PIAF: Developing Computational and Algorithmic Thinking in Fundamental Education

Yannick Parmentier[3,5,6], Robert Reuter[1], Sarah Higuet[2], Lara Kataja[4], Yves Kreis[1], Marie Duflot-Kremer[3], Christophe Laduron[2], Christian Meyers[1], Gilbert Busana[1], Armin Weinberger[4], Brigitte Denis[2]

[1] Department for Education and Social Work, Université du Luxembourg, Esch-sur-Alzette, Luxembourg
{gilbert.busana, yves.kreis, christian.meyers, robert.reuter}@uni.lu
[2] CRIFA, Université de Liège, Liège, Belgique
{b.denis, sarah.higuet, christophe.laduron}@uliege.be
[3] LORIA, Université de Lorraine, Vandoeuvre-Lès-Nancy, France {marie.duflot-kremer, yannick.parmentier}@loria.fr
[4] Department of Educational Technology, Saarland University, Saarbrücken, Germany
{a.weinberger, l.kataja}@edutech.uni-saarland.de
[5] LIFO, Université d'Orléans, Orléans, France
[6] INSPE de Lorraine, Université de Lorraine, Maxéville, France

Abstract: In this article, we present the objectives and first achievements of the PIAF project supported by the European Union and aiming at developing computational and algorithmic thinking in basic education. This project brings together researchers in educational sciences and computer science from four countries (Belgium, France, Germany and Luxembourg) around the theme of teacher training. More concretely, the aim is to define a framework (competency framework, pedagogical scenarios) enabling teachers to (i) appropriate the concept of computational and algorithmic thinking and (ii) implement learning activities that promote the development of this kind of thinking in children.

Introduction

Digital tools are becoming more and more important in our daily lives, whether for our purchases, for administrative procedures, in a professional context, to travel or plan our activities.

A certain mastery of these tools is becoming necessary for every citizen. Based on this observation, Luxembourg, France and Germany, like other countries, have given a relatively important place to the acquisition of digital skills in the training of future citizens, as evidenced by adaptation of school curricula for several years now (Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherché, France, 2015; 2019). Despite this favourable context for the development of digital skills in children, the implementation of learning activities using digital tools (either as instruments or as objects of study) encounters several difficulties related to various obstacles, mainly technical (such as the durability of equipment) or psychological (such as a feeling of insecurity) (Baron & Boulc'h, 2011). Advances in the material field (lower costs, simplification of interfaces, generalisation of mobile equipment) suggest that the technical obstacles will diminish. When it comes to the more psychological obstacles, studies have shown that an effective lever to combat them lies in the training of personnel (Delarbre, 2017).

If we look at training in the educational uses of digital technology, we see that resources are relatively rich and freely accessible, and that there are many training opportunities. Furthermore, a certain number of staff have been trained, 63% of respondents to the PROFETIC 10 survey in France (DANE, 2016). However, there is still a need (felt or real) for training, even among staff who have already been trained, 48% of respondents in the same survey (DANE, 2016). This suggests that resources and training, in their current form, do not make it possible to turn teachers into empowered educational technology users. In spite of training, digital technology seems to remain a complex field, even to a certain extent inaccessible.

In order to combat these prejudices (and more generally to prepare future citizens to live in a hyper-connected world), we propose to train teachers in the development of computational thinking from an early age. We here adopt the definition of Wing (2006), which states that "[c]omputational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system; [...] It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable."

Concretely, we propose

(i) to develop a new training framework (based on a competencies framework) that is both interpretable by non-specialists (primary school teachers, etc.) and scientifically founded (i.e. in line with the foundational theoretical aspects of the discipline "digital and computer sciences") and

(ii) to collaboratively develop pedagogical scenarios illustrating in a simplified (but well-founded) way how to work towards the targeted competencies from an early age.

A particular challenge here is to show that these scenarios (and competencies) are accessible to a diverse audience (adults or children).

The PIAF project

Funding

The PIAF (Pensée Informatique et Algorithmique dans l'enseignement Fondamental) project is funded under the European Commission's Erasmus+ programme (https://ec.europa.eu/programmes/erasmus-plus/). Erasmus+ is a programme in favour of education, training, youth and sport. A particular aspect of this programme is that it is aimed at both individuals and organisations (universities, training structures, research institutes, etc.). For the latter, three key actions are proposed:

1. Mobility of individuals for education and training purposes
2. Innovation and good practice
3. Support for policy reform

Each of these key actions includes a number of possible types of action, such as Key Action 2:

- Strategic Partnerships
- Knowledge Alliances
- Sectoral Skills Alliances
- Capacity Building

The PIAF project was selected under Key Action 2 - Strategic Partnerships. It will run for three years (September 2018 to August 2021).

