the first time. In this case, the scenario involved fifteen sessions, eleven of which were observed and filmed using a protocol similar to the protocol applied in the first case study. In total, eight hours of activity were recorded and analyzed.

2.1.3 Analysis of Student Activity

The analysis of classroom activity was performed using video recordings based on the instrumental approach (Rabardel, 1995). The point was to document the object of the activity undertaken by the students being filmed, the sequence and development of the activity, and the instruments used by students. The analysis aims to highlight the process of instrumental genesis—in other words, how the artefact becomes an instrument—by documenting the changes made to the artefact (instrumentalization) and the formation of schemes associated with the artefact (instrumentation) (Rabardel, 1995).

Alongside this, a microgenetic analysis of the development of knowledge was also performed (Parnafes, 2013; Di Sessa, 2014). The microgenetic approach involves examining episodes during which skills, changes in knowledge or conceptual developments are identified and highlighted.

Thus, for each phase of the pedagogical scenario, a three-step analysis of student activity was carried out. The three stages were:

- Selection of episodes involving changes.

- Analysis of the student instrumented activity during the selected episodes to highlight instrumental mediations.

- Characterization of changes during these episodes (changes in semiotic representations, instrumental and genetic geneses).

In particular, the aim was to document the range of semiotic representations and artefacts used during the activity, to identify their roles, to describe the formation of new schemes associated with these artefacts (instrumental genesis), and to identify indicators of change and development.

For each of the two case studies, we selected episodes during which artefacts are used, analyzing their functions, the schemes associated, and their role during the learning process.

#### **3. Findings**

#### 3.1 Case Study 1: Programming Movements in Space in a CP Class

The focus of the sequence observed in CP was to program movements in space (Note 1). The 18 students were split into groups of three to use the six available *Bee-Bots*. The sequence involved two phases:

(1)The first phase involved getting the robot to travel along an itinerary traced by students on the floor (sessions 1 to 4): the aim was to introduce the Bee-Bot's control commands. Each group of students was required to "*trace an itinerary*" on the floor before learning how to use the robot by themselves by getting it to travel along the planned route.

(2)The second phase required the students to work in groups to trace an itinerary to be programmed (sessions 5 and 6): the aim was for the students to program a sequence of instructions. Each group was required to trace an itinerary on paper and then to program the robot to complete the group's itinerary and then the itinerary programmed by another group.

3.1.1 Phase 1. Learning about the Control Commands—Construction and Deconstruction of an Initial Scheme

An analysis of the students' activity during the first phase (i.e., getting the robot to complete an itinerary) shows that an initial instrumental genesis soon developed, with the association of a particular scheme (unchanging structure of action) with the robot: *press Go and then the forward arrow* (which we will refer to as the "*Go-Arrow*" scheme) (Figure 2).



The student presses Go

The student presses the Arrow

The robot completes a longer than expected itinerary, causing surprise and emotion

### Figure 2. Illustration of the "*Go-Arrow*" Scheme: Action Sequence Frequently Performed to Program the Robot's Movements

All the students adopted this scheme—a scheme shared collectively in class during a group session and reflecting an attempt at direct control. In doing so, the students appeared to adopt an anthropomorphic conception of the robot whereby the machine semantically processes the operations at work (Rogalski, 2015). The underlying scheme may be interpreted as a "go—forward/back/etc." instruction. This may be viewed as a failure to understand the control commands and the deferred nature of the execution of the program instructions. The *Go* command executes the instruction sequence previously implemented. During the first phase, the teacher's questions gradually resulted in the students deconstructing this scheme by questioning the cause-and-effect relationships between the selected buttons and the robot's

behaviour and by ascribing a particular meaning to the different control commands (Nogry, 2019) (Note 2). Following session 4 (out of 6), the teachers introduced a new action sequence to the group as a whole: "*reset by using the "Clear" button, press the arrows, press Go*". This then becomes a substitute for the initial scheme.

3.1.2 Phase 2. Deferring the Programming of the Robot by Drawing a Plan

In the next phase, the teacher set two separate tasks: *draw a simple itinerary* followed by *program the robot*. The programming of the robot was thus deferred to a second phase. The steps proposed in the second task resembled the approach taken by computer science engineers—i.e., planning, implementation of the algorithm, assessment, and debugging.

a) From drawing to schematizing the itinerary

The first phase involved drawing the robot's itinerary on paper. An examination of the drawings produced shows that some of the initial drawings were not on task. In other words, some of them did not include the information needed to direct the programming, and their shape (curves, acute angles) was not compatible with the robot's range of movements since the robot was only able to perform right-angle turns.

The teacher then provided more specific instructions by asking the students to draw "*a simple itinerary composed of at least 6 steps*" and to represent the robot's steps as arrows (Nogry, 2019). The teacher's instructions resulted in a conversion of the groups' representations, with conventions developed collectively in class being gradually incorporated (e.g., right-angle turns, arrow representing a step, etc). The degree to which the students were able to appropriate these conventions varied (Figure 3).



Figure 3. Drawings Incorporating the Conventions Introduced by the Teacher

#### b) Multiple instruments used in programming

In carrying out the second task (i.e. programming the robot to perform the planned itinerary), the initial drawing was used as a programming instrument. It was found to perform a range of mediations during activity and to be the subject of successive conversions, understood as constituent elements of the development of the sequence of instructions. These roles are presented below.

