

Interest of Teaching Programming for Primary Preservice Teachers

Jean Simon*

Abstract

In the teachers' training schools of Reunion Island and Mayotte, we have been training primary pre-service teachers in programming for the past three years. Most trainees have not learned programming before during their school and university curriculum. As a result, this subject is new to them and they find themselves in the situation of their own pupils. This is why we asked them, during their learning, to produce a "feelings" notebook in which they explain their successes, their failures and what it brings to them for understanding their own pupils. We held two types of studies of these documents: quantitative and qualitative. The analysis shows that if on the whole, the group of trainees have globally identified the key aspects of a learning situation, it is not the case for each trainee.

Key Words: Teaching of programming, pre-service teachers, socio-constructivism.

Introduction

Since September 2016, primary school teachers have to teach programming to their pupils, while most of them do not know how to program. This is why, on the one hand, continuing training courses have been set up and, on the other hand, this training is now provided as initial training to some of the trainees. These trainees are trained by the Ecole Supérieure du Professorat et de l'Education (ESPE), the teachers' training school. In this article, we are particularly interested in the initial training in programming for trainees from two French departments: Reunion Island and Mayotte Island. This teaching has a double interest: disciplinary and pedagogical. From a disciplinary point of view, the learning of programming allows the learner for better understanding the world in which he lives. It also, and above all, allows him to learn to think with rigor.

From a pedagogical point of view, the learning of programming allows trainees to experience the interest of a socio-constructivist pedagogy. Most trainees have not learned programming through their school and university curriculum. As a result, this subject is new to them and when they learn programming, they find themselves in the situation of their own pupils. Living it will make them more likely to understand socio-constructivism from "the inside" and reproduce it in class with their own pupils. To make them aware of this, we asked them to keep a "feelings" notebook (or "sufferings," we will see below, why) during their apprenticeship.

In the first step, we present rapidly the interest of programming. In the second step, we present the teaching of programming in Mayotte and Reunion Island and the "feelings" notebooks. In the third step, we present the methodology used to analyze these notebooks. In the fourth step, we present the results. Finally, in the fifth step we discuss results and methodology.

* ESPE de La Réunion, LIM, France
Corresponding Email : jean.simon@univ-reunion.fr

Interest of Programming

In 2016, French Ministry of Education decided that all children have to learn programming to understand the world in which they live. It is true that we are now in a computerized society and everything around us contains information technology: cars, washing machines, watches, telephones, etc. Learning programming helps to understand how all these things work (Rushkoff, 2010). However, for us there is another reason to learn programming: *learning to program is learning to think rigorously* (Kaila, Laasko, & Kurvinen, 2018; Rees et al., 2016).

To justify this, it is necessary to explain what programming is. *Programming* consists of two important moments:

- finding an *algorithm*,
- *coding* it in a given language.

That is why we do not like the expression "learning the code" (instead of "learning programming"), which seems to focus on the second aspect when the first one is essential.

An *algorithm* is a finite and unambiguous sequence of operations, instructions or actions used in order to solve a problem. It should be finite, otherwise the problem will never be solved and it should be unambiguous otherwise we will not know what action to take. We regularly practice one algorithm or another, for example, when we prepare the coffee in the morning, or when our car has a flat tire and we need to change the wheel. A good example of everyday life algorithms are the cooking recipes. Unfortunately, this kind of algorithms is often ambiguous, which explains why we are sometimes disappointed with the meal we have cooked. Algorithmic thinking is, therefore, rigorous thinking, if thinking is wrong, the problem will not be solved.

Most of the time, in French primary schools, the language used to code the program is Scratch. According to its designers (Maloney et al., 2010), "Scratch is a visual programming environment that allows users (aged 8 to 16) to learn computer programming while working on personally meaningful projects, such as animated stories and games. A key design goal of Scratch is to support self-directed learning through tinkering and collaboration with peers". Scratch is the heir of another programming language: Logo, one of the designers of which was Seymour Papert (Papert, 1999) who has worked with Jean Piaget and also Mitchel Resnik. Papert's main idea was to offer children 'tools to think with'.

The advantage of Scratch is that it is possible to get pupils, but also adults, to think about very simple problems. For example (see figure 1), they quickly understand that in order to draw a square, they just have to write the following program: *Repeat 4 times (Move 100 steps; Turn right 90°)*.

But they will take a long time, especially adults, to understand that the program to draw an equilateral triangle

is NOT: *Repeat 3 times (Move 100 steps ; Turn right 60°)*

but: *Repeat 3 times (Move 100 steps ; Turn right 120°)*.

On this example, difficulties are not the same for children and for adults. Children have to find the right rotation angle through trial and error. Adults, in one way, have to forget what they know: "the angles of an equilateral triangle are 60° each". More precisely, they have to realize that the cat does not turn on the internal angle of the triangle. However, in both cases, children and adults have to think rigorously, otherwise the result given by the computer will be wrong.

