



HAL
open science

Revising a programming task in geometry through the lens of design-based implementation research

Raimundo Elicer, Andreas Lindenskov Tamborg, Uffe Thomas Jankvist

► To cite this version:

Raimundo Elicer, Andreas Lindenskov Tamborg, Uffe Thomas Jankvist. Revising a programming task in geometry through the lens of design-based implementation research. Twelfth Congress of the European Society for Research in Mathematics Education (CERME12), Feb 2022, Bozen-Bolzano, Italy. hal-03765101

HAL Id: hal-03765101

<https://hal.science/hal-03765101>

Submitted on 30 Aug 2022

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.

Revising a programming task in geometry through the lens of design-based implementation research

Raimundo Elicer¹, Andreas Lindenskov Tamborg² and Uffe Thomas Jankvist³

¹Aarhus University, Danish School of Education, Denmark; raimundo@edu.au.dk,

²University of Copenhagen, Centre for Digital Education, Denmark; andreas_tamborg@ind.ku.dk,

³Aarhus University, Danish School of Education, Denmark; utj@edu.au.dk

The paper addresses the role of educational task design in implementation research. Its point of departure is the first revision of a task developed for 4th grade in collaboration with a Danish school teacher. The authors informed the task from a hypothetical learning trajectory that requires students to draw on their knowledge both of mathematics and programming and computational thinking. From a task-design standpoint, the teacher's experience proves necessary to adapt the task to local reality. The collaborative process illustrates broader implementation issues, such as problems and roles of different stakeholders, innovation adaptation and capacity building. The lessons are consistent with the emergence of design-based implementation research as a more comprehensive model.

Keywords: Computational thinking, design-based implementation research, hypothetical learning trajectories, programming, task design.

Introduction

The mathematics education community has recently gained a renewed interest in programming and computational thinking (PCT) as related subject matter areas. Although there is consensus on the potential synergies between mathematics and PCT, exploiting them has proven to be a difficult task (Misfeldt, Szabo & Helenius, 2019). Research in mathematics education has taken several approaches to address this difficulty. For example, by considering how to train and prepare mathematics teachers to teach PCT (e.g., Kilhamn & Bråting, 2019), coming up with ways to conceptualise its relation with mathematics (e.g., Benton et al., 2017; Weintrop et al., 2016), and trying to understand better students' learning processes and dispositions in computational thinking-driven mathematics classrooms (Pérez, 2018).

More locally, Denmark is planning to implement technology comprehension as a new subject in K-9 education comprising four competency areas: digital empowerment, digital design and design processes, computational thinking, and technological knowledge and skills—including programming (Smith et al., 2020). The Danish Ministry of Education is conducting a pilot project at 46 schools is seeking to gain experience with implementing it (1) as a subject in its own right and (2) as an element integrated into existing subjects, here among mathematics. In both cases, the ministry has published a tentative curriculum with added technology comprehension learning goals and prototypes of teaching resources.

This paper takes an implementation perspective on the challenges of intertwining PCT with mathematics teaching, anchored in the Danish school context. The broader aim of the research project is to support mathematics teachers in connecting PCT and mathematics in their everyday practice through the development of carefully designed tasks. By engaging in an educational design process,

we draw on this approach to iteratively design, implement, and refine activities developed to fit the mathematics teachers' needs and support students' learning.

We anchor the content of this paper at a school that is part of the aforementioned pilot project, where we collaborate with a mathematics supervisor to refine a task developed to integrate PCT and mathematics in grade 4. In this context, matters of implementing a new subject and of task design interweave.

Design-based implementation research

Implementation research in mathematics education seeks to close the gap between research and practice (Jankvist et al., 2021), including the characterisation of a particular innovation and the factors conditioning its implementation (Century & Cassata, 2016). We aim to illuminate the characteristics of the new subject as embedded in the mathematics curriculum as a factor. In particular, we narrow down the innovation to a particular task.

One emergent model acknowledging the role of educational design in the context of implementation is design-based implementation research (DBIR). As noted by Fishman et al. (2013), DBIR is different from conventional design research, which only focuses on student learning, in that it points out “how the deployment of new tools (e.g., curricula, technologies) can bring to light new needs for coordination across different system levels and for capacity building” (p. 144). DBIR is then characterised by four principles “(1) a focus on persistent problems of practice from multiple stakeholders' perspectives; (2) a commitment to iterative, collaborative design; (3) concern with developing theory and knowledge related to both classroom learning and implementation through systematic inquiry; and (4) a concern with developing capacity for sustaining change in systems.” (Fishman et al., 2013, pp. 136–137). Based on this framing, we ask the following research question: *What can we learn about the implementation of PCT in mathematics education by designing a single task with a teacher from a DBIR perspective?* The second and third principles pertain to standard design research (Bakker & van Eerde, 2015). In this paper, we focus on the first and fourth principles to organise our discussion, which is anchored in the early stages of designing a programming and geometrical task.