- The drawing and the body: instruments for planning the robot's actions

In each group, the scheme, associated with an action sequence that consisted in counting the number of steps and identifying the rotations to be performed (Figure 4), served as a tool for planning the process of programming the robot. An initial conversion process occurred at this point, with the move from a graphic representation to a synthetic verbal description of the itinerary.









The student refers to the planned (drawn) itinerary

The student then counts the number of arrows and ticks them off The student then picks up the robot to program it and considers the direction of rotation

S1: turn right or left?

The student represents the robot's rotation by moving her body, with accompanying commentary. S2: right S1: left, if you go like that it's left

### Figure 4. Planning of the Program to Be Carried out by a Student: An Action Sequence Performed

Interestingly, in planning the rotation, some students felt the need to resort to a bodily representation: in other words, by embodying the rotation in order to mimic the movements to be performed by the robot, the students engaged in another conversion process, moving from an allocentric representation of space (the scheme) to an egocentric representation.

During planning, the students encountered difficulties due to the discrepancy between the schematic representation (with the curved arrows representing corners) and the Bee-Bot's control commands (forward/back arrows controlling one-step movements, the right and left arrows controlling rotation on the robot's axis). The discrepancy between the codes associated with the plan and the robot initially caused many counting mistakes and errors in selecting the correct direction.

- The plan as an artefact for implementing the program

The plan was then used to implement the program (selection of buttons). Using the plan, a member of the group would indicate to the student tasked with programming the robot which button to press and how many times.

Because of the difficulty of anticipating the arrow to choose for performing rotations, some of the students adopted a subprogram-type strategy (Komis & Misirli, 2015): in other words, the complete itinerary was split into subprograms, with the first subprogram corresponding to a straight line, followed by another subprogram controlling the rotation step by step (with a single instruction between the Clear and Go commands), followed by another sub-program. In this case, the planning phase alternated with the implementation phase. The students gradually incorporated rotations, thus tending toward a "total strategy" (Komis & Misirli, 2015) geared toward executing a single program for the entire itinerary.

The plan and the robot as artefacts for assessing the program

The plan played a role, lastly, in the assessment phase of the program following implementation. This phase, initially carried out under the teacher's guidance, involved measuring the discrepancy between the route travelled by the Bee-Bot and the planned itinerary. Combined with deictic gestures and comments, the plan served as an instrument for matching the planned route with the robot's actual movements (Figure 5).



(a) The teacher and the (b) Discrepancy observed (c) Analysis of (d) Analysis of the problem student count the steps by between the planned and the problem and search for a solution matching the plan against actual itinerary (different the robot's direction) actual movements Teacher & student: one (.) Teacher: what went wrong? Teacher: what Student: it should do this and two (.) three (.) four (.) Student: it went that way happened? Was then this five \*pointing to the left\* \*moves the robot along the it here that it \*the teacher points to the went wrong? route shown on the plan\* corresponding arrow\*

\*points to the Teacher: what instructions do corresponding you need to give it to make it arrow\* do that?

## Figure 5. Assessment of the Program by a Group of Students: Matching of the Planned Itinerary against the Robot's Actual Movements under the Teacher's Guidance

Published by SCHOLINK INC.

In this phase, the teacher's interventions played a key role in helping the students to work out the causes of the discrepancy (Figure 5, picture (b)) and encouraging them to modify the program or the representation of the itinerary in cases where it lacked precision.

By the end of the session, all the students were able to program a simple (triangular) route by programming a complete sequence or two subprograms (forward motion until rotation, forward motion following rotation).

c) Geneses during the process

During the learning situation, the students shifted from a focus on the outcome of the process (i.e., the robot's targeted destination) to a focus on the process implemented to achieve the outcome through the introduction of an intermediate phase: schematization. In the second phase (sessions 5 and 6), the structure of the task and the proposed mediations resulted in the students resorting systematically to an approach that involved planning the robot's movements, thinking about the instruction sequence to be coded, implementing the program, and observing the discrepancy between the planned itinerary and the actual route performed. In doing so, the students gained an insight into the deferred nature of the execution of the program.

By performing conversions between different representations of the itinerary (sketched plan, verbalization and bodily representation, or enactment, of the rotations, selection of instructions on the control console), the students also developed the ability to anticipate the sequence to be programmed and to implement increasingly elaborate action sequences. These conversions were governed by conventions proposed by the teacher. The concept of sequence or program was never verbalized by the teacher but emerged during the activity as a concept in action.

3.2 Case Study 2: A Multiplicity of Artefacts Contributing to the Construction of the Concept of Sequence

In CE1, the teacher initially introduced the Bee-Bots with the simple aim of getting the students to program movements on a grid. Having realized that the students felt the need to materialize the robot's itinerary in order to picture it "in their mind", the teacher introduced several artefacts in the following sessions, including a large board, a deck of cards, a small board, and a strip of paper representing the algorithm.

Drawing on the observations performed, this section describes the various instrumental geneses associated with the artefacts gradually introduced during the sessions in the different phases of the programming activity, together with their roles. The aim is to highlight the different conversions and transformations that they allow for and to determine how they contribute to conceptual development.