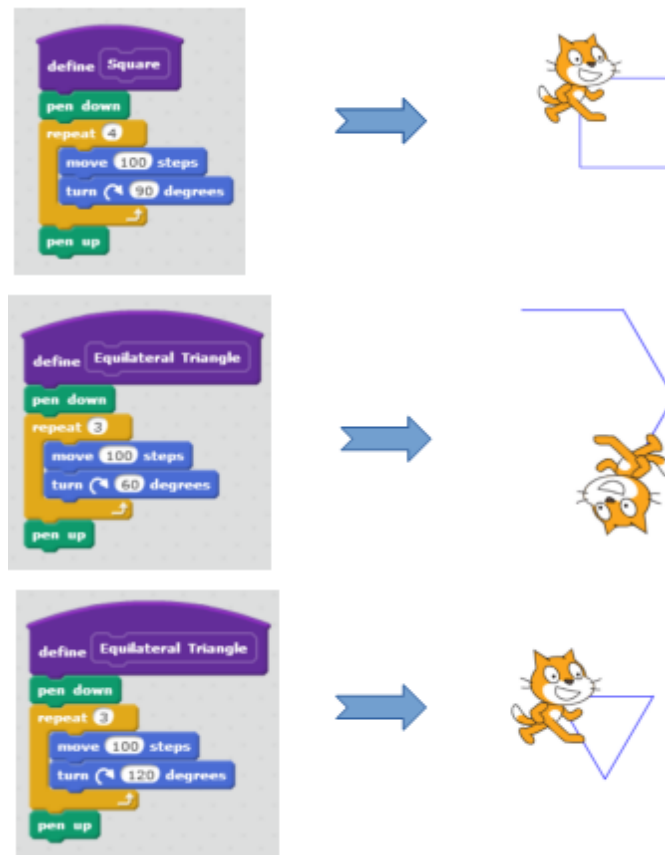


Figure 1: 3 programs written in Scratch and what they give

Learning of Programming by Trainees

Since the reform launched by the Ministry, all trainees of the ESPE have had two hours of training in programming. However, this was more getting information than training. Two other types of training were held, one in Mayotte for all trainees and the other in Reunion Island for a few groups of trainees. It is on these last two types of training that the study described here focused.

Programming training courses

The training courses for programming in Mayotte and Reunion Island do not have the same format:

- For one year, in Mayotte, 6-hour training sessions are held for all trainees where most of the main concepts are seen. This training time allows trainees to understand what programming is, but does not allow them to be autonomous due to lack of practice.
- For the past 3 years, in Reunion Island, 15-hour training sessions for a few groups of trainees where the main concepts are presented and where they are asked to develop a small project.

During these trainings, the trainees are two per machine and the training is based on the resolution of problem situations. Each problem situation leads to learning. For example, when the trainee is asked to build the equilateral triangle, he will learn two things: on the one hand, that he must put himself in the place of the sprite (the cat), and on the other hand, that he must learn to contextualize his previous learning (the sprite does not rotate according to the inner angle of the equilateral triangle). Each

time a pair has solved a problem situation, they move on to the next problem situation. The moments of synthesis and feedback on learning take place each time when all the pairs have completed a set of steps. The teacher walks through the class and helps the trainees who remain stuck for too long on the same problem situation.

We can see that this teaching approach contains the main elements of a socio-constructivist type of learning:

- The student builds his own learning,
- The student learns by solving problem situations. Care is taken to ensure that the proposed problem situation is in the student's zone of proximal development (Vygotski, 1935),
- The student can practice a trial and error approach,
- The student can progress through small socio-cognitive conflicts thanks to his classmates.

Through this type of teaching, we bring trainees to live from the 'inside' what socio-constructivism is. At the ESPE, they have already been taught socio-constructivism. They were explained the main concepts and the importance of this approach. However, most of the time, this teaching does not itself obey a socio-constructivist approach, because it involves a lecture course. This is one of the paradoxes of higher education: is it not contradictory to teach socio-constructivism in a lecture course?

The "feelings" notebooks

Our first objective was, therefore, to teach programming to the trainees. However, our second objective was also to help them understand what their own pupils were experiencing during learning and the benefits of setting up a socio-constructivist type of education. To make them more aware of this, we asked them to keep a "feelings" notebook in which they had to record their feelings as students:

- their successes and their desire to continue and face other problems,
- their failures and anxieties when they saw their colleagues progress while they stayed behind,
- how working in pairs allowed them to solve problems,
- the importance of the teacher's role in providing support neither too early so that there is learning, nor too late to avoid fatigue and demobilization,
- ...

They were also asked to reflect in this notebook on their own class, to transfer what they experienced as students into practices as teachers. These notebooks could be written alone or in pairs. Table 1 shows the number of notebooks obtained and the total number of words contained in these notebooks according to the two types of training.

Table 1 Number of notebooks and words according to the type of training

feelings notebooks	number of notebooks	number of words	average number of words per notebook
Reunion Island	48	15962	333
Mayotte	41	23312	569
Total	89	39274	441

As can be seen, the average number of words per notebook is significantly higher in Mayotte than on Reunion Island. It can be assumed that the duration of the training, which is much shorter in Mayotte, is the main reason.

In the following section, we explain the methodology used to analyze these notebooks.

Methodology

To study these 89 “feelings” notebooks, we used a text-mining software called Tropes (Molette, 2009). The interest of using this kind of software is that it automatically groups terms from the same semantic family. For example, for the field ‘education’, the software will group terms like ‘teacher’, ‘pupil’, ‘student’, ‘learning’, ‘instruction’, ‘training’... Then it returns what its authors call a ‘Universe’ (Figure 2) that is constituted of the most important semantic families appearing in the text with the number of occurrences of the terms of this family. For example, in figure 2 we see that there is 555 words related to “education” in the text analyzed by Tropes. By this way, Tropes allows the researcher to focus rapidly on the main aspects of the text. At the same time, for a given semantic family, the software indicates in another window the extracts of texts in which it has found the terms. This allows the researcher to remove any ambiguity. Indeed, if the software may very well indicate that the terms from the field “motivation” often appear in the text, the interpretation will not at all be the same if the extract is ‘the students were very motivated by the task’ or the opposite ‘the students were not motivated by the task’. At the end, the researcher can return to the original documents to explain this or that aspect.

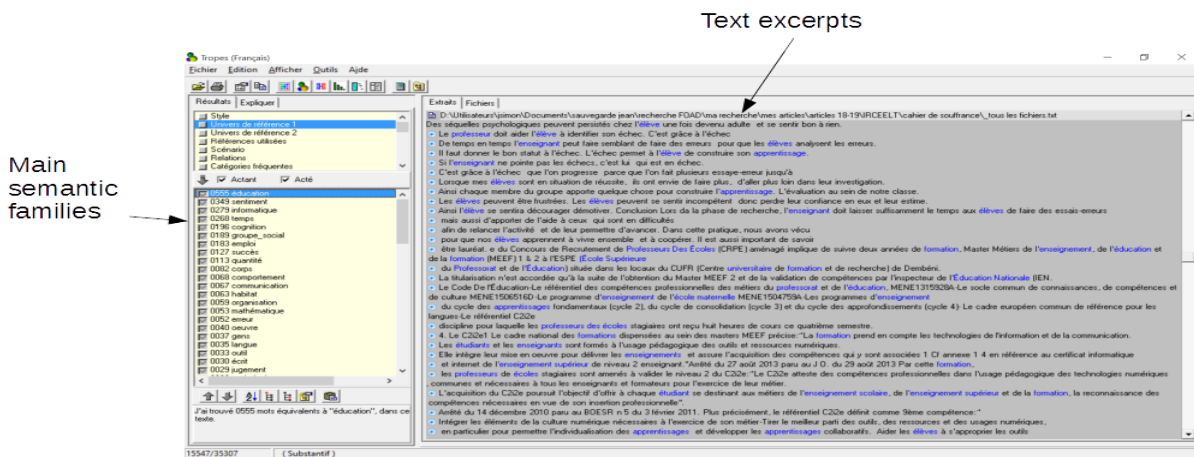


Figure 2: A screen view of Tropes software with the two windows, one for the semantic families and the other for the text excerpts