An essential conceptual tool for developing the design and learning from iterations is that of a Hypothetical Learning Trajectory (HLT). In the context of one or two lessons, an HLT is “the consideration of the learning goal, the learning activities, and the thinking and learning in which students might engage” (Simon, 1995, p. 133). Artigue (2021) highlights HLTs as a critical theoretical resource from design-based research to connect research and practice. An HLT seeks to make explicit what will happen when the design is brought into the hands of the practitioners concerning the real-world problem it aims to address. It describes the process that ought to take place during the intervention with the design. An HLT is informed by theory and should be sufficiently exhaustive so that its assumptions can be challenged by a real-world enactment of the design (Cobb & Gravemeijer, 2008). The HLT enables one to expose unforeseen interplays between intention, design, and reality, thereby challenging and qualifying our theoretical assumptions. After an intervention, one is thus able to consider whether it led to the desired outcome or not and point expressly to where our understanding of the design was confirmed or challenged—and what needs to be modified (Doorman et al., 2013). In the following, we describe the HLT informing our task design.

Designing a programming and geometrical task

Before the interaction with the expert mathematics teacher, the authors developed an a priori HLT addressing what we anticipated to be the main challenges in implementing PCT in mathematics. We envisioned that to ensure that students engaged in learning processes involving both mathematics and PCT, it was necessary to develop a task that required them to activate knowledge from the domains of both geometry and programming. Moreover, we believed that creating such a task would have a positive side effect for mathematics teachers in that the relevance of integrating PCT in mathematics would be clear. The learning goal is to deduce and test a general expression for the internal angle of any regular polygon. To solve the programming task, students should do the mathematical action of conjecturing a general expression for the internal angle of an n -gon. In turn, to test out such an expression, they must find a way of programming it in *Scratch*. A sample solution is depicted in scratch.mit.edu/projects/541978601.

The learning activities, depicted in Table 1, scaffold students' work by drawing on the *use-modify-create* principle from programming education (Lee et al., 2011), which states that it is easier for students to start with a pre-made script and adapt it to solve variations of a problem before elaborating new code. We also envisioned a progression in the task based on increasing difficulty by starting from a square, then a triangle, hexagon, pentagon and finally generalised polygons.

Table 1: Original hypothetical learning trajectory

Learning activities		Hypothetical learning process
Use	a <i>Scratch</i> code that draws one segment. a <i>Scratch</i> code that turns the pen 90° . Explore what happens when run repeatedly.	Become familiarised with basic <i>pen up</i> , <i>down</i> , <i>move</i> blocks. Become familiarised with <i>turn(degrees)</i> block. Realise that four repetitions suffice to draw a square.
Modify	the code to draw a square with one run using a repeat loop. the code so that it draws an equilateral triangle. the code so that it draws a regular hexagon. the code so that it draws a regular pentagon.	Intuitive inclusion of a <i>repeat(4)</i> -block. Modify to <i>repeat(3)</i> and <i>turn(60°)</i> , revising to <i>turn(120°)</i> distinguishing between internal and external angles. Modify to <i>repeat(6)</i> and <i>turn(60°)</i> , after previous experience or trial and error. A trial-and-error strategy will be slow or unsuccessful, prompting to find the 72° external angle with pen and paper.
Create	a code that draws any regular n -sided polygon. Teacher introduces <i>variable</i> block.	After registering previous cases, students find collectively a pattern that can be generalised for internal/external angles.

The design decisions and hypothetical learning processes were rooted in the literature on embedding PCT in mathematics classrooms. One of the mathematical courses from technology comprehension available in teknorsøget.dk/forlob involves programming *BeeBots*¹ to draw polygons in first grade. The robots, however, can only turn 90° , limiting the tasks to squares and rectangles. Drawing

¹ BeeBot is a robot designed for use by young children, which can be used for teaching sequencing, estimation, problem-solving and more.

polygons is also part of the *ScratchMath* project’s “Beetle geometry” module (see, e.g., Benton et al., 2017). Their design proposes exploring four increasingly difficult polygons, figuring the turning angle: square, rectangle, equilateral triangle, and regular hexagon.

Furthermore, Herheim and Johnsen-Haines (2020) explored two seventh-grade students’ productive struggles while drawing a pentagon. From here, some of the expected obstacles are mistaking internal by external angles and a trial-and-error strategy that will not suffice to reach a solution, thus calling for reasoning and calculating the exact 72° angle. Accordingly, students should overcome sequential challenges. First, students would use a loop to draw a square in one run. Secondly, they will sort out the internal angle for a triangle and hexagon and adapt their code. Next, they will require a pen-and-paper solution for the pentagon, as opposed to trial-and-error. Finally, they would generalise the latter case, introducing a variable for the number of sides.