3.2.1 Large Board: An Artefact Proposed by the Teacher to Regulate the Robot's Movements

The students were given a large grid-patterned board consisting of squares corresponding to the length of the robot's unit of movement (i.e., one step). The aim was to facilitate the process of breaking the robot's movements down into steps. The teacher asked the students to observe the Bee-Bot's movements and encouraged them to plan its movements on the board. To facilitate the planning process, the start and finish squares were materialized using color cards positioned on the board (Figure 6).



Figure 6. Large Board Used to Regulate the Robot's Movements

Having initially been encouraged to imagine the Bee-Bot's itinerary in their mind on the grid pattern, the students soon came to materialize the robot's movements by moving the Bee-Bot manually from square to square (control by direct manipulation). In the following sessions, the teacher gradually introduced other artefacts to help the students anticipate the Bee-Bot's movements.

3.2.2 Coding Cards as a Mediating Tool for Designing an Algorithm

Having observed that some of the students had struggled to memorize and communicate about the itinerary during the second session, the teacher suggested using cards (Note 3) to help them map an itinerary from start to finish without having to manipulate it (

Table 1).





Droite	Right
Commencer	Start
Vider	Clear

Guided by the teacher's instructions, the students used the tool in two separate phases of the activity: the design of the itinerary (during which the students placed the cards on the large board in order to model the planned itinerary; Figure 7), following by the implementation of the program (during which the students read the graphic instructions shown on the cards and pressed the relevant command on the robot's console). In doing so, the students were able to convert the itinerary planned using the grid into instructions represented by the cards and, at a later stage, with the symbols represented on the robot's tangible interface.

The students encountered various difficulties in both phases of the activity. In the first phase (the design of the itinerary), the students struggled with the direction of the cards since they tended to rotate them, assigning meaning to both the cards and the rotation movements (Figure 7). The symbolic representations shown on the cards gave the students the illusion of being able to perform the planned itinerary by carefully positioning the cards in such a way that they were adjacent to each other, even enabling connections.



Figure 7. Planning of the Itinerary with Rotation of the "Go" Card by a Quarter Rotation to the Right

In the second phase (in other words, the phase corresponding to the implementation of the program), the students encountered difficulties by incorrectly representing the meaning of the cards, assigning to the "turn left" and "turn right" cards, in addition to rotation, a single-square forward movement action. Their incorrect representation was compounded by the fact that the dimensions of each card were identical to the dimensions of each square, causing the students to place only one card on each square.

Having noted that the students were employing a "step-by-step" strategy, the teacher designed two new tools intended for different phases of the activity: a small board designed to facilitate the design of the planned itinerary and a strip of paper designed to support the programming of the robot.

3.2.3 The Small Board as an Artefact for Representing the Itinerary based on Shared Norms

In the third session, the students were given a grid-patterned paper board of smaller dimensions resembling the large board placed on their desk ("*you've got your little plan now. You're going to use it to trace the itinerary; it's a miniature version of the itinerary*"). During the session, the teacher emphasized two main functions of the artefact: memorization of the itinerary and communication within the group. By using this tool, the students were able to agree within their group on the planned itinerary by representing it using an erasable marker. The students took ownership of the tool by representing the itinerary in different ways (Figure 8).



(a) Succession of straight (b) Continuous curved line (c) Succession of arrows segments

#### Figure 8. Itineraries Traced on the Small Board

Having been the initial focus of group activity, the itinerary traced on the small board then served as a tool for guiding the process of programming the robot (Figure 8, picture (a); a conversion is performed, and the itinerary drawn is verbalized).

The students struggled with spatial positioning and the direction of the small board relative to the large board, a problem which the teacher was able to solve by getting the students to project themselves onto the robot, saying: "the robot's your eyes" (Figure 9, pictures (b) and (c)).



Figure 9. Programming Using the Small Board and Looking for Errors with the Teacher

By using these artefacts, the students learned to embed an itinerary in both a delimited physical space (the large board) and a figurative space (the small board). As a result, they gradually came to understand the connection between an actual step and a grid-pattern step by using the squared surface to standardize the robot's movement in terms of distance (number of squares) and direction. In doing so, the students were able to see that the itinerary was relative to initial conditions (square one).

3.2.4 The Algorithmic Strip as an Artefact for Supporting the Design of the Algorithm

a) A guided instrumental genesis

The students encountered difficulties in developing their program caused by the fact that the programming content entered into the robot's console was not visible. To overcome these difficulties, the teacher introduced a new artefact in the next session: a strip of paper showing the algorithm. The tool was presented as a support for memorizing the implemented program and correcting errors: "to check things, you need to be able to see them. If there's a mistake, the strip will help you to see it. That's how you'll realize there's a mistake." The students were therefore given a laminated strip designed to make the program visible and presentable.

The strip also helped to structure the process of writing and reading the algorithm. The use of squares helped to improve the granularity of the algorithm, with the grid pattern suggesting that the instructions were sequential. Since the strip was laminated, the routes drawn could be partially or fully erased using the marker.

To help students take ownership of the artefact, the teacher then specified the sequence of actions to be performed with the various artefacts: "trace the itinerary on the small grid-patterned board, then note it

on the algorithmic strip to keep a trace of it, then program the robot using the strip". The use made of the various tools was also discussed at the end of the session ("What difficulties did you encounter?").