We analyzed all these notebooks by distinguishing between training in Reunion Island, training in Mayotte and both of them. As it can be seen in Table 1, the length of these notebooks is not the same in the two types of training, so it seemed useful to make a distinction. Tropes has returned three ‘universes’. The ‘universe’ of Reunion Island includes 61 different fields, the one of Mayotte 60 and for the whole 90. If it is not surprising that the whole set includes more fields, it is surprising that the one in Mayotte where there are more words is smaller (60 fields for 23312 words) than the one in Reunion Island (61 fields for 15962 words). This suggests that the ‘feelings’ notebooks in Mayotte are more homogeneous.

We have selected the first 10 most important fields in these three universes in Table 2.

Table 2. The ten first fields of the Universes returned by Tropes

Reunion Island		Mayotte		All notebooks	
Feelings	202	Education	451	education	597
education	146	Time	192	feelings	384
information technology	106	information technology	185	information technology	291
social group	96	Feelings	182	time	279
Task	95	Cognition	115	cognition	206
cognition	91	Task	103	task	198
Time	87	social group	96	social group	192
amount	67	success/failure	77	success/failure	130
success/failure	53	<i>body</i>	65	amount	117
<i>behaviour</i>	49	Amount	50	<i>communication</i>	90

As we can see, Reunion Island and Mayotte share nine fields among ten: ‘education’, ‘feeling’, ‘information technology’, ‘time’, ‘cognition’, ‘task’, ‘social group’, ‘success/failure’, and ‘amount’. Three fields are different: ‘behavior,’ ‘body’, and ‘communication’. These three fields are also fields in the two others Universes but with a lower number of occurrences. These nine same fields show that the trainees understood what they had to do in this notebook. These nine fields highlight learning from a socio-constructivist perspective.

Among these nine fields, “amount” can be related to “information technology” because it refers to numbers used when programming: number of steps, number of iterations and so on. In what follows, we begin with these two fields. Then we take the other fields in descending order and analyze what the trainees write about it.

Information technology (291 terms) and amount (117 terms)

There are 291 computer related terms: 'computer technology', 'computer', 'software', 'programming language', etc. These terms are essentially used to describe the context and purpose of the training: *'Discovery of the programming software'*; *'What we were going to do with the software'*; *'We can even play against the computer'*; *'Scratch is a programming language'*. Similarly, there are 117 terms related to quantity. The use of these terms appears when the trainees want to express the difficulties encountered: *'after having understood the rule for polygons (measuring angles) we were able to realize the requested polygons'*. It is normal for trainees to use these terms to describe the problem situation they are facing. However, what interests us more is how they experienced this problem situation and the transfer of these feelings to their class.

Education: 597 terms

The main terms related to education are: pupils (201: 'pupil'), teachers (167: 'teacher', 'trainer', 'professor'), learning (122: 'learning', 'teaching', 'training'). Far behind we find tutoring (37: 'tutoring', 'tutor', 'tutored'), school (25: 'school') and a few other terms.

The word 'pupil' is the one that appears most often (201 occurrences). The notebook, therefore, achieved its objective in the sense that the trainees did indeed transfer what they were experiencing in training to what their own pupils were experiencing in class: *'The objective is to understand how the pupils experience our teaching. To do this, we will put ourselves in their place and receive training on programming software'*. As we will see further, in a deeper way, they express their joy in succeeding. This success is a source of motivation to continue: *'When my pupils are in a successful situation, they want to do more, to go further in their investigation'*. They also express their fear of failure: *'The pupils, if they were me, would feel the same way as I do, seeing their classmates find the solution easily'*. This failure risks demobilizing them: *'When the pupil is constantly failing, he can no longer hold on'*. They believe that this must be remedied: *'This clearly shows that we put our pupils in a situation of failure if we do not propose pedagogical differentiations with progress paths to allow pupils to progress at their own pace'*. They also think that they must teach pupils to accept their mistakes *'Failure must be given the right status'*. It is also necessary that in the face of the pupil's failure, the teacher must be able to question himself: *'When our pupils are in a situation of failure, the teacher must be able to question himself about his activity'*.

There are three different words used for the term 'teacher': 'teacher' (113), 'trainer' (14), 'professor' (40). These words are the most frequent after 'pupil'. The teacher is not only the one who defines the task, but also the one who provides help: *'the professor came to help us'*; *'We had to ask the teacher for help'*; *'The pupil must ask for help at times to avoid getting stuck for too long'*. Nevertheless, this help can have limits: *'The trainer was helping us, but at one point, we wanted to give up. Not succeeding annoyed us'*. The trainees also became aware of the importance of the timing of the shoring: *'If I give him the solution too soon, the pupil will not make any effort to think about it, but if I give him the solution too late, he will have already abandoned and will no longer want to go through with the problem'*.

122 words refer to teaching: 'teaching' (25), 'learning' (49) or 'training' (48). However, these terms are most often used to define the framework of the training or the trainees' own lessons: 'learning situation', 'learning session'. When it comes to reflecting about 'learning, it is for thinking on what they have experienced themselves:

- the need for oralization: *'Make pupils verbalize the different steps to give meaning to their learning'*; *'he sees his learning confirmed by reformulation'*;
- the advantage of working in pairs: *'Working in pairs helps learning because when one does not understand, the other can give us explanations without recursing to the teacher'*; *' So each member of the group brings something to build the*

learning; *develop collaborative learning*'.