Partners

The consortium brings together, in close collaboration, partners (teachers and researchers) from education and computer sciences (basic and applied), from different countries, cultures and educational systems, and includes in a participatory way, a relatively large number of field staff (teachers and future teachers).

The richness of this consortium aims:

- to promote a transdisciplinary vision of computational thinking,
- to foster cooperation between countries,
- to bring out innovative pedagogical practices and scenarios, in particular through strong interaction between pedagogues and computer scientists.

Project management

The PIAF project is steered by a committee made up of members from each of the four countries involved and chaired by the project's lead country. This committee meets by videoconference on a regular basis (approximately every six weeks), to plan the current tasks in accordance with the provisional calendar. A face-to-face meeting is organised every four months, in turn, by each of the partners, in order to take stock of the project's progress and to settle outstanding administrative issues.

In addition, two of the partners organise annual training seminars (in the language of the host country) with the aim of raising the awareness of the members of the partner institutions (teachers and students) to the problem of computational thinking and to encourage them to use the competencies framework and to contribute to the development of pedagogical scenarios.

Dissemination seminars are also planned from 2020 in the partner countries in order to disseminate the productions (competencies framework, multilingual pedagogical scenarios and teaching resources), to build a network promoting the exchange of practices and resources.

Objectives

The deliverables targeted by the project are:

- a competencies framework tested and validated by different stakeholders of the project (researchers, teachers, institutional partners),
- a set of pedagogical scenarios with learning activities tested in classrooms (to feed a future open-access portal),
- useful teaching resources for their implementation.

(Yet another) Competencies framework

The question of defining the concept of computational thinking is complex *per se*. We place ourselves here in line with the work of Wing (2006), and define these terms, in general terms, as well as the set of competencies specific to the representation and manipulation of information (including problem solving).

It should be noted that some of these competencies are also found in other disciplines (the representation and manipulation of information when applied to numerical values or the notion of a problem are addressed in mathematics, for example). It is therefore difficult to grasp the contours of computational thinking. A first step in this direction is to break it down into clearly identified and interrelated competencies (in other words, a competencies framework).

In what follows, we are going to make a critical analysis of a certain number of reference frameworks (mainly proposed in Anglo-Saxon countries) in relation to our previously mentioned objectives (precise definition, intelligible to non-specialists and independent of any particular electronic system or programming language).

State of the art: Anglo-Saxon and European reference frameworks

Without forgetting the visionary work of Seymour Papert (1971), who nearly half a century ago already laid the foundations of computational thinking, we can consider that the collective awareness of the importance of defining a framework for training in what is similar to computational thinking took place some fifteen years ago in various Anglo-Saxon countries. This awareness is mainly due to lobby groups: Association of Computer Science Teachers (Computer Science Teachers Association) created in 2004 in the United States, and the *ComputerScience4Fun* and *computing@school* associations created respectively in 2005 and 2008 in England (Crick & Sentance, 2011).

The case of the United States

The CSTA association proposed in 2011 a common framework (competency framework) to learn computer science from elementary school on (Seehorn et al., 2011). This framework, entitled CSTA K12 standards (https://www.csteachers.org/page/about-csta-s-k-12-nbsp-standards), was revised in 2017, and describes a set of competencies that are worked on from the age of five, and grouped into five thematic axes:

1. Information systems
2. Networks and Internet
3. Data and Analysis
4. Algorithms and programming

5. Impacts of information technology

As this thematic breakdown suggests, the CSTA curriculum has a relatively strong technical orientation, in the sense that a significant part of the competencies framework (the first two axes) is related to the use of particular software or hardware.

However, the CSTA approach constitutes a major contribution, as it includes a scientifically-based competencies framework (validated by peers, members of one of the oldest societies in computer science, the Association for Computational Machinery), validated by field staff (CSTA members), accompanied by a fine progression of competencies by age group, and integrated into a broader scheme of further education (secondary and then higher education). The CSTA is also currently working on the definition of a reference system of competencies for computer science educators, in collaboration with the International Society for Technology in Education (ISTE), https://www.csteachers.org/page/standards-for-cs-teachers.