Figure 10. Coding of the Itinerary Using the Digital Strip

#### b) Writing of the instruction sequence

The digital strip was first used in a group setting with one student using the "small board" to control the direction of the robot ("turn, go forward") and another student coding the robot's movements using a symbolic representation similar to the representation found on the robot's console (Figure 10). The students were able to change the direction of the "small board" in such a way that the itinerary and the student's body pointed in the same direction when designing the instruction sequence. In doing so, the students moved from an allocentric representation of the itinerary to an egocentric representation consisting of an ordered sequence of steps represented graphically on the algorithmic strip.



Figure 11. Verbalizing the Instruction Sequence

Second, the small board and the large board were used to check that the itinerary on the small board and the digital strip matched (Figure 10, picture (b)).

The digital strip was then used to program the robot by verbalizing the instruction sequence written on it prior to implementing it (Figure 11). The first strategy involved verbalizing the program step by step (i.e., "go, go, go, turn right, go, go"). The second strategy involved breaking the itinerary down into several sections and then counting the squares to be traversed in each section before verbalizing them ("go forward three squares, turn right, go forward two"). This last strategy is similar to procedural programming in which program blocks are identified to form procedures. Through the instructions and tools provided, a "total" programming strategy gradually came to replace a "step-by-step" strategy.

In other words, the program corresponded to a sequence consisting of an ordered series of steps represented graphically on the algorithmic strip. By erasing its content partially or totally, the strip was used to develop the algorithm.

#### c) An artefact for testing the algorithm

Combined with deictic gestures, the algorithmic strip became an instrument for testing the validity of the algorithm. Students had to determine the degree of consistency between the number of squares traversed on the small board and the number of program steps on the strip. This process led to a number of difficulties. The first difficulty involved the notion of interval, with the students counting the squares forming the itinerary rather than the squares through which the itinerary passed. The second difficulty related to the instrumentation of the algorithmic strip: when the same square contained two instructions, only one instruction was counted. The teacher provided support during this matching process (Figure 12).



Figure 12. Combined Reading of the Itinerary and the Program

#### d) Debugging

In the event of an error being identified during movement, all or part of the instructions shown on the digital strip were erased or modified. Three different procedures were used to develop the algorithm. In

the first case, the algorithm written on the strip was completely erased (complete erasure) and then corrected and rewritten. In the second case, the algorithm was partly erased, starting from the square containing the error right up until the last square (partial erasure). In the third case, only the command or commands that were assumed to be incorrect were erased (targeted erasure) and corrected by leaving some of the squares empty or even by writing two steps in the same square.

3.2.5 Introducing a New Task and a Second Algorithmic Strip to Learn How to Use the "Pause" Button Reflecting on the value of using the Pause button, the teacher opted to turn the matter into a problem for the students by introducing a second Bee-Bot. The aim was to encourage the students to design two programs and to use a Pause instruction to avoid a collision between the two robots. The students were given a second algorithmic strip to help them synchronize the movements of the two Bee-Bots (Figure 13).



Figure 13. The Two Algorithmic Strips

#### a) Description of the activity

In the first phase, the students were encouraged to use the *Pause* button and found that pressing it when the robot was in motion had no effect. Within each group, discussions among the students, supported by directly manipulating the robots, helped them to reconsider their use of the *Pause* button. During these exchanges, the *Pause* function acquired the status of an instruction—a status further reinforced by the introduction of a symbol used to represent it on the algorithmic strip alongside the movement instructions. The students were also encouraged to materialize the square on which the instruction had to be executed (Figure 14).

Numerous conversion processes were found to be at work: the symbol on the control interface, likened, through the button's color, to a control command for the execution of the program, was converted into different kinds of graphic representations capable of being inserted into the instruction sequence, thereby becoming just one instruction among others.



Figure 14. Positioning of the "Pause" Instruction

In the second phase, the students were required to program the robots to perform two movement sequences that crossed paths by using the Pause instruction to avoid collisions.

In this phase, the students made use of all the instruments (small board, digital strips and associated gestures) previously involved in forming a system of instruments performing complementary roles in order to design the robots' itinerary, define each robot's instruction sequence, implement the sequence, and debug the program if necessary.

The difficulty lay in synchronizing the two robots. In the event that the two robots collided, the students returned to using the direct manipulation strategy employed in the first sessions. By manually simulating each robot's movements, the students were able to reflect further on the potential uses of the *Pause* instruction.

#### b) Error management

The learners realized that an error had occurred if the robots collided. Various strategies were used to avoid collisions, including modifying both programs, adjusting the execution time by incorporating Pauses, or delaying one of the robots by holding it back. By using specific strategies to optimize their algorithm, the students learned that a movement instruction involves both a spatial and a temporal dimension.

The simultaneous execution of two programs also served to introduce the students to the concept of event—in particular, the event of triggering movement.

c) Instrumental and conceptual geneses

In this learning situation, following an introduction to the robots involving direct manipulation and then direct control, different artefacts were introduced in succession to help the students graphically represent the itinerary to be executed and the instruction sequence to be implemented and thereby support group discussion, the implementation of the program, and debugging. Through their properties (size, shape, etc.), each artefact was designed to generate specific representations of the itinerary and of the instruction sequence.