The word 'teaching' is in relation with the notion of curricula (in French 'programme d'enseignement'): *'The curriculum of the nursery school'*; *'the curricula of the cycle of fundamental learning'*. It can be also in relation with the notion of teaching computer science: *'sometimes forced to leave out computer science teaching'*; *'programming is a real challenge in teaching'* or with technology-enhanced learning: *'to build, implement and animate teaching situations integrating digital technologies and resources'*. When trainees use the term 'training', it is only to refer to their own training: *'What was our experience during these 6 hours of training'*.

The reflection on working in pairs continues with the terms 'tutor' (20), 'tutoring' (15) or 'tutored' (3). If the tutor can be one of the most comfortable students in the discipline, *'thinking about tutoring, especially for the most advanced pupils who could assist the teacher'*, tutoring is not something that happens immediately, *'I try to set up a tutoring system with my pupils. This does not always work'* and it can be necessary to consider training the tutor: *'training them to be a tutor. Teaching them what is expected of them and how to do it'*.

The last term of a certain weight is 'school' (25). Most of the time, the word is associated with the name of trainees known as 'school teachers'. When it comes to the school itself, it is to indicate the lack of computer equipment, particularly for trainees in Mayotte: *'few schools are equipped with computer equipment'*; *'by providing schools with more computer equipment'*.

Table 2 shows that the term 'education' comes first for trainees in Mayotte and second for those in Reunion Island. It is used 3 times more in Mayotte than in Reunion. In Mayotte, trainees place a greater emphasis on training, tutoring and school and in Reunion Island - on the respective roles of teachers and students. The difference is easily explained for tutoring and school. As the training in Mayotte was very short and several trainees did not attend the first course, a tutoring system was set up. In addition, the schools in Mayotte are insufficiently equipped with material and in particular with computer equipment.

Feelings (384 terms)

To analyze this field we have grouped the different terms that can be divided into 4 categories (apart from ambiguous words (67)): feelings of fear (43 words), negative or painful feelings (83 words), positive feelings (147 words), feelings about the other (44 words).

Ambiguous words are those that only make sense in context and can be positive (*'feeling proud'*; *'desire to continue'*) or negative (*'feeling ashamed'*; *'desire to stop'*) or both (*'we have gone through all the feelings from joy to disappointment through pride in success'*).

The feelings expressed related to fear surprised us even if, in hindsight, they are easily understandable. In this category of 43 terms, we find words such as: 'anguish', 'anxiety', 'apprehension', 'panic', and 'fear'. To understand these fears, it should be remembered that the trainees are the students who have successfully passed the competition to become primary school teachers. As a result, most of them have always had a 'good student' profile. This new discipline frightened them because they feared failure, something they rarely experienced in school: *'But no, it's the fear of making mistakes!'*; *'I was afraid I couldn't do it'*. However, when we go back to their writings, we realize that this feeling of fear often appears before or at the beginning of the training: *'From the beginning we had some apprehensions, we told ourselves that we are not good at computers'*; *'He accentuated our fear, when he talked about programming'*. Most of the time, this feeling disappeared afterwards: *'This anguish was ultimately ephemeral'*; *'I was apprehensive. Once we started, programming was easy. It was just a matter of finding the right approach'*.

Negative or painful feelings (83 occurrences) were: 'disappointment', 'discouragement', 'shame', 'suffering', 'frustration', 'horror', 'grief'. The most frequent word was 'frustration': 51 occurrences, 13% among words expressing a feeling and 61% among words expressing a negative feeling. This word is very often associated with failure: '*Frustration and failure*'; '*we will first see moments of frustration and failure*'; '*This failure creates a feeling of frustration, loss of confidence and self-esteem*'. Fortunately, these moments of frustration alternate or are followed by moments of satisfaction or joy when they manage to overcome the obstacle: '*Feelings of satisfaction and frustration are intertwined*'; '*I find that with programming failure and frustration do not last, they are quickly replaced by feelings of joy and success*'. Some trainees do the transposition with what happens in their classrooms: '*It is also a way of putting ourselves in the pupils' shoes and understanding their frustrations when they fail an exercise*'; '*wanting to help them progress quickly or at our pace is only a source of frustration and demotivation for our pupils*'. The word 'suffering' is also present (15 occurrences). Suffering is seen in a positive light: '*It is a deep feeling like a cry of suffering, but it encourages us to move forward and not to admit that we have been defeated by multiple attempts*'. It is seen also in association with what pupils can experience: '*This session allowed us to learn how to use the software but also to understand the suffering that pupils can experience during a learning session*'. The other words used are much rarer and are part of an expression. Thus, a trainee quotes himself when he sees his mistake: '*I quickly told myself: ah no, horror!*'

Positive feelings (147 words) are expressed by the words: 'contentment', 'enthusiasm', 'joy', 'satisfaction', 'relief', 'pleasure', 'laughter', 'beloved', 'good humour', 'pride', 'hope'. The four most common words are 'joy' (43 occurrences), 'pleasure' (34 occurrences), 'satisfaction' (20 occurrences) and 'pride' (12 occurrences). They mirror the previous paragraph and express what the trainees felt when they succeeded, when they overcame their failure: '*self-confidence, pride, joy at success*'; '*when I managed to take a step forward, I felt a satisfaction, a tremendous joy*'; '*at the success of these productions, I shouted with joy cheerfully I felt better and it encouraged me to continue*'; '*what a joy when we understood the logic calculation to do*'. However, in the same way that frustration did not last, joy could have an end with a new problem to solve: '*Unfortunately the joy was short-lived since we encountered a difficulty again*'.

Moreover, this feeling of satisfaction, of joy, is not only due to the success of the task but also in relation to the successes of others:

- by feeling at the same level as them: '*In my place, the pupils would have this joy because they would feel at the same level as their peers*'.
- but also sometimes with a certain spirit of competition: '*We also have another feeling that appears, it is the fact of getting there but faster than others, thus providing a secret personal joy*'.

