



**HAL**  
open science

# Introducing Computational Thinking to K-5 in a French Context

Vanea Chiprianov, Laurent Gallon

► **To cite this version:**

Vanea Chiprianov, Laurent Gallon. Introducing Computational Thinking to K-5 in a French Context. 21st Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2016), Jul 2016, Arequipa, Peru. 10.1145/2899415.2899439 . hal-01908224

**HAL Id: hal-01908224**

**<https://hal-univ-pau.archives-ouvertes.fr/hal-01908224>**

Submitted on 2 Apr 2019

**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.

# Introducing Computational Thinking to K-5 in a French Context

Vanea Chiprianov  
University of Pau & Pays Adour  
Mont de Marsan, France  
vanea.chiprianov@univ-pau.fr

Laurent Gallon  
University of Pau & Pays Adour  
Mont de Marsan, France  
laurent.gallon@univ-pau.fr

## ABSTRACT

Computational Thinking (CT) is beginning to be accepted as one of the fundamental 21<sup>st</sup> century skills for everyone. Curricula and environments are being developed for different group ages, from kindergarten to university, in several countries. As part of this global tendency, France has recently taken political decisions to integrate CT Education (CTE) in the mandatory national curriculum. However, many challenges remain until a full implementation is achieved. In this paper we report on a partnership between a university, local elementary schools and the county Ministry of Education (MoE), and on an exploratory project of introducing CT to K-5 students. This project has provided us with valuable feedback on the specifics of integrating CT in a national curriculum and the creation of a partnership and a community. These lessons will be used in the following stage of scaling up to more elementary schools in the entire county, but also addressing other school levels such as kindergarten, middle school and high school.

## Keywords

Computer Science Education, Elementary School, Curriculum, Community of practice, Teacher training

## 1. INTRODUCTION

Computational Thinking (CT), while still being debated on its exact definition, deals mainly with a generalization of Computer Science (CS) concepts, skills and principles for a larger demographic. It involves [23], [16], [11]: solving problems; designing systems; conditional logic; iterative, recursive and parallel thinking; using abstraction and pattern generalizations; problem decomposition and remixing; critical thinking; creativity; systematic processing of information and algorithmic notions of flow of control; symbol systems and representations; thinking in terms of prevention, protection, and recovery from worst-case scenarios; debugging and systematic error detection; efficiency and performance constraints; using heuristic reasoning to discover a solution;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*ITiCSE '16, July 09 - 13, 2016, Arequipa, Peru*

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4231-5/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2899415.2899439>

teamwork, communication, leadership; seeing programming as a collaborative, distributed effort - computational participation; assessment.

In practice, CT comprises [4] *concepts* such as: sequences, loops, parallelism, events, conditionals, operators, data; *practices* such as: being incremental and iterative, testing and debugging, reusing and remixing, and abstracting and modularizing; and *perspectives* such as: expressing (oneself, with a creative tool), connecting (by creating *with* others and *for* others), questioning (interrogating the taken for granted).

CT has been argued to be a fundamental skill for everyone, not only computer scientists [23], [16], [11]. Due to the pervasiveness of computers, many of the advances in numerous fields nowadays involve CS. Moreover, many of the current problems require diverse teams with backgrounds in multiple domains, and an understanding of the fundamentals of CS by most team members may help enhance critical elements of proposed solutions.

In many countries there is an important change towards promoting CT as a standard subject, e.g., UK [5], USA and Israel [9], New Zealand [2]. Moreover, the focus is from teaching how to *use* computers - digital literacy, to how to *program* - informatics, empowering children.