Analysis shows that the students converted the artefacts into instruments within their groups by incorporating them into action sequences gradually forming schemes (i.e. unchanging structures of action), with the teacher providing guidance in specifying their roles and, in the event of difficulty,

supporting the students by demonstrating how to use them. The instruments were organized into an instrumental system (Rabardel & Bourmaud, 2003), with each instrument performing a role complementing that of the others. The resulting instruments enabled numerous conversions, including:

- Matching of the space within which the robot moves (large board) with a standardized figurative space (small board).
- Conversion of an allocentric representation of the itinerary into an egocentric representation by manipulating the direction of the small board and by projecting oneself "into the eyes" of the Bee-Bot.
- Conversion of a continuous representation of the itinerary into a representation using symbols similar to those found on the robot's control pad.
- Conversion of a two-dimensional representation of the itinerary into a linear representation, imposed by the characteristics of the algorithmic strip materializing the sequential nature of the instructions.
- Verbalization of the itinerary in a more or less condensed form and subsequently implemented on the Bee-Bot.
- Matching of the Bee-Bots' movements with the various intermediate representations constructed to test the validity of the program.

It should be noted, however, that this process was not supported by all the artefacts. For example, while the symbols shown on the deck of cards had a similar shape to those found on the robot's control pad, the partial consistency of the action that they allowed for in each of the registers caused confusion.

The numerous abstractions required to construct the concept of sequence are thus embedded and materialized by the artefacts and the associated action sequences, thereby forming a concrete abstraction of the concept of sequence.

#### 4. Discussion

The focus of this paper was the activity of students aged 6 to 8 years old during robotics learning sessions designed and implemented by two elementary school teachers (with the objective being to program a Bee-Bot's movements along a route mapped by the students). The aim was to examine the role of semiotic and material artefacts during the learning process. The findings highlight their importance in the construction of the symbolic representations making up the formal language to be acquired by students when learning how to program (Bers, 2019) and of the concept of instruction sequence.

In both case studies, the learners initially encountered difficulties often faced by programming beginners, including difficulties in interpreting the robot's behavior (Duboulay, 1986; Levy & Mioduser, 2008) and in taking into account the deferred nature of the execution of the program (Rogalski, 2015). Like other learning situations (for example, Kalas et al., 2018), to overcome these difficulties, the

students initially resorted to manipulating the robot directly. They then went on to develop more sophisticated strategies involving direct control, associated with a step-by-step strategy (in which the instruction given is immediately executed) and, at a later stage, computational control. The latter strategy involves programming the instruction sequence to be executed by the robot.

In the first case study, the shift from direct control to computational control achieved by all the students involved producing a plan of the itinerary. The task forced the students to defer the actual programming work in order to focus on planning. Producted by incorporating collectively designed conventions, their graphic representations used as a structuring instrument for programming the robot, supporting conversions from one semiotic register to another (Duval, 2006, 2017) and serving to anticipate the program to be implemented. By way of complementing the graphic representation, the students showed initiative by using their body as an instrument for supporting the conversion from an allocentric representation to an egocentric representation to handle changes in direction. In doing so, the students were forced to consider the deferred nature of the execution of a program, to appropriate the symbolic representations used to code, and to adopt an investigative and exploratory approach. By the end of the session, all the students were able to program a simple itinerary consisting of several steps and one or two rotations.

In the second case study, the transition from direct control to computational control was made possible by the teacher introducing several artefacts (large board, small board, algorithmic strip), choices prompted by the specific difficulties encountered by the students. The properties of the artefacts (shape, size, pre-existing inscriptions) restricted the representations which the students could construct of the planned itinerary. By being gradually associated with collectively constructed, socially shared schemes organized into systems (Rabardel & Bourmaud, 2003), the artefacts-turned-instruments helped to direct the students' attention to the properties relevant to programming (standardized length of the robot's step, sequentiality of the program): representing traces of programming based on a process of collective co-construction, these instruments supported numerous conversions between different semiotic registers. These caused the students to move from a continuous, allocentric and two-dimensional representation to a discontinuous, sequential and standardized representation focused on the robot's axis of rotation. Once the instrumental system was in place, all the students were able to program complex itineraries (consisting of a significant number of steps and multiple rotations), to debug them, and to plan the itinerary of several robots within the same space.

However, the study found that the various artefacts were not equally effective in supporting the construction of symbolic representations necessary for programming. In the second case study, while the teacher opted to use a deck of coding cards with symbols similar to those found on the robot's control pad, as in other scenarios designed by researchers (Komis & Misirli, 2015, 2017), the effect was different. The schemes associated with the cards (Note 4)—i.e., placing cards on the board—were not effective in enabling the learners to become sufficiently detached from "doing" to anticipate the

robot's movements. In the scenario proposed by Komis and Misirli (2015; 2017), another scheme was associated with the cards: students were required to place the cards sequentially on a small ruler, enabling them to move from a two-dimensional representation of the itinerary to a linear sequential representation before programming.

In both case studies, the students experienced difficulties in representing changes in direction. These difficulties may be interpreted as a lack of congruence between the symbols used by the learners, the symbols shown on the Bee-Bot's tangible interface to represent rotations, and the action performed. As noted by Duval (2006; 2017), the conversion from one representation to another occurs spontaneously when they are congruent, i.e. when there is a semantic correspondence between the signifying units, the same mode of understanding these units in the two representations, and the possibility of converting an initial signifying unit into a single signifying unit in the final representation. When programming a change in direction, these are only partially consistent. On the control interface, the "right arrow/left arrow" buttons suggest movement directed to the right or the left, yet they activate a rotation movement. In the representations initially used in class (on the plans, the cards, and the algorithmic strip), the representation of a change in direction corresponded not to one but to two instructions (go, turn/turn, go), thereby causing confusion.