This is how we get to the relationship with the other. The 44 words that refer to the other or the relationship with the other are: 'affinity'; 'friend'; 'classmate'; 'trust'; 'sympathy'; 'respect'; 'comfort'; 'jealousy'. The word most often used is 'classmate' (17 occurrences). These words are often used to express the help the other person provides: '*I had to ask a friend for help to do this*'; '*by exchanging with a classmate, we can acquire notions that we had not been able to understand*'. However, this term is often used by trainees to reflect on the possible mutual support between pupils in their own class. They need to make them understand that they must listen to the other, accept what they propose: '*be able to listen to their classmate and respect their opinion*'; '*with programming I think it is necessary to work in affinity groups to be able to accept that their classmate tries and do not stay focused on themselves*'. This reflection is sometimes even more profound: '*The effort made by the latter to come back to what his classmate did not understand by reformulating it, by producing explanations corresponds to a double complementary work of transmission and translation*'.

The second most commonly used word is 'trust' (17 occurrences) implicitly or explicitly this term refers to self-confidence and goes hand in hand with self-esteem. This self-confidence can be lost by trainees, *'This failure creates a sense of frustration, loss of confidence and self-esteem'*, or regain it *'self-confidence was regained when defining the algorithm of the rosace whose logic was obvious'*. Very often, they do the transposition with their class: *'Pupils may feel incompetent, therefore lose their self-confidence and self-esteem'*; *'knowing what they have to do, pupils will have feelings of trust and motivation'*.

The other words appear less frequently. The word 'jealousy', for example, appears only once and without context. It is probably related to the fact that any success or failure is assessed in relation to the success or failure of others.

If we compare the productions of the trainees in Reunion Island and those of the trainees in Mayotte, it is mainly on the feeling of "fear" that they differ. Most of the words referring to fear or anxiety are found in the reports of Mayotte trainees: 30 out of 43. We can suppose this is in relation to the very short time of their training, 6 hours, whereas the one in Reunion Island was 15 hours. This too short time did not allow them to master programming and it made them anxious.

Time (279 terms)

As we can see, with 279 terms, time is very present in the trainees' notebooks: in second place in Mayotte and in 7th place in Reunion Island. The main words used are: 'time', 'hour', 'day', 'year', 'days of the week'... Very often these words are used to contextualize their feelings about the training: *'I approached the digital sessions of this semester with apprehension'*; *'at this time of the day, my head, my brain and my concentration were not supposed to be at their best'*.

Sometimes, trainees make comments about time in their learning: *'But my colleague took the time to explain it to me'*; *'It would have taken a little longer to understand the concept well'*; *'We worked for two hours without dropping out, which is exhausting'*; *'it remains a pleasure, you don't see time passing'*. This could lead to reflections about the pupils' academic rhythms: *'During these days, how to organize learning, so that the pupils' receptivity is optimal?'*; *'this time, which we do not necessarily take, and which we should leave to our pupils'*.

However, this was also to indicate that the training in Mayotte was too short. Indeed, for institutional and cost reasons, it was organized in 6 hours over two days: *'Can we follow in 6 hours of concentrated courses in 2 days?'*; *'why were the six hours of classes not spread over the whole week or three different weeks to allow us to assimilate the different processes?'*; *'our only regret is the lack of time'*; *'earning to program in six hours with several notions, towards the end of the course, I felt that the brain was reaching saturation point'*.

Cognition (206 terms)

Among the 206 words used for the 'cognition' field are: 'idea' (36 occurrences); 'competence' (28); 'notion' (26); 'logic' (24); 'knowledge' (22); 'reflection' (18); 'consciousness' (9); 'understanding' (8); 'capacity' (8); 'reasoning' (5); 'memory' (5); 'thought' (3); 'imagination' (3); 'concept' (2); 'common sense' (2); 'forgetting' (2); 'hypothesis' (2); 'ignorance' (2); 'memory' (1); 'ignorant' (1); 'misunderstanding' (1). The word 'idea' comes first because it is usually associated with the idea that allowed solving the problem: *'From the beginning I had the idea that it was necessary to add a means to rotate the triangle'*. The term is also often used in the context of the confrontation of ideas: *'Working in pairs has allowed us to share our ideas, to confront our points of view'*. The trainees also mentioned the difficulty of bringing their own pupils to this confrontation of ideas: *'As I noticed during the internship, some pupils were unable to work together... with their young age it was probably difficult to impose their own ideas or to follow those of the other'*.

In the context of these writings, the terms 'competence', 'notion', 'knowledge' are very similar. We found them 76 times in the reports: 'on the new skills I have developed regarding this software'; 'I already had some programming skills during my university studies in economics'; 'the last two hours of the course, we had a lot of skills that we had to learn'; 'this has allowed us to acquire programming knowledge'. Trainees also refer to the fact that working in Scratch involves skills that are transversal to other disciplines: 'we work on several skills at the same time (French, numeracy, geometry, sense of direction'; 'it also allows us to work in a multidisciplinary way and to develop transversal skills'.

Other terms such as 'reflection', 'logic', 'reasoning', 'common sense' (49 terms) refer to the reflection inherent in programming: *'We are asked to reflect more acutely on the problem situation'; 'I understood the logic of the process'; 'so this work requires logic'. Very often, this need to think more rigorously appears after a failure: 'We made a mistake, but we managed to make the first program, we needed more distance and reflection to understand'; 'the other tasks required more reflection because we did not find a follow-up. Our trials have led to errors invalidated by the software'; 'Programming requires a high capacity for concentration and reflection that can be demotivating in the face of feeling of failure'.*

Some trainees highlight the interest of programming to encourage their pupils to think: *'Finally, pupils gradually enter into a logic of algorithmic creation by trial and error. The pupils in Cycle 3 are encouraged to develop their sense of rigor and a taste for reasoning'; 'it was a very interesting work and rich in reflection and research. We are therefore thinking of implementing it in our classes with our pupils'.*

They sometimes ask themselves the question of the teacher's support in this process of reflection: *'If I give him the solution too soon, the student will not make any effort to think, but if I give him the solution too late, he will already have dropped out and will no longer want to go through with the problem'.*