While it appears the value of learning CT has been established, some **big questions** still remain [12], among which: *At what age should CT Education (CTE) start? Which content, learning objectives, methods, and environments are suitable to learn CT concepts at each developmental stage? How to integrate CT teaching time with other important learning fields? How to best prepare and sustain teachers in acquiring the necessary CT skills? How to take into account national/local curriculum specificities?*

In this paper we report on a *partnership* between a university, local elementary schools (ES) and the departmental French Ministry of Education (MoE). This produced an exploratory project of introducing CT to K-5 students, in the 2014-2015 school year. Such intervention studies in the regular classroom are the next step to evaluate proposed curricula, environments and computational practices, as recommended e.g., by [16]. In the remainder of the paper *we describe how we answered in this exploratory project to the big research questions on introducing CT in French ES.*

## 2. PRESENT STATE OF CTE IN THE FRENCH SCHOOL SYSTEM

A good synthetic and up-to-date presentation of the French Education System can be found in [1]. We focus here on presenting only the aspects pertinent to our work. The school

system in French starts around age 6 with the ES, which lasts for 5 years until about age 11. Children are organized in groups of about 20-30. One teacher takes care of the group, teaching most subjects.

Most ES teachers are trained mainly by Superior Schools of Teaching and Education - SSTE (*Ecoles supérieures du professorat et de l'éducation*) which deliver them a masters. They are then recruited through a national yearly contest and are then tenured civil servants. They are organized in educational districts (*académies*), under the responsibility of a regional pedagogical inspector.

The curriculum is established by groups of experts - the Superior Curricula Council (*Conseil supérieur des programmes*), appointed by the MoE.

In France [1], like in many other countries [10], [2], [5] until recently, the curriculum included CS only from the *user* point of view. The general public, including teachers and students, had little to no awareness of the difference between *using* (digital literacy) and *programming* (informatics) a computer. For example, in France, even at high school level, the tests for CS required just elementary usage and office skills<sup>1</sup>. However, this is changing.

On 7<sup>th</sup> May 2015, the French president announced<sup>2</sup> one billion euros dedicated to initiating all children, from ES to high school, to both *programming* and digital literacy, as part of a larger plan for the digitalization of schools, from the start of the 2016 school year. He also promised an exceptional program to prepare the teachers and school personnel on the next three years, 2016, 2017, and 2018. While the SSTE will take care of the initial training of teachers, additionally there will also be an exceptional effort for the ongoing education / lifelong learning.

In September 2015, the Superior Curricula Council proposed new curricula for kindergarten, elementary and middle school [6]. It explicitly includes programming for ES: "Ils décrivent un système technique par ses composants et leurs relations. Les élèves découvrent l'algorithme en utilisant des logiciels d'applications visuelles et ludiques. Ils exploitent les moyens informatiques en pratiquant le travail collaboratif." Roughly translated as: "They (pupils) describe a technical system through their components and relations. The pupils discover the concept of algorithm by using graphical and playful software. They explore computing tools collaboratively."

To help implement these objectives in 2016, the French Education Minister announced<sup>3</sup> that in ES, an introduction to coding will be proposed, on a optional basis, in the extracurricular time. Moreover, to help teachers, all SSTE have included CS in their curriculum, starting 2014. However, it should be noted that until now, this mainly consisted in digital literacy, and not programming.

To conclude, the French education system is undergoing a major change, at all levels, to introduce and develop CT. This raises many challenges, from developing suitable curriculum that weave informatics in a multidisciplinary manner, to training the teachers. Weaving CS with existing curriculum subjects comes to meet the very concept of CT

as a multidisciplinary and fundamental skill that helps developing cognitive aptitudes of students.

### 3. CHILD COGNITIVE AND AFFECTIVE DEVELOPMENT

As discussed earlier, one of the big questions of CT research is related to *the age at which CTE should start*. Many aspects need to be taken into account. We will focus here on affective and cognitive ones.

Research from educational and cognitive psychology [15] suggests that both affect and cognitive attributes develop easier and are easier to change in the early years, while they are much more stable and therefore harder to influence later in the student's life. This suggests that student exposure to CT and STEM in general, earlier in their life, may create and maintain their affect towards these subjects, thus students being more likely to integrate them in their later education. It is therefore *critical to include CT integrated learning at the earlier elementary levels*. Another reason is related to equity - presenting the fundamentals of CT at a young age gives students of all backgrounds the possibility to absorb and practice these principles as a life-long way of thinking.