Several limitations are worth noting. First, video data is dense, including interactions between students and teacher. The role played by these social mediations have been analyzed in another paper (Nogry, 2019).

In these case studies, several groups were filmed during each session. Similar schemes and conversions were observed between them. However, such a study would benefit from being supplemented by a more systematic analysis of one group activity, session after session. This would enable a more systematic analysis of micro-geneses mediated by artefact during the learning process for each group member, in order to document the collective and individual learning processes. An intrinsic analysis of the activity could complement these video data, highlighting the concerns of each learner, what s/he pays attention to, or the meaning s/he constructs during the activity (Dieumegard et al., 2019).

#### 5. Conclusion

Following Mioduser et al. (2009), this study highlights the embeddedness of the process of abstraction in teaching and learning. More specifically, by articulating semiotic perspective and instrumental approach, it identifies the different functions ensured by semiotic and material artefacts during the programming learning process.

In our view, these results support the hypothesis that the ability to understand what an action sequence is and to implement it requires students to be able to perform multiple conversions from one semiotic register to another, to the point of being able to use symbolic representations corresponding to programming language. These conversions are not simply the result of individual mental activity. Rather, in becoming instruments, they support both individual and collective activity, shaping and constraining the construction of these semiotic representations. The semiotic representations developed and inscribed on these artefacts are also governed by conventions developed collectively in class. They serve to keep a trace of the group's activity, support each individual's reasoning, and promote coordination within the group. The move from one representation to another is managed collectively during this process.

This study highlights the need for a more systematic analysis of the roles performed by the artefacts used and inscriptions produced in studies devoted to the learning process. To complement this study, conducted among children aged 6-8 years discovering IT for the first time, further research is needed to improve our understanding of the role of semiotic and material artefacts in the robotics learning process from a developmental perspective by combining microgenetic and ontogenetic approaches. At a methodological level, the analysis of student activity based on video recordings of sessions could be extended by taking into account students' perspective on their own activity with a view to understanding their experience during the learning process and the meanings constructed by students in these situations (Decortis, 2015; Dieumegard et al., 2019; Jung & Won, 2018).

This study also has implications for teaching, highlighting the need to design a range of artefacts capable of supporting the production of semiotic and material artefacts consistent with computational language and the concepts that need to be developed in designing pedagogical scenarios. With the design process continuing through usage and practice (Folcher & Rabardel, 2004), the instrumental geneses that students associate with these artefacts in specific situations deserve particular attention.

#### Acknowledgments

This study was funded by the French National Research Agency (ANR DALIE & IE-CARE projects). We are especially grateful to the teachers who participated in the study, as well as the participating students and their families. We also wish to thank Georges-Louis Baron for reviewing our manuscript so carefully. This work is dedicated to the memory of François Villemonteix, who launched the IE-CARE project and was an ardent supporter of IT teaching.

#### References

- Abrahamson, D., & Lindgren, R. (2014). Embodiment and embodied design. In *The Cambridge Handbook of the Learning Sciences, Second Edition. UC Berkeley*. https://doi.org/10.1017/CBO9781139519526.022
- Ackermann, E. K. (2013). Microgenetic learning analysis: A methodology for studying knowledge in transition. *Human Development*, 56(1), 38. https://doi.org/10.1159/000345540

Published by SCHOLINK INC.

- Alimisis, D. (2012). Robotics in Education & Education in Robotics: Shifting Focus from Technology to Pedagogy. In *Proceedings of the 3rd International Conference on Robotics in Education* (pp. 7-14). Prague, Czech Republic: Charles University in Prague.
- Baron, G. L., & Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française? Mise en perspective historique. *Revue française de pédagogie*, 195(2), 51-62. https://doi.org/10.4000/rfp.5032
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978-988. https://doi.org/10.1016/j.compedu.2011.10.006
- Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology*, 24(5), 628-647. https://doi.org/10.1007/s10956-015-9552-x
- Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, 6(4), 499-528. https://doi.org/10.1007/s40692-019-00147-3
- Bers, M. U., Flannery, L., Kazakoff, E. R., & 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
- Black, J. B., Segal, A., Vitale, J., & Fadjo, C. L. (2012). Embodied cognition and learning environment design. *Theoretical foundations of learning environments*, 2, 198-223.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association*, Vancouver, Canada (Vol. 1, p. 25).
- Chen, G, Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175. https://doi.org/10.1016/j.compedu.2017.03.001
- Crahay, M. (1987). Logo, un environnement propice à la pens é proc édurale. *Revue française de p édagogie*, 80(1), 37-56. https://doi.org/10.3406/rfp.1987.1473
- Decortis F. (2015). Ergonomie orient ée enfants. Paris: PUF. https://doi.org/10.3917/puf.decor.2015.01
- diSessa, A. A. (2014). The Construction of Causal Schemes: Learning Mechanisms at the Knowledge Level. *Cognitive Science*, *38*(5), 795-850. https://doi.org/10.1111/cogs.12131
- Du Boulay, B. (1986). Some difficulties of learning to program. Journal of Educational Computing Research, 2(1), 57-73. https://doi.org/10.2190/3LFX-9RRF-67T8-UVK9
- Duval, R. (2006). A cognitive analysis of problems of comprehension in a learning of mathematics. *Educational studies in mathematics*, *61*(1-2), 103-131. https://doi.org/10.1007/s10649-006-0400-z
- Duval, R. (2017). Registers of semiotic representations and analysis of the cognitive functioning of mathematical thinking. In *Understanding the mathematical way of thinking–The registers of*