The term "understanding" refers rather to the understanding of the situation, the type of software in the *instruction, etc. 'Programming requires knowledge and understanding of specific codes'; 'difficulties in understanding the instructions'; 'a sense of failure due to lack of practice and understanding'.*

The term 'awareness' was of particular interest to us: *'awareness of everything we have learned'; 'awareness that we learn better on our own and when we are motivated'. More than any other term, this term could indicate that trainees had transferred what they were experiencing in training to what their own pupils were experiencing in their classes: 'Awareness of how pupils experience our teaching'; 'they have made us aware of the importance of programming with our pupils'; 'therefore, I became aware that when an activity seems difficult to do, we can propose to divide it into several sub-activities. Build on what has already been achieved and improve it. Always come back to what the child has already understood in order to make it evolve'; 'it made us realize that when students are in difficulty, leaving them too long without help can be negative'.*

Task (198 items)

The two main words used were 'work' (140 occurrences), 'task' (44). Very often, they were simply used to indicate the work to be done: 'Although the work was very interesting but at certain times we were tired and we started to get bored'; 'we were proud of ourselves and our work'. At other times, it is used to indicate the complexity of the requested work: 'The task is quickly becoming more complex for us'; 'the task was simple and allowed me to actually tell myself the programming actually it's pleasant'.

They could also include comments on group work: *'This was possible thanks to effective group work'; 'group work is very satisfying because it allows us to have a first validation of ideas'; 'group work was done. Our team has been complementary despite the various failures we have experienced'; 'the cooperative work has worked well; we have shared the tasks, which has*

allowed us to move quickly in the development of our game'. *These remarks were not without limitations: 'My analysis of the group work during these programming sessions was quite mixed. Indeed, first of all, it provides a feeling of frustration, when the two brains are not connected and both do not see the final result of each other's thinking'*. Therefore, when trainees refer to group work in the classroom, they point out the need for a certain complementarity between group members: 'So during the experiment, I was able to confirm my ideas on group work and put myself in the shoes of my pupils. I have seen how difficult it is to work with someone who does not have the same logic or the same way of thinking. Conversely, if brains converge, group work is only more productive'; 'group work is beneficial work but it must be thought of beforehand. As I noticed during the internship, some pupils were unable to work together.'

Social group (192 items)

This field is a natural continuation to the preceding 'group work'. It is all about the words 'peer' (138 occurrences), 'colleague' (44), 'cooperation' (5), 'collaboration' (5). It is very often referred to as the help that the other can provide: '*colleagues come to help us*'; '*we asked for the help of other colleagues*'. Help can also come from the trainer: '*When I saw the others working, my partner took over and with the help of the teacher, we succeeded without pride*'.

The trainees insist on the necessary complementarity between the members of the pair: '*Our pair has been complementary despite the various failures suffered*'; '*My partner has completed the program by inserting the variables. We were then complementary. Everyone brought a touch to build the blocks*'; '*it was very interesting to work in pairs, because we could share our knowledge, experiences*'; '*I think it is really better to do some work with a pair since we always learn a little from our classmate*'; '*working in pairs helps learning because when one does not understand, the other can provide explanations without having to resort to the teacher*'.

This complementarity can moreover be in the form of tutor/tutored: '*My colleague explained to me with precious advice that allowed me to better understand the lesson of the day*'; '*My partner helps me a lot because she understands certain things faster than me*'.

The social group often goes beyond the pair to get help from colleagues in the other pairs: '*colleagues helped each other beyond the previously formed teams*'; '*our colleagues next door allowed us to look at their work a little bit to get inspired or even copy stupidly*'.

The group's objective in all cases is to overcome failure: '*We can move from failure to success through cooperation*'; '*However, it was an enriching work in terms of knowledge and above all collaboration because when faced with an obstacle, the fact of being in pairs allowed us to return to work from a new angle and with trial and error, to succeed in completing the programming*'. It also made it easier to accept the failure: '*Responsibility and shared failure weighs less on our shoulders*'.

There is also and necessarily an emotional dimension to working together: '*At this moment my frustration is enormous, I don't want to work with Henri*'; '*my illustrious partner Virginia (whom I succeeded in taking from Guillaume)*'. This emotional dimension also appears in the support: '*The pairs thus made have also made it possible to support each other in moments of uncertainty*'; '*fortunately, we were in pairs and we tried to motivate and encourage each other*'; '*by working with a partner I feel less alone*'.

Some trainees report that what they have experienced has triggered a transfer to their class: '*Group work is enriching, productive and beneficial when the pair is complementary. Finally, having experienced mentoring myself, I was able to intellectualize the process that takes place on both sides. As a result, I am now able to implement it effectively in the classroom. By that, I mean,*

choosing pairs intelligently; 'working in pairs is used to help pupils establish and facilitate positive interpersonal relationships'; 'working in pairs is a way of comforting our pupils'. This transfer may even concern a detail: 'It is therefore important in a pair that both students have access to the mouse'.

Success/failure (130 items)

The terms used are 'failure' (63 occurrences); 'success' ('succès' in French, 56 occurrences); 'success' ('réussite' in French, 7 occurrences); 'victory' (1 occurrence). We see that in terms of number of occurrences, there is a certain equality between failure (63 occurrences) and success (64 occurrences) even if some uses indicate a negation: 'not successful'. For us, it was important for the trainees to think about this relationship to failure because a good part of them have been during their schooling what can be called "good students", not really familiar with failure and what it can mean for their pupils: *'that feeling of failure that I am not necessarily used to encounter'*. Thus, some discover the difficulty of accepting failure: *'Failure creates a feeling of frustration, loss of confidence and self-esteem'; 'during the programming tests I was confronted several times with situations of failure that caused frustration to appear in me'.*

This leads them to reflect on the role of failure and its overcoming in learning (often cited): *'From this perspective, we are not afraid of failure. Error allows us to learn'; 'It is through failure that we learn'; 'it is through failure that we progress because we make several trial and error until we find the right result'; 'we learn from mistakes'; 'but sometimes it is with failures that we manage to reorganize ourselves in a more effective way to avoid mistakes'.*

As we have already seen, failure is not permanent: *'Nevertheless, I find that with programming failure and frustration do not last, they are quickly replaced by feelings of joy and success'*. The role of the teacher is also important in getting out of the situation of failure: *'failure is put into perspective when the teacher offers help in case of difficulties'*.