Upon designing the task described above, the two first authors interviewed an expert mathematics teacher who acted as a supervisor at the school on which the task is to be implemented. The purpose of the interview was two-fold: (1) to test and reach a mutual understanding of the core implementation challenges to address and (2) to refine the task according to these challenges. The interview was semi-structured and 60 minutes long. The teacher agreed to have the meeting recorded for research purposes. Below, we describe the insights regarding our initial HLT that this interview brought.

Revisions of the task based on teacher interview

The teacher agreed on the learning goal, but the sequence of activities and hypothetical learning processes are revised in Table 2. Based on her input, three changes are worth mentioning: starting with a blank script, as opposed to using a working code, further use of pen-and-paper, and sequential order of polygons according to the number of sides.

Table 2: Revised HLT based on teacher’s input

Learning activities	Hypothetical learning processes
What are polygons? Write your findings in <i>Book Creator</i> .	Students collectively activate their previous knowledge on polygons and their elements (sides, angles, vertices).
Create a <i>Scratch</i> code that draws an equilateral triangle. Create a <i>Scratch</i> code that draws a square. Create a <i>Scratch</i> code that draws a regular pentagon. Create a <i>Scratch</i> code that draws a hexagon.	Students engage in a trial-and-error strategy, registering in <i>Book Creator</i> the parameters used for <i>repeat</i> and <i>turn</i> blocks in increasing order. These are tools for the next activity.
Create a code that draws a figure of your choice: a flag, a labyrinth, a logo, or your own artwork.	This motivation is set up in the introduction but executed in the end. It gives students purpose to explore polygons’ properties and how to depict them in <i>Scratch</i> .

The teacher challenged the pedagogy of *use-modify-create* by proposing that students should start creating their own code. The idea of starting with a given script is that it avoids frustration and makes the first step easy.

Teacher: It’s easy to do it. But ... instead of giving the code from the start, the first thing is to let the children make it themselves. Maybe they can do it; maybe they can’t. But they need to understand what are we doing now. Instead of just having the code,

“how can we do this?” “What is the problem?” “What do we need to know?”
“Which kind of code do we need to do to code a triangle?”

Overall, her take is that more relevant than a smooth start in terms of difficulty, the students ought to perceive the posed problem as their own. Moreover, enabling struggle (Pérez, 2018; Herheim & Johnsen-Haines, 2020) and debugging (Weintrop et al., 2016) is consistent with the literature on computational thinking in mathematics classrooms.

Since an essential part of our HLT was requiring students to activate both mathematical and programming knowledge, all mathematical operations in our task were conducted in *Scratch*. We envisioned that the students would occasionally enter a flawed code, which the *Scratch* drawing would make clear (a flawed figure), and that the students iteratively would correct the code to draw the desired one. While the teacher acknowledged and agreed with the importance of this iterative approach, she anticipated that the students would not by themselves engage systematically in a new iteration if they were not encouraged to document their working process. Therefore, she suggested supplementing the task with pen and paper for students to log their attempts and use these logs actively in the next iteration.

Teacher: I call it [to] scribble. They have to take notes of what they did. After all, the idea is not that they randomly should enter numbers from 1–200. What they enter should be based on their thinking. They have to learn to be systematic, you know.

In the original design, pen-and-paper was reduced to a minimum and based on the idea of computational thinking as a thought process that becomes concretised by programming in a programming environment. In technology comprehension, programming is considered a technological skill detached from computational thinking (Smith et al., 2020). However, the teacher had a different approach:

Teacher: I think the children start, and then we need to engage [them] in the thinking along the way. They are not sitting and planning.

The task ultimately aims at educating students to draw an n-sided polygon, for which they require to find a general expression for its internal angle. Though the literature supports the scaffolding in our initial design (e.g., Benton et al., 2017), the teacher suggests arranging the sequence in increasing order. She draws on the Swedish researcher Olof Magne’s juxtaposition of arithmetic operations in teaching:

Teacher: And I have worked with that. Instead of just starting with addition, then I took it all [i.e. all arithmetical operations], and the children didn’t have the same problems as they usually have. They learnt to make divisions in the first grade because we learnt it all. That’s part of my mindset.

Overall, these proposed changes to the task respond to letting students *own* the problem and *explore* possible solutions systematically instead of solving a fixed sequence of exercises and offering scaffolded solutions.

Discussion

As described previously, DBIR consists of four main principles, namely persistent problems from multiple stakeholders, iterative and collaborative design, theory development, and building capacity for sustainable change. Although we believe the study reported in this paper has accommodated all

of these principles, the teacher interview led to refinements of the task that we anchor in the first and fourth.