The influence of CS on cognitive aspects such as the development of thinking skills (e.g., causal reasoning, metacognition) has been studied since the 1980s [11], [16]. Such studies underline the manner in which teaching happens, for example by remarking the importance of group collaborative activities which reinforce interaction, teachers offering suggestions during difficulties, or designing learning activities that are meaningful and challenging - thus engaging - while also being achievable so as to avoid discouragement [8].

As it is still an important reference, we remind here Piaget's cognitive development stages [20], which for ages 7-11, comprise the concrete operational stage. This stage is characterized by the fact that abstract, hypothetical thinking is not yet developed, and children can only solve problems that apply to concrete events or objects. They are also able to use inductive reasoning, but they struggle with deductive reasoning. Children also begin to think in more scientific and trial-and-error fashion, using hypothetical-deductive reasoning, making plans which they test in a systematic manner.

Of course, any curriculum, activities, tools as well as social organization of teaching need to take into consideration the cognitive development of children, so as to propose appropriate activities and content.

### 4. THE PROJECT: THEORETICAL CONSIDERATIONS

To answer the objectives of introducing CT in French ESs, the University of Pau partnered in 2014 initially with departmental MoE services and two local schools to launch an exploratory project in the department of Landes. The persons directly involved comprised initially: 2 researchers, 4 teachers, 3 classes, and a MoE counselor. The project finally involved many more persons, among which most notably the directors and the technical teachers of the local schools, as discussed further.

In choosing the education objectives, content, activities, tools and in planning their order, we looked at existing curricula in the world and analyzed them in light of specifics of the French education system.

<sup>1</sup><http://www.legifrance.gouv.fr/affichTexte.do;jsessionid=?cidTexte=JORFTEXT000027811513&dateTexte=&oldAction=dernierJO&categorieLien=id>

<sup>2</sup><http://www.elysee.fr/chronologie/#e9309,2015-05-07,d-placement-1-cole-change-avec-le-num-rique->

<sup>3</sup><http://www.gouvernement.fr/action/1-ecole-numerique>

## 4.1 CT in World Curricula

Several curricula are being proposed worldwide to teach CS/CT/Informatics, including the Computer Science: Principles site<sup>4</sup>, the Computing at Schools Initiative<sup>5</sup>, national curricula like those of New Zealand<sup>6</sup>, UK<sup>7</sup> and finally the Computer Science Teachers Association (CSTA) standards [21]. Not all of these address teaching CS/CT from the ES level. One that does is for example the CSTA, which recommends at Level 1 (grades 3-6), in the CT and Computing Practice and Programming strands, among others, that pupils:

1. Understand and use the basic steps in algorithmic problem-solving.
2. Develop a simple understanding of an algorithm (e.g., sequence of events) using computer-free exercises.
3. Demonstrate how a string of bits can be used to represent alphanumeric information.
4. Describe how simulation can be used to solve problems.
5. Make a list of sub-problems to consider while addressing a larger problem.
6. Understand the connections between CS and other fields.
7. Construct a program as a set of step-by-step instructions to be acted out (e.g., make a sandwich activity).
8. Implement problem solutions using a block-based visual programming language.

A lot of emphasis is put on *the concept of algorithm and the flow of control*, using both computer-free activities and block-based visual programming languages.

## 4.2 Specificity of CT in French Curriculum

In France, the new curriculum [6] explicitly includes programming. It is noteworthy that it is well aligned with the CSTA standard, though an exploration in detail of this alignment is out of the scope of this paper.

However, the curriculum for the year 2014-2015, did not explicitly include CS. This could have been a problem for our project, as there seems to be no place for it. However, that curriculum does include a STEM section, which recommends acquiring a scientific awareness and approach - to be noted the concordance with Piaget's concrete operational development stage at this age. An important aspect in our project was then *the presentation of CT in the light of a scientific, systematic investigation, discovery-based approach*.