On the other hand, the trainees highlight the motivating effect of success: *'This success has motivated us and given us a boost'; 'when these productions were successful, I shouted with joy and cheered for it, I felt better'; "In these moments of success, we were very happy and proud of ourselves'*. Something that is rarely reported or only implicitly reported is that it is the machine that validates: *'This feeling of success is immediate because you get the validation the instant the coding is correct'*.

It is in this pass/fail field that trainees make the most transfers toward their classes. As before, we will not give all the quotes, but we think it is important to focus on this aspect. The first remarks concern the negative experience that the student may feel in the event of failure: *'The experience allowed me to find myself in a position of failure and therefore to put myself in the position of my failing pupils. This feeling is not pleasant and is quite frustrating'*. This failure can even go beyond simple frustration: *'failure can sometimes be devastating for some children. Psychological sequelae may persist in the pupil as an adult and feel good for nothing'*.

If the pupil is left in this situation, he may give up: *'when the pupil is permanently in failure, he cannot hold on'*. However, *'failure allows the student to build his learning'; 'it is through failure that we learn'*. The trainees thus point out the essential role of the teacher: *'The master's support is essential. A pupil must not stay in failure for too long or he may drop out'*.

The teacher must therefore work with pupils on their mistakes. First, by identifying them: *'the teacher must help the pupil to identify his failure'*. Then by reducing its emotional impact: *'Failure must be given the right status. Finally, by giving the tools to overcome failure: 'Do not let a pupil fail: give him as many tools as possible to succeed'*. Sometimes the failure of the pupils must lead the teacher to question himself: *'Our pupils when they are in a situation of failure, the teacher must be able to question himself'*

on his activity'. On the other hand, they also talk about the role of success for the student: '*When my pupils are in a successful situation, they want to do more, to go further in their investigation*'.

Discussion

The use of a notebook to encourage students to reflect on their learning is frequent particularly in e-learning. For example, Denis (2003) works on the tutor/tutored relationship in distance learning. He suggests that students keep an up-to-date notebook to self-regulate their learning. This notebook is also used by the tutor to encourage them to reflect on their actions. In Charlier and Henri (2004), they study hybrid training and the establishment of learning communities. The logbook must make it possible to identify the conditions that facilitate or hinder learning and the conditions for transferring this learning to other objects...

The notebook is also used in teacher training (Korthagen, 1999). In this type of training, the trainee often considers that there is a gap between theory and practice, between what he learns about teaching at the University as a student and his teaching practice when he is on an internship (Louden & Rohl, 2006). Thus, the notebook will be used to lead the trainee to a metareflection (van Halen-Faber, 1997). The objective will be to bridge this gap between theory and practice (Allen & Peach, 2007).

More broadly, research on this topic can be divided into two large categories, according to the purpose of the meta-reflection. On the one hand, those where the notebook will be used to encourage the student to reflect on his own learning process concerning a particular subject. On the other hand, those where the notebook will be used during the internships to encourage the trainee to reflect on his own teaching approach.

In the first case, the notebook is often used in science learning or in science education learning. In Alayyar, Fisser, & Voogt (2012), students are asked to describe the problems they have encountered in learning science pedagogy, how and with whose help they have solved them. In Morrison (2008), primary school teachers used notebooks to reflect on all aspects of science and scientific research. They have come to view the workbooks as a formative evaluation tool.

In the second case, the trainee is required to keep a notebook during the internship (Allen & Peach, 2007; Johnson, 1996; van Halen-Faber, 1997). The trainees note any significant elements in the life of the class. These notebooks are called 'practicum book' (Allen & Peach, 2007), 'practice-teaching logbook' (van Halen-Faber, 1997). These notebooks should allow for critical reflection (van Halen-Faber, 1997). The practicum notebook provided the framework to reflect on the knowledge the trainee had used in class (Allen & Peach, 2007). This reflection can be conducted with the help of a supervisor (van Halen-Faber, 1997) or in a peer group (Daniel, 1999).

For our part, our objective is also a metareflection, but not in the same way as in those previous works. We differ from the first category, because we do not ask trainees to reflect on learning programming, but on learning in general. We will observe below that this objective was not achieved by all trainees; some of them focused their thoughts only on learning how to program. We differ from the second category for two reasons. The first one is that they must keep their notebook while they are at the ESPE and not during their internship. The second is in the nature of the theoretical-practical link that we are aiming for. The idea is that they live this theory as a pupil and not as a teacher. The objective is to make them aware of the difficulty of the pupil's 'profession' and to see how the way in which they teach can alleviate these difficulties. This awareness should then lead them to make a transfer to their own class and pupils. This was not the case for all trainees as well.

We have seen that trainees have clearly identified what happens in a learning situation and the main elements of this learning which are found in socio-constructivism. The pupil builds his own learning. Learning can be done by problem situations;

the problem situations must be in the pupil's proximal development zone. To solve the problem, the pupil can practice a trial-and-error approach. The pupil also progresses through the confrontation of points of view with his classmates. In their writings, the trainees do not theorize these different points. They describe how they experienced them, they lived them, and, in this sense, we have achieved our objective.

However, it is important to indicate the limits of our restitution process. Whenever we have highlighted a point, we have cited a few trainees in support of this point; we have not cited all the trainees who have spoken about this point. For example, in the last point of the previous section, 'success as a driver of motivation', we cited only one trainee when several of them mentioned this point. At other times, when we quoted several trainees it was because these quotes introduced slight differences.

Furthermore, if there were several of them reporting a point, not all trainees reported it. *In other words, trainees, altogether identified most of the key aspects of a learning situation, but not each trainee individually did.* To illustrate this, it is sufficient to take as an example the word 'failure' that we previously indicated that it appeared 63 times in all the texts. Knowing that there were 89 texts (Table 1), this means that 26 trainees do not refer to this problem, and probably more, because this term may have been used several times by the same trainee.