*semiotic representations* (pp. 45-71). Springer, Cham. https://doi.org/10.1007/978-3-319-56910-9\_3

- Elkin, M., Sullivan, A., & Bers, M. U. (2018). Books, Butterflies, and 'Bots: Integrating Engineering and Robotics into Early Childhood Curricula. In *Early engineering learning* (pp. 225-248). Springer, Singapore. https://doi.org/10.1007/978-981-10-8621-2\_11
- Fluckiger, C. (2019). Une approche didactique de l'informatique scolaire. Presses universitaires de Rennes.
- Folcher, V., & Rabardel, P. (2004). Artifacts as 'design-for-use' propositions for 'design-in-use' activity. *International Journal of Psychology*, 39(5-6), 386-386.
- Forcisi, L., & Decortis, F. (2018). Conception participative pour l'activit é narrative et créative avec des enfants de 9-11 ans dans une perspective de l'ergonomie de l'activit é In 30eme conférence francophone sur l'interaction homme-machine (p. 12).
- Gaudiello, I., & Zibetti, E. (2013). La robotique éducationnelle: état des lieux et perspectives. *Psychologie française*, 58(1), 17-40. https://doi.org/10.1016/j.psfr.2012.09.006
- Gourlet, P., & Decortis, F. (2018). Prototyping a designerly learning through authentic making activities in elementary classrooms. *International journal of child-computer interaction*, 16, 31-38. https://doi.org/10.1016/j.ijcci.2017.11.002
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. https://doi.org/10.3102/0013189X12463051
- Hall, R., & Nemirovsky, R. (2012). Introduction to the special issue: Modalities of body engagement in mathematical activity and learning. *Journal of the Learning Sciences*, 21(2), 207-215. https://doi.org/10.1080/10508406.2011.611447
- Hershkowitz, R., Schwarz, B. B., & Dreyfus, T. (2001). Abstraction in context: Epistemic actions. Journal for Research in Mathematics Education, 32(2), 195-222. https://doi.org/10.2307/749673
- Hoc, J.-M., Green, T. R. G, Samurcay, R., & Gilmore, D. J. (Eds.) (1990). Psychology of Programming. Academic Press.
- Jung, S. E., & Won, E. S. (2018). Systematic review of research trends in robotics education for young children. Sustainability, 10(4), 905. https://doi.org/10.3390/su10040905
- Kaptelinin, V., & Nardi, B. A. (2006). Acting with technology: Activity theory and interaction design. MIT press. https://doi.org/10.5210/fm.v12i4.1772
- Komis, V., & Misirli, A. (2015). Etude des processus de construction d'algorithmes et de programmes par les petits enfants à l'aide de jouets programmables. In B. Drot-Delange, G.-L. Baron, & E. Bruillard, *Informatique en éducation: Perspectives curriculaires et didactiques*. Clermont-Ferrand: Presses Universitaires Blaise-Pascal.
- Komis, V., Romero, M., & Misirli, A. (2017). A scenario-based approach for designing educational robotics activities for co-creative problem solving. In D. Alimisis, M. Moro, & E. Menegatti (Eds.)

*Educational Robotics in the Makers Era* (pp. 158-169). Springer. https://doi.org/10.1007/978-3-319-55553-9\_12

- Kopcha, T. J., Ocak, C., & Qian, Y. (2021). Analyzing children's computational thinking through embodied interaction with technology: A multimodal perspective. *Educational Technology Research and Development*, 69(4), 1987-2012. https://doi.org/10.1007/s11423-020-09832-y
- Lave, J. (1988). Cognition in practice: Mind, mathematics and culture in everyday life. Cambridge University Press. https://doi.org/10.1017/CBO9780511609268
- Levy, S. T., & Mioduser, D. (2008). Does it "want" or "was it programmed to..."? Kindergarten children's explanations of an autonomous robot's adaptive functioning. *International Journal of Technology and Design Education*, 18(4), 337-359. https://doi.org/10.1007/s10798-007-9032-6
- Li, H., & Zhang, Y. (2021), Practice Exploration of Innovation Education Mode based on Science and Technology Innovation Team under New Engineering Background. *IEEE 2nd International Conference on Education, Knowledge and Information Management* (pp. 590-593). https://doi.org/10.1109/ICEKIM52309.2021.00134
- Lindgren, R., Tscholl, M., Wang, S., & Johnson, E. (2016). Enhancing learning and engagement through embodied interaction within a mixed reality simulation. *Computers & Education*, 95, 174-187. https://doi.org/10.1016/j.compedu.2016.01.001
- Marougkas, C., Troussas, A., Krouska, C., Sgouropoulou, A. (n.d.). Framework for Personalized Fully Immersive Virtual Reality Learning Environments with Gamified Design in Education. *Frontiers* in Artificial Intelligence and Applications, 338, 95-104.
- Michayluk, J. O. (1986). LOGO: More than a decade later. British *Journal of Educational Technology*, *17*(1), 35-41. https://doi.org/10.1111/j.1467-8535.1986.tb00495.x
- Mioduser, D., & Levy, S. T. (2010). Making sense by building sense: Kindergarten children's construction and understanding of adaptive robot behaviors. *International Journal of Computers* for Mathematical Learning, 15(2), 99-127. https://doi.org/10.1007/s10758-010-9163-9
- Mioduser, D., Levy, S. T., & Talis, V. (2009). Episodes to scripts to rules: Concrete-abstractions in kindergarten children's explanations of a robot's behavior. *International Journal of Technology* and Design Education, 19(1), 15-36. https://doi.org/10.1007/s10798-007-9040-6
- Misirli, A., & Komis, V. (2020). Emerged Debugging Abilities in Early Childhood Education. *Constructivism*, 92.
- Nogry, S. (2019). Robotique pédagogique à l'école primaire: Quelle activité des élèves de Classe Pr éparatoire (6-7 ans) et quels apprentissages dans une séquence conçue par l'enseignant? *Review of Science, Mathematics and ICT Education*, *13*(1), 93-110.
- Noss, R., Hoyles, C., & Pozzi, S. (2002). Abstraction in expertise: A study of nurses' conceptions of concentration. *Journal for Research in Mathematics Education*, 33(3), 204-229. https://doi.org/10.2307/749725