The case of England

Since the above-mentioned associations (*ComputerScience4Fun* and *computing@school*), as well as the Royal Society, (https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf) have been advocating for the recognition of computing as a Computing Science and not as a mere tool, the English government has decided to introduce a new discipline entitled "Computing" in its school curricula in 2013 (https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study).

Training in this discipline is divided into 4 Key Stages, each of which corresponds to a set of competencies to be acquired. For example, the first Key Stage includes the competency "create and correct simple programs", and the final Key Stage includes the competency "understand how changes in technology impact security, including new ways to protect privacy and identity online, and how to raise a range of concerns".

As this last example illustrates, the national program includes relatively broad competencies that may seem difficult to acquire. To assist teachers in this task, the *computing@school* association has developed a detailed progression framework called Computing Progression Pathways based on six areas:

1. Algorithms
2. Programming
3. Data and representations
4. Equipment and treatments
5. Communication and networks
6. Information Technology

This competencies framework is also a major contribution, if only because of its level of detail. Indeed, each of the six domains is broken down into eight training steps, each of which groups together three to five competencies illustrated by activities and annotated using key concepts (abstraction, generalization, algorithmic thinking, etc.). It benefits from the support of various players in the computer world (academics, teachers, professional world), and has played a pioneering role in the introduction of computer science training (not only computational thinking) in national curricula. Its main limitation lies in its richness, as it brings together a large number of competencies from various fields: computer science, information technology and digital literacy, which may hinder its readability.

The case of Australia

Another interesting proposal lies in the Australian curriculum, which from 2015 introduces two distinct subjects: information and communication technology literacy and digital technology literacy. See https://www.australiancurriculum.edu.au/f-10-curriculum/ as well as https://www.australiancurriculum.edu.au/f-10-curriculum/general-capabilities/information-and-communication-technology-ict-capability/ for more details.

While the former are attached to general competencies, the latter are part of the technology discipline. Both are taught as early as elementary school. As mentioned in the introduction to the ICT domain, the competencies covered by this domain refer to "skills that enable the learner to make optimal use of the digital tools available". Digital technology literacy, on the other hand, refer to competences specific to computers (including computational thinking). These are defined around ten key concepts:

1. Abstraction
2. Digital systems
3. Data representation
4. data collection
5. Interpretation of data
6. Specification
7. Algorithms
8. Implementation
9. Impact
10. Interactions

This curriculum therefore enables every Australian student to develop computational thinking from an early age, but by labelling it as a part of technology. This choice of categorisation of computational thinking may hinder its accessibility by participating in a technical vision of computing and may even have consequences on the attractiveness of this field for a female audience (Collet, 2004).

The European framework

Europe is not to be forgotten here, since a group of experts has drawn up a reference framework of digital skills for European citizens as early as 2013. This framework, entitled "Digital Competencies", revised in 2017, is based on five areas of competence (Carretero, Vuorikari, & Punie, 2017):

1. Data and information literacy
2. Communication and Collaboration
3. Creation of digital content
4. Security
5. Problem solving

Each of these five domains is broken down into a set of three to six sub-domains (21 sub-domains in total). For each of these, eight levels of proficiency are defined, ranging from basic to highly specialized skills. Each level of competence is associated with knowledge, know-how and interpersonal skills, and is illustrated by one or more examples of situations in which the competence is applied. This repository is also an important contribution to the development of computational thinking in the learner for several reasons. First of all, through the presence of numerous concrete examples, it explains the link between digital skills and common situations (thus contributing to an awareness of the underlying societal issues). Secondly, through its normative role, it has a *de facto* impact on educational policies in Europe, thus opening the door to a redefinition of curricula that is as close as possible to current issues. Its main limitation lies in the fact that the skills related to computational thinking are not very precise and are drowned in the quantity of skills described.

Development of a new framework

Motivations

From the above review of existing solutions, we can retain some positive points and some limitations. For the positive aspects, these frameworks show an awareness of the importance of developing skills related to the digital world in future citizens, and this from elementary school on. In addition, they propose a relatively detailed progress in training courses to help them understand the digital world, thus contributing to a certain emancipation of the future citizen.

The main limitations of these references seem to us to be the following: (1) they do not explicitly distinguish (with the exception of the Australian framework) between computational thinking, which can be seen as transversal skills that can be mobilized in various disciplines, and (2) they are based on relatively rich and complex progressions, which are not necessarily readable by non-specialists.