Therefore, *our exploratory project has been introduced as a project of Support on Science and Technology in ES - in French Accompagnement en science et technologie à l'école primaire (ASTEP)*<sup>8</sup>. ASTEP is an initiative that brings together scientists and school teachers, so as to facilitate getting started with new subjects.

## 4.3 Theoretical Framework: Constructionism

Based on Piaget's *constructivism* [20]: the child actively builds knowledge through experience, and the related "learn-by-doing" approach to education, Papert proposed [18] the *constructionism*: children learn deeply when they build their own meaningful projects in group and reflect on the process.

<sup>4</sup><http://www.csprinciples.org/>, <http://apcsprinciples.org/>

<sup>5</sup><https://www.computingatschool.org.uk/>

<sup>6</sup><http://nzcurriculum.tki.org.nz/Curriculum-documents/The-New-Zealand-Curriculum>

<sup>7</sup><https://www.gov.uk/government/collections/national-curriculum>

<sup>8</sup><http://www.fondation-lamap.org/fr/astep>

Based on such theoretical frameworks, many learning environments have been proposed. For example, [16] proposed that a constructionism-based problem-solving learning environment, with information processing, scaffolding and reflection activities, could be designed to foster computational practices and computational perspectives.

## 4.4 Educational Games, Computer Programming Environments and Robots

In the previous subsections we gave elements towards answering the big question of which content, learning objectives and methods may be suitable to teach CT to ES in France. In this subsection we investigate which environments and tools are suitable to support them.

Many different educational tools and games, activities that use or not the computer, and programming environments specially designed for teaching CT to young children have been proposed. For a short history from Logo to Scratch, cf. e.g., [16]; a review can be found in e.g., [11]; also many resources are grouped in [17]. Most of them are based on constructionism principles.

Choosing among so many environments is not obvious. However, one approach that seems particularly well adapted and is one of the few that propose a full K-12 pathway [19], is that proposed by Code.org<sup>9</sup>. The site is structured as a series of puzzle-based games. To play them, children have to give instructions in a visual programming language of a Blockly type, in which they drag and drop command symbols to compose programs. Key concepts that are taught include, among others: CS, computer scientist, applications of CS, algorithm, decomposing, abstracting, debugging and programming concepts such as: sequence, loops, conditionals, functions with and without parameters, variables. It mixes unplugged, computer-free activities, based on real-life knowledge children already have, with online ones, thus facilitating transfer, connection and abstraction of knowledge. One can *notice the good, if not total accordance between the proposals of Code.org and the definition of CT and aspects of various curricula proposed worldwide, such as that of CSTA*.

Code.org has many valuable aspects, such as pedagogical materials and well organized activities, from simple to complex. It proposes a learning-management system, in which teachers can monitor the activities of their entire class, the progress of each pupil, at their own pace, in terms of completed levels and time spent, which makes it easier to provide guidance and feedback. Individual child progress can be followed by parents as well, thus enabling a deeper implication of children, teachers and parents in the learning process.

More recently, Code.org has partnered with CS: Principles and Exploring CS<sup>10</sup>, which have developed cohesive well-designed CS programs [19]. As of summer 2015, Code.org had prepared more than 10.000 teachers; 2 million girls - 43% of their students - are learning introductory CS [19]. A study [14] teaching programming to 32 ES pupils with the Code.org site found that students developed a positive attitude towards programming and that programming could be part of their future plans.

Another category of tools that have been investigated for education are robots. A systematic review [3] presents a synthesis of the available quantitative empirical evidence on

<sup>9</sup><https://code.org/>

<sup>10</sup><http://www.exploringcs.org/>

the effectiveness of robotics as an educational tool in schools. It concludes that educational robotics usually act as an element enhancing learning. It notes that robotics help emphasize skills in problem solving, logic and scientific inquiry.