In the same way, even if the characteristics of a learning situation have been well reported as a lived experience, not all trainees systematically transfer this experience to their class. The word 'pupil', for example, appears 200 times, but it only appears in 51 notebooks among 89. In other words, at least, 38 trainees do not refer to their own pupils.

Moreover, some notebooks remain very close to what is experienced as learning of programming and do not go towards an abstraction on learning situations in general. For example, the trainee explains at length why he did not find the right angle of rotation to draw a triangle and how he managed to discover the solution. In other words, he focuses on the problem situation he is experiencing, without generalizing to problem situations in general, even less to learning, and not at all to the learning of his own pupils.

However, even if some trainees do not refer to their pupils in their notebooks, it does not mean that they will not transfer what they have experienced toward their class. This is the bet we made here. The idea is that making trainees live a completely new learning experience, and having them point out the main characteristics in a notebook so that they are well aware of them, is complementary to their lecture courses on learning.

Regarding our analysis approach, the method, relying on a text mining tool that we used to study the texts saved time and allowed us to quickly focus on the main themes that appear in the documents. As we have seen, the 'universes' proposed by tropes are consistent with what we expected. The software immediately identified it as education and computer science. It also identified that it was about feelings. More generally, the text-mining software is increasingly used in research, particularly for question reviews (ananiadou et al., 2009; hung, 2012; Thomas, mcnaught, & ananiadou, 2011). They are also used in learning analytics, in particular to study student behaviour through their productions on distance learning platforms (blikstein, 2011). However, in the case of results subject to interpretation, it is advisable to return to the original writings to remove any doubts.

Conclusion

In this article, we have analyzed the 89 "feelings" notebooks that trainees kept during their programming learning courses at the teachers' training school of Reunion Island and Mayotte. This analysis was carried out, using a text-mining tool Tropes. It was quantitative and qualitative.

The analysis shows that most of the key aspects of a learning situation were identified by the whole group of trainees, but not by each trainee. Moreover, some trainees focus on the problem situation they are experiencing, without generalizing to problem situations in general, even less to learning, and do not do any transfer to the learning of their own pupils.

Finally, it should be noted that these notebooks are a formative evaluation for the trainer himself and allow him to adapt his teaching. They also constituted a formative evaluation for the institution: the result of this study led to the duration of the training in Mayotte being increased for the coming years.

References

- Alayyar, G.M., Fisser, P., & Voogt, J. (2012). Developing technological pedagogical content knowledge in pre-service science teachers: Support from blended learning. *Australasian Journal of Educational Technology*, 28(8), 1298-1316.
- Allen, J.M., & Peach, D. (2007). Exploring connections between the in-field and on-campus components of a preservice teacher education program: A student perspective. *Asia-Pacific Journal of Cooperative Education*, 8(1), 23–36.
- Ananiadou, S., Rea, B., Okazaki, N., Procter, R., & Thomas, J. (2009). Supporting systematic reviews using text mining. *Social Science Computer Review*, 27(4), 509–523.
- Blikstein, P. (2011). Using learning analytics to assess students' behavior in open-ended programming tasks. *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 110–116. Retrieved from <http://dl.acm.org/citation.cfm?id=2090132>
- Charlier, B., & Henri, F. (2004). Démarche d'évaluation, communauté de pratique et formation professionnelle. *Schweizerische Zeitschrift Für Bildungswissenschaften*, 26(2), 285–304.
- Daniel, M.-F. (1999). P4C in pre-service teacher education: difficulties and successes encountered in two research projects. *Analytic Teaching*, 19(1), 15-28.
- Denis, B. (2003). Quels rôles et quelle formation pour les tuteurs intervenant dans des dispositifs de formation à distance? *Distances et Savoirs*, 1(1), 19–46.
- Hung, J. (2012). Trends of e-learning research from 2000 to 2008: Use of text mining and bibliometrics. *British Journal of Educational Technology*, 43(1), 5–16.
- Johnson, K. E. (1996). The role of theory in L2 teacher education. *TESOL Quarterly*, 30(4), 765–771.
- Kaila, E., Laakso, M. J., & Kurvinen, E. (2018). Teaching future teachers to code—Programming and computational thinking for teacher students. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 0677–0682. IEEE.
- Korthagen, F.A. (1999). Linking reflection and technical competence: The logbook as an instrument in teacher education. *European Journal of Teacher Education*, 22(2–3), 191–207.
- Louden, W., & Rohl, M. (2006). "Too many theories and not enough instruction": Perceptions of pre-service teacher preparation for literacy teaching in Australian schools. *Literacy*, 40(2), 66–78.



- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education, 10*(4). Retrieved June 10 from <https://doi.org/10.1145/1868358.1868363>
- Molette, P. (2009). De l'APD à Tropes: comment un outil d'analyse de contenu peut évoluer en logiciel de classification sémantique généraliste. *Communication Au Colloque Psychologie et Communication*. Retrieved June 10, 2019 from <https://www.tropes.fr/PierreMolette-CommunicationColloquePsychoTarbesJuin2009.pdf>
- Morrison, J. (2008). Elementary pre-service teachers' use of science notebooks. *Journal of Elementary Science Education, 102*(2), 123-140.
- Papert, S. (1999). What is Logo? Who needs it. *Logo Philosophy and Implementation*. Retrieved June 10, 2019 from <http://www.microworlds.com/company/philosophy.pdf>
- Rees, A., García-Peñalvo, F.J., Jormanainen, I., Tuul, M., & Reimann, D. (2016). *An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers*. Retrieved June 10, 2019 from <https://repositorio.grial.eu/bitstream/grial/688/1/TACCLE3O5Literaturereview%20-%20final.pdf>
- Rushkoff, D. (2010). *Program or Be Programmed: Ten Commands for a Digital Age*. New York: OR Books.
- Thomas, J., McNaught, J., & Ananiadou, S. (2011). Applications of text mining within systematic reviews. *Research Synthesis Methods, 2*(1), 1–14.
- van Halen-Faber, C. (1997). Encouraging critical reflection in pre-service teacher education: A narrative of a personal learning journey. *New Directions for Adult and Continuing Education, 1997*(74), 51–60.
- Vygotski, L.S. (1935). *Pensée et langage*, trad. F. Sève, Paris: La Dispute.