Our framework

In order to respond to these limitations, we propose to draw inspiration from the above frameworks to define a new one, which is (i) centred on computational thinking (as defined above), (ii) targeted at basic education (kindergarten and primary school) while laying foundations compatible with secondary or even higher education, and (iii) accessible to non-specialists.

This competencies framework (which can be consulted on our project webpage https://piaf.loria.fr/contributions/) has been built collaboratively by successive iterations, through discussions during project meetings on the one hand, and through exchanges with field staff during training seminars on the other hand.

This framework has its source in the definition of Win g (2006) (which has had a certain echo in the IT community) and in the IT curricula of the Anglo-Saxon countries (see frameworks discussed above).

The aim here is to define a set of competencies specific to computational thinking, using an accessible vocabulary (i.e. using relatively common terms).

We have distinguished the following six competencies, each of which is broken down into a set of three to seven sub-competencies):
C1 - Defining abstractions / generalizing
C2 - Compose/decompose a sequence of actions
C3 - Controlling a sequence of actions
C4 - Evaluating objects or sequences of actions
C5 - Translate information into different representations
C6 - Construct a sequence of actions iteratively

The main aim of competency C1 is to develop the ability to name (groups of) objects or actions for later reference. An object can be a value (e.g. a person's first name) or a variable (e.g. outdoor temperature). Competency C2 aims to develop the ability to manipulate actions to solve tasks. These actions may involve objects (or groups of objects) as defined in C1. Competency C3 aims to develop the ability to control when a sequence of actions can be performed. Conditional expressions and repetitions are introduced. Competency C4 is designed to develop the ability to think about the construction of sequences of actions (i.e. to judge the suitability of a sequence in relation to given constraints). Competency C5 should develop the ability to adapt to various equivalent representations (e.g. understand that 2 × 2 and 8/2 refer to the same object). Lastly, Competency C6 aims to develop a work methodology based on an iterative approach (i.e. by successive refinements) specific to computational thinking.

The sub-competencies are divided into two levels: fundamental sub-competencies (whose acquisition can begin between 5 and 8 years) and advanced sub-competencies (whose acquisition can begin between 9 and 12 years).

In addition, the framework is accompanied by two documents: a glossary to explain and illustrate the "technical" terms, and a set of concrete examples to illustrate the various sub-competencies.

Methodology for creating pedagogical scenarios

The dissemination of this competencies framework cannot take place without (i) the provision of (at least some exemplar) pedagogical activities, and (ii) the integration of these activities within pedagogical scenarios allowing for transfer in classrooms. We present here the methodology used to create these pedagogical scenarios.

A participatory approach

What seems to us to be an important condition for the successful dissemination of the competencies framework lies in the teachers' ability to propose activities related to the targeted competencies. This creative capacity would not only be the mark of an appropriation of the competencies framework, but also a motivational tool for teacher involvement. The methodology used to develop this creative capacity is based in particular on training seminars, during which participants are led to design pedagogical scenarios (set of activities) aimed at developing computational thinking among young people aged 5 to 12. They are also asked to test these scenarios in the field. Take, for example, the training conducted jointly by the Universities of Liège and Lorraine. This training was hybrid and started with a first two-day face-to-face seminar. During the first phase (day 1), two types of facilitators, pedagogues and computer scientists, were involved in the following activities:

1. Introduction to computational thinking, and presentation of the PIAF competencies framework;
2. Illustration of these concepts through examples of activities;
3. Realization of these activities in groups (rotating workshops);
4. Linking the activities carried out to the competencies of the competencies framework.

The second phase (day 2) focused on the pedagogical scenarios:

1. Presentation of the structure of a pedagogical scenario;
2. Familiarization with this structure by asking participants to fill it in, based on an activity they had experienced during phase 1;
3. Collaborative design of a pedagogical scenario that will be tested in classrooms.

The designed scenario must propose at least one activity working on one or more of the competencies of the framework. In addition, it must respect a triple concordance between competencies, learning activities and evaluation. During this second phase, the learners were assisted by experts (facilitators from phase 1) to help them take a step back, or to deepen certain points, if necessary. The rich exchanges during this phase allowed feedback on the reference system. The proposed methodology reuses the levers Gregory et al. (2013), in particular:

- Encouraging the generation of ideas by asking questions (or problems) with several answers (solutions);
- Ask students to propose several ideas to the questions or problems and remind them to make each solution as varied as possible;
- For each potential solution suggested by a student, ask them to also think about its implications and implementations;
- Include external mediators in the group work.