The robots we selected are Dash<sup>11</sup> and Thymio<sup>12</sup>. Dash can be programmed using a Blockly-like environment, which is very similar to that used by Code.org. Thymio has a programming environment which combines visual and textual programming, which makes it a potential good candidate to ease the transition towards textual programming languages.

## 5. THE PROJECT: PUTTING IT INTO PRACTICE

In parallel with addressing questions related to content, methods and tools of teaching, partnerships had to be established and decisions taken.

### 5.1 Organization, Preparation and Developing a Community of Practice

Several meetings and informal discussions took place during the school year 2013-2014, to prepare the project for the year 2014-2015. It was decided 2 schools be part of the project. One is in Mont de Marsan, the county town, and the other in a nearby village. One of them has a good internet connection, and access to 10 laptops; their class size is around 30; 2 classes from this school would participate in the project. The other school has a weak internet connection, one computer in the classroom, and class size is around 20 pupils; one class would participate.

All teachers were motivated and had inclinations towards digital literacy, but none had previous knowledge of programming. A training session of 3 hours was organized, in which the project was officially kick-started in the presence of the MoE inspector and then presentations and discussions of the proposed curriculum and tools took place. This training session was a good start, but had to be completed by informal discussions and sometimes live decisions in the classroom. Also, teachers shared between them by mail or by discussion their decisions and adaptations. Moreover, a site enabling exchange of such information was put in place.

The material consisted in 10 tablets, 5 Dash robots and 10 Thymio robots. This meant that during lessons using Code.org, when tablets were used, depending on the schools and their own computers, pupils worked either individually or in pairs. For lessons involving robots, pupils were organised in groups of 4 to 6 for Dash, and 2 to 3 for Thymios.

### 5.2 Lesson Plan

In the context of ASTEP, 10 lessons were dedicated to the project. Our curriculum is primarily inspired from the one proposed by Code.org, Course 2. We selected and adapted activities from it. For example, we selected the lessons dealing with CT concepts such as algorithm, programming, variable, loop and conditional. These seem essential concepts, as recommended by worldwide curricula, and in particular the CSTA standards. We alternated unplugged activities with computer ones. Due to French specific curriculum, some activities could not be used. For example, the puzzles involving an “artist” use trigonometry concepts which

are not taught in the French ES. Most teachers thus decided to exclude them. Among the unplugged activities, the dance introducing the concept of loop was replaced with other repetitive activities that involved less rearranging of class furniture.

The curriculum encourages active learning. The unplugged activities are based on pupils’ previous knowledge, such as introducing the concept of algorithm based on routine activities like teeth brushing. This scaffolding process eases the acquisition of CT concepts and skills. Based on the concepts thus introduced, the computer-based puzzles invite pupils to search solutions, first individually and then collaboratively, and ultimately search the advice of the teacher. In this way, CT concepts and problem-solving skills, are actively acquired, using an inquiry-based pedagogy.

To these lessons we added an introductory lesson in which programming was introduced as an objective to program behaviors like robots finding their way on a carpet on which a labyrinth inspired from Code.org puzzles has been drawn. After this, the Code.org activities followed. At the end, two more sessions used the acquired concepts to program the robots, one for Dash and one for Thymio.

The Blockly-based programming environment for Dash resembles the one used by Code.org. Moreover, it can be installed on tablets, therefore the transition was quite straightforward for pupils. For Thymio, at the moment we used it for this project, the programming environment existed only for computers, which posed additional constraints, as the tablets could not be used. Also, the programming paradigm for Thymio is an event-based one, with no graphical entities for concepts such as iteration. Therefore, while using Dash seems well suited at this stage, Thymio seems better suited at a later stage.

### 5.3 Strategies of Teaching

The classroom delivery involved several methods. Co-teaching between the class teacher and a researcher was frequently employed. While the teacher conducted the lesson, they would solicitate the researcher for specific points, usually more technical or unclear content. Similar to other case studies [13], a typical lesson would start with whole-class directions and demonstrations, thus enabling learning by example, after which independent and collaborative work followed. Depending on the available material, pupils worked either individually, or in pairs, or in groups of 4 to 6, thus enabling collaborative learning and peer instruction.