Odgaard, A. B. (2022). What is the Problem? A Situated Account of Computational Thinking as Problem-Solving in Two Danish Preschools. *KI-Künstliche Intelligenz*, 1-11. https://doi.org/10.1007/s13218-021-00752-4

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books.

- Parnafes, O. (2013). Microgenetic learning analysis: A methodology for studying knowledge in transition. *Human Development*, 56(1), 5-37. https://doi.org/10.1159/000342945
- Rabardel, P. (1995). Les hommes et les technologies. Paris: Armand Colin.
- Rabardel, P. (2001). Instrument Mediated Activity in Situations. In A. Blandford, J. Vanderdonckt, & P. Gray (Eds.), *People and Computers XV—Interaction without Frontiers*. Springer, London. https://doi.org/10.1007/978-1-4471-0353-0\_2
- Rabardel, P., & Bourmaud, G. (2003). From computer to instrument system: A developmental perspective. *Interacting with computers*, *15*(5), 665-691. https://doi.org/10.1016/S0953-5438(03)00058-4
- Radford, L. (2009). Why do gestures matter? Sensuous cognition and the palpability of mathematical meanings. *Educational Studies in Mathematics*, 70(2), 111-126. https://doi.org/10.1007/s10649-008-9127-3
- Radford, L. (2014). Towards an embodied, cultural, and material conception of mathematics cognition. *ZDM*, *46*(3), 349-361. https://doi.org/10.1007/s11858-014-0591-1
- Rogalski, J. (2015). Psychologie de la programmation, didactique de l'informatique déjà une histoire... In B. Drot-Delange, G.-L. Baron, & E. Bruillard (Eds.), *Informatique en éducation: Perspectives curriculaires et didactiques*. Clermont-Ferrand: Presses Universitaires Blaise-Pascal.
- Saenz-Ludlow, A., & Presmeg, N. (2006). Guest editorial semiotic perspectives on learning mathematics and communicating mathematically. *Educational Studies in Mathematics*, 61(1-2), 1-10. https://doi.org/10.1007/s10649-005-9001-5
- Spach, M. (2017). Activités robotiques à l'école primaire. Quelle place du scénario pédagogique? Les limites du co-apprentissage. Unpublished phd thesis, universit éDescartes, Paris.
- Strawhacker, A., & Bers, M. U. (2019). What They Learn When They Learn Coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*, 67(3), 541-575. https://doi.org/10.1007/s11423-018-9622-x
- Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, 26(1), 3-20. https://doi.org/10.1007/s10798-015-9304-5
- Sung, W., Ahn, J.H., Kai, S. M. & Black, J. (2017). Effective Planning Strategy in Robotics Education: An Embodied Approach. In P. Resta, & S. Smith (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference* (pp. 1065-1071).

Published by SCHOLINK INC.

- Toh, L. P. E., Causo, A., Tzuo, P. W., Chen, I. M., & Yeo, S. H. (2016). A review on the use of robots in education and young children. *Journal of Educational Technology & Society*, *19*(2), 148-163.
- Yasar, O. (2018). Computational thinking, redefined. In Society for Information Technology & Teacher Education International Conference (pp. 72-80). Association for the Advancement of Computing in Education.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. https://doi.org/10.1145/1118178.1118215
- Wu, T. T., & Chen, J. M. (2021). Combining Webduino Programming With Situated Learning to Promote Computational Thinking, Motivation, and Satisfaction Among High School Students. *Journal of Educational Computing Research*. https://doi.org/10.1177/07356331211039961

#### Notes

Note 1. This case study was also described in another paper examining the different forms of supporting explanations provided by the teacher in a given learning situation (Nogry, 2019).

Note 2. This paper presents an analysis of the social mediations provided by the teacher during the sessions.

Note 3. The deck of cards consisted of six different cards of similar dimensions to those of a single square on the large board. Each card showed an icon resembling the icon on the Bee-Bot's buttons and captioned with a word describing the corresponding action. A small dot located in the lower righthand corner of each card was used to facilitate the process.

Note 4. This scheme consisted in placing them on the board by way of representing the itinerary in the form of instructions and in inscribing a rotation movement on the cards to simulate the Bee-Bot's movements