At the end of the seminar, each pair was asked to continue the collaborative design of their pedagogical scenario. For this purpose, distance coaching was proposed, in particular to comment on the productions and to encourage participants to produce detailed and coherent scenarios. Once validated, the scenarios were tested in classrooms. Some pairs moved towards activities similar to those provided by the experts, others took more "risks". A second face-to-face seminar was then organized to get feedback on the scenario experimentation in the field.

At this stage of the project, 4 training seminars have been organized and about ten scenarios have been tested in classrooms.

Outlook

One long-term objective is to create a community of practice, by feeding an open-access online portal of pedagogical scenarios and learning resources. Dissemination seminars will make it possible to share these productions, and to enlist other collaborators in this participatory design process.

This three-year project also aims to evaluate the impact of these resources on classroom practices and on the audiences trained. It will also validate (or not) the hypothesis according to which training such as the one proposed here allows the development of the pedagogical uses of digital technology.

Conclusion

In this article, we presented the PIAF project which aims at developing computational thinking in elementary school, with a double challenge: to allow digital autonomy (securing digital users), and to develop fundamental transversal skills useful to every citizen. This development requires (i) the definition of a competencies framework that is scientifically based, but also understood and accepted by teachers (and to some extent by learners), and (ii) the construction and exchange of inspiring practices (pedagogical scenarios).

References

Baron, G.-L. & Boulc'h, L. (2011). Les technologies de l'information et de la communication à l'école primaire. État de la question. *Revue de l'association Enseignement Public et Informatique*. Retrieved 10.04.2020 from http://www.epi.asso.fr/revue/articles/a1202b.htm

Carretero, S., Vuorikari R. & Punie, Y. (2017). *DigComp 2.1 - The Digital Competence Framework for Citizens - With eight proficiency levels and examples of use*. Retrieved 10.04.2020 from http://publications.jrc.ec.europa.eu/repository/bitstream/JRC106281/web-digcomp2.1pdf_(online).pdf

Collet, I. (2004). La disparition des filles dans les études d'informatique : Les conséquences d'un changement de représentation. *Carrefours de l'éducation*, 17, pp. 43–56. DOI: https://doi.org/10.3917/cdle.017.0042

Crick, T. & Sentance, S. (2011). Computing at school: stimulating computing education in the UK. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*. ACM, New York, NY, USA, 122–123.

Delarbre, P. (2017). *Initiation à la programmation à l'école primaire : les activités «débranchées»*. Mémoire de CAFIPEMF, académie de Lyon.

Direction Académique au Numérique Educatif (DANE) de Besançon (2016). Enquête PROFETIC sur les usages pédagogiques numériques des enseignants du second degré. Retrieved 10.04.2020 from https://dane.ac-besancon.fr/wp-content/uploads/sites/56/ 2016/08/Academie_Colleges_Tte_discipline_Enquete-Profetic_2016_sc.pdf

Gregory, E., Hardiman, M., Yarmolinskaya, J., Rinne, L., & Limb, C. (2013). Building creative thinking in the classroom: From research to practice. *International Journal of Educational Research*, 62, 43–50.

Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherche (2015). Programmes d'enseignement du cycle des apprentissages fondamentaux (cycle 2), du cycle de consolidation (cycle 3) et du cycle des approfondissements (cycle 4), Bulletin officiel de l'éducation nationale, no 10 du 19 novembre 2015.

Ministère de l'éducation nationale (2019). Décret no 2019-919 du 30 août 2019 relatif au développement des compétences numériques dans l'enseignement scolaire, dans l'enseignement supérieur et par la formation continue, et au cadre de référence des compétences numériques. *Journal Officiel de la République Française*, 1er septembre 2019, no 0203, texte 25.

Papert, S. (1971). Teaching Children Thinking. *Artificial Intelligence Memos*, 247, Massachussets Institute of Technology, A.I. Laboratory. Retrieved 10.04.2020 from https://dspace. mit.edu/handle/1721.1/5835

Seehorn, D., Carey, S., Fuschetto, B., Lee,I., Moix, D., O'Grady-Cunniff, D., Boucher Owens, B., Stephenson, C & Verno, A. (2011). *CSTA K–12 Computer Science Standards : Revised 2011*. Technical Report. ACM.

Wing, J., M. (2006). Computational thinking. *Communications of the ACM - Self managed systems*, *49*(3), 33–35.