Although reflection time for sharing solutions with the entire class and reviewing comments on examples, were included in the curriculum for each lesson, in practice this was not always possible. This was sometimes due to technical problems, or to incorrect estimates of the time necessary to organize the pupils into groups, or because of the teacher to pupil ratio. In some cases, for classes of about 30 pupils, in which each worked individually, the teacher and researcher were over-solicited. In some cases, a third university staff or student had to participate for additional support.

## 6. LESSONS LEARNED

One of the major benefits of this project have been the multiple lessons related to the various actors involved.

### 6.1 The Teachers

Teachers play an essential role for multiple reasons. They

<sup>11</sup><https://www.makewonder.com/dash>

<sup>12</sup><https://www.thymio.org/en:thymio>

stimulate pupils and give them positive attitudes [3]. They also implement the curriculum change [2] and adapt it to their environment, pupils, material. For example, some teachers chose to eliminate certain lessons from the proposed curriculum (e.g. to teach the concept of iteration, they estimated the second lesson using planes was not necessary). It is thus *essential they have a degree of autonomy in implementing the curriculum*. It is them who ensure the pedagogical presentation of the material. Their main motivation is providing better opportunities for pupils.

The teachers involved in our project were enthusiastic about integrating CT. However, coaching on how to use the material, the software and on how to solve the puzzle-based content was essential. This brings us to a possible major barrier [13], [2]: *professional development of teachers*. In the beginning of our project, teachers had only one 3-hour training session. Additional training happened during the lessons, or in informal discussions beforehand. While this was feasible for a reduced number of motivated teachers, it is of course not feasible for the long term.

To tackle this issue, in the current year starting September 2015, in partnership with the French MoE, 3 more hours have been allocated. This is a step in the right direction, and a considerable effort on the part of the MoE. However, to extend this training for the entire department or region, more time and tools like MOOCs [7] would be needed.

Feedback from teachers allowed us to identify strong and weak points in our curriculum. For example, groups of 4 (especially in robotics classes, as in other studies [3]) was estimated as too big a number. Teachers also evoked the need for more time being allotted for activities that allow them to observe and evaluate how and where each pupil is on their path of acquiring the concepts and to share with the entire class different solutions and discuss them.

Many of these comments have been integrated in the curriculum version for the current year. To ensure smaller groups, in some cases we will experiment with dividing the class in two. Another solution is to have more material (e.g., a bigger number of robots). This will allow for teams of 3 or even pairs. However, we expect having a bigger number of groups (10 to 15!) to put a bigger strain on the teacher.

To allow for more sharing time, all lessons have been allotted 15 minutes at the end (but which can be used anytime the teacher decides) to share solutions and discuss them. To enable discussions, solutions with comments on advantages and disadvantages have been provided.

## 6.2 The Children

A typical robotics lesson sequence consists in children programming the robot, either at the tables of their group or a little aside from the labyrinth carpet, then coming on the carpet, sitting down the robot and themselves on it, and testing the program. In the lessons dedicated to puzzle-based games, on the hand-held tablets, we observed more interaction between the children when they were seated at tables in groups of 4-5, than when seated at tables by 2.

An important note is related to *discovering the material*: robots, but also tablets; this may probably be applied to desktops or laptops or any other material as well. The first sessions, mainly because of time limitations, the teacher presented different visible components, naming them and explaining their function. While this was very time-efficient, and many pupils retained well and afterwards used the ma-

terial correctly, pupils were mostly inactive during this period. In the current iteration of the project we changed this. We dedicate more time, give the material to the pupils, let them look, touch and explore it, and then the teacher asks questions about what they have seen. In this way, children co-construct in an active manner their knowledge.

The introduction to programming and robotics also *activated the use of other skills* by pupils. The current versions of robot programming environments are in English - foreign language. Pupils worked in groups, so their social and language skills were activated.

One skill that deserves special mentioning is *space location*. In the initial curriculum, we started with computer-based puzzles and finished with applying learned CT concepts to robot programming. In the puzzles, pupils need to guide an on-screen character through a labyrinth. They give it commands to advance or rotate left, or right. To determine the right instruction, children would often role play, positioning themselves in place of the character. This seems harder to do when the environment has only 2 dimensions and is much more abstract - the screen, than in the case of robots, with 3 dimensions and which are more concrete, tangible. Therefore, in the current iteration, we are experimenting with introducing spatial relations through robots, in the first lesson, and then transferring it to puzzles. Preliminary observations seem to indicate that children still role-play in the case of puzzles, at least in the first lesson, but that this seems to almost completely disappear in later puzzle lessons.

While we did not have the opportunity to perform extensive evaluation, other studies [14] found that children using Code.org stated learning programming, improving their mathematical knowledge, computer literacy, and cognition, finding solutions through less steps, creating love of mathematics, learning to think logically and to find directions.

## 6.3 The Administrators of Learning

This category includes school directors, but also personnel of the MoE. Their support was instrumental in realizing our project. It is them that gave us permission to enter the schools, and helped us find the right type of administrative form into which we could experiment - the ASTEP project.

They are instrumental in passing at a larger scale. While during the 2014-2015 year the project was deployed around one town, during the year 2015-2016 it is being deployed in schools all over the county. It is the personnel of the MoE who found the teachers interested in being part of our project and who will probably manage the shared material.

Due to their intervention, we can address a larger diversity of pupils, from urban and rural schools, girls and boys, of various ethnic and social background. This will allow for a more representative population on which the project will be experimented and evaluated. It is also instrumental into ensuring equity and equal access to opportunities.

## 6.4 The Researchers and the Universities

One of the main and selfish! reasons for which we, as CS university teachers, are interested in promoting the importance of pupils having a good CS level at pre-university levels, is related to the CS level of students starting courses at universities. The better their CS level, the easier for us, and the further we can go in our courses!

One of the main actions was to begin constructing a community of practice - as advised by example in [5] - , involving

local schools, departmental personnel of the MoE, the local SSTE. This community reinforces connections, enables further research actions, as well as rapid transfer of ideas from research into practice, and mutual support.

As advised also by [2], we are looking into involving selected university students. This may be done in pairs: one CS and one SSTE student. This would enable having both technical and pedagogical skills and knowledge and better assisting the teacher. It would provide further experience for the 2 students and a direct link for pupils to the university.

## 7. CONCLUSION AND PERSPECTIVES

In this paper we presented the first phase of a project introducing CT to ES in a French context. We focused on the kick-start of the project, creating the local community, adapting a curriculum to French specifics, teacher implementation and first feedbacks. In this first phase, formal feedback and validation from pupils was not a main focus.

In the second phase, which started September 2015, our experimental community has expanded to include 3 ESs with 5 classes. Two classes are from urban area, the other 3 from rural. The local SSTE has manifested interest in our project and will very likely join us next year 2016-2017. This will enable pairing of CS students with their students.

Another dimension into which our project is expanding is assuring a continuation from ES through middle and high school. We currently have a partnership with 2 high schools, into which CT is present as an all year long project to prepare Mindstorm robots compete in a local olympics. There is also one middle school interested in joining the robotics olympics. Such partnerships will enable us to test a complete curriculum. We are also considering tackling CT in kindergarten using approaches such as [22].

## 8. ACKNOWLEDGMENTS

We would like to thank especially the primary school teachers: Mélisa Devaux, Sophie Evangelista, Florence Rassinier and Sandra Saint-German and the French MoE counselor, Frederic Carrincazeaux.

## 9. REFERENCES

- [1] G.-L. Baron, B. Drot-Delange, M. Grandbastien, and F. Tort. Computer Science Education in French Secondary Schools: Historical and Didactical Perspectives. *Trans. Comput. Educ.*, 14(2):11:1–11:27, 2014.
- [2] T. Bell, P. Andreae, and A. Robins. A Case Study of the Introduction of Computer Science in NZ Schools. *Trans. Comput. Educ.*, 14(2):10:1–10:31, 2014.
- [3] F. B. V. Benitti. Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3):978 – 988, 2012.
- [4] K. Brennan and M. Resnick. New frameworks for studying and assessing the development of computational thinking. In *American Educational Research Association, Vancouver, Canada*, 2012.
- [5] N. C. C. Brown, S. Sentance, T. Crick, and S. Humphreys. Restart: The Resurgence of Computer Science in UK Schools. *Trans. Comput. Educ.*, 14(2):9:1–9:22, 2014.
- [6] Conseil Supérieur des Programmes. Projet de programmes pour les cycles 2,3,4 (French), 2015.
- [7] K. Falkner, R. Vivian, and N. Falkner. Teaching Computational Thinking in K-6: The CSER Digital Technologies MOOC. In *17th Australasian Computing Education Conference*, 2015.
- [8] G. Fessakis, E. Gouli, and E. Mavroudi. Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63:87 – 97, 2013.
- [9] J. Gal-Ezer and C. Stephenson. A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States. *Trans. Comput. Educ.*, 14(2):8:1–8:18, 2014.
- [10] W. Gander, A. Petit, G. Berry, B. Demo, J. Vahrenhold, A. McGettrick, R. Boyle, A. Mendelson, C. Stephenson, C. Ghezzi, et al. Informatics education: Europe cannot afford to miss the boat. *ACM [online] Available at: <http://europe.acm.org/iereport/ie.html>*, 2013.
- [11] S. Grover and R. Pea. Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42:38–43., 2013.
- [12] P. Hubwieser, M. Armoni, and M. N. Giannakos. How to Implement Rigorous Computer Science Education in K-12 Schools? Some Answers and Many Questions. *Trans. Comput. Educ.*, 15(2):5:1–5:12, 2015.
- [13] M. Israel, J. N. Pearson, T. Tapia, Q. M. Wherfel, and G. Reese. Supporting all learners in school-wide computational thinking. *Comput. Educ.*, 82(C):263–279, Mar. 2015.
- [14] F. Kalelioglu. A new way of teaching programming skills to K-12 students: Code.org . *Computers in Human Behavior*, 52:200 – 210, 2015.
- [15] R. Lamb, T. Akmal, and K. Petrie. Development of a cognition-priming model describing learning in a STEM classroom. *Journal of Research in Science Teaching*, 52(3):410–437, 2015.
- [16] S. Y. Lye and J. H. L. Koh. Review on teaching and learning of computational thinking through programming: What is next for k-12? *Computers in Human Behavior*, 41:51 – 61, 2014.
- [17] L. Mannila, V. Dagiene, B. Demo, N. Grgurina, C. Mirolo, L. Rolandsson, and A. Settle. Computational Thinking in K-9 Education. In *ITiCSE*, pages 1–29, 2014.
- [18] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., 1980.
- [19] H. Partovi. A Comprehensive Effort to Expand Access and Diversity in Computer Science. *ACM Inroads*, 6(3):67–72, 2015.
- [20] J. Piaget. *The construction of reality in the child*. Routledge, 2013.
- [21] D. Seehorn, S. Carey, B. Fuschetto, I. Lee, D. Moix, D. O’Grady-Cunniff, B. B. Owens, C. Stephenson, and A. Verno. CSTA K–12 Computer Science Standards: Revised 2011. 2011.
- [22] D. Wang, T. Wang, and Z. Liu. A Tangible Programming Tool for Children to Cultivate Computational Thinking. *The Scientific World Journal*, 2014.
- [23] J. M. Wing. Computational thinking. *Commun. ACM*, 49(3):33–35, Mar. 2006.