Discussing the task gave perspective towards different stakeholders and their presumed roles. In particular, the new version of the task will be more open, thus engaging students to solve a problem perceived as their own. Moreover, we envision tackling the issue of teacher training in programming by seeing students coding themselves:

Teacher: They (...) think that [the] responsibility for teaching *Scratch* in on their shoulders; they don't take it. But then they see that the children can manage *Scratch* and they have (...) seen what the children can do, and have seen the children's eyes, and they have felt the good spirit about working with math and *Scratch*, then they are convinced.

Tasks supporting PCT and mathematics integration should accommodate both aims of ensuring students' mathematical learning processes *and* anticipated or perceived challenges from teachers' perspectives. The design process reported in this study corroborates that these ends are compatible by meaningfully relating the two subject matters and acknowledging that teachers do not feel sufficiently proficient in PCT to teach it. On the same line, it is worth mentioning that the teacher did not question the HLT's learning goal, as it addresses one of the main issues of technology comprehension tasks, namely their connection to the mathematics subject matter:

Teacher: The teachers know math, and they know the didactics around that. But they don't know the didactics around technology comprehension. (...) Suddenly it was something they had to work with (...) and then, they saw the math and they thought the math wasn't... eh... very good.

The case of this first design iteration illustrates how one can anticipate adaptation and reduce the distance between what is supposed to be and actually being implemented. For example, the initial design included a working code assuming that students had not been introduced to the pen environment on *Scratch*. However, the teacher challenged this claim.

Although previous research has found *use-modify-create* to be an appropriate approach to scaffold students who are not accustomed to coding, the teacher had experience and knowledge of the importance of showing students that they are able to develop code on their own from very early on. Besides informing refinement of the task, this insight indicates that while theory on the didactics of PCT can inspire initial task design, it does not necessarily align with what is likely to work in a specific context. That is not to say that one particular teacher's opinion must be generalised at face value. In turn, these conflicts can be tried out as hypotheses for theory development, in its humble sense (diSessa & Cobb, 2004). Furthermore, from a DBIR perspective, the quest for designing resources that build capacity over time must account for the diversity of pedagogical choices.

Conclusion

In this paper, we have investigated what we can learn about the implementation of PCT in mathematics education by designing a single task with a teacher from a DBIR perspective. Revisions to the task were made by interviewing a teacher involved in the future process of implementing such a sequence. The new version promotes further ownership on behalf of the teacher by making her take part in the design process and revise implicit and explicit

assumptions. Further, working on a particular HLT goes beyond local task design. Fidelity is a recurrent theme in implementation research (Century & Cassata, 2016), likely because the design of innovations may conflict with teachers' established and preferred practices. Involving the teacher in the initial design phase allowed us to anticipate such conflicts and to alter the innovation design in accordance with her reality. Our future collaborative endeavour with the teacher will teach us more about whether it leads to high or low degrees and innovation enactment adaptation.

The openness to practitioners' needs can make theory development more grounded (Bakker & van Eerde, 2015), but it leaves at least three options to move forward. In a humble sense, the educational design enables developing a local theory of how a task could successfully implement innovations that integrate mathematics and PCT at a specific school. A second alternative is to inform the conjecturing of hypotheses to try out at a larger scale. For example, one may validate or dismiss the pedagogical choice for *use-modify-create*. DBIR, however, offers a broader perspective. For capacity building's sake, a variety of possible approaches can be provided to practitioners, who may pick and adapt tasks according to their priorities. In that sense, theory development ought to be at a higher level than a particular HLT. Instead, resources that sustain change for embedding PCT in mathematics may refer to general domain-crossing themes or scenarios, such as drawing geometrical figures with programming.

Acknowledgement

This work is funded by the *Novo Nordisk Foundation*, grant number NNF19OC0058651.

References

- Artigue, M. (2021). Implementation studies in mathematics education: What theoretical resources? *Implementation and Replication Studies in Mathematics Education*, 1(1), 21–52. <https://doi.org/10.1163/26670127-01010002>
- Bakker, A., & van Eerde, D. (2015). An introduction to design-based research with an example from statistics education. In A. Bikner-Ahsbabs, C. Knipping, & N. Presmeg (Eds.), *Approaches to qualitative research in mathematics education: examples of methodology and methods* (pp. 429–466). Springer. https://doi.org/10.1007/978-94-017-9181-6_16
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, 3(2), 115–138. <https://doi.org/10.1007/s40751-017-0028-x>
- Century, J., & Cassata, A. (2016). Implementation research: Finding common ground on what, how, why, where, and who. *Review of Research in Education*, 40(1), 169–215. <https://doi.org/10.3102/0091732X16665332>
- Cobb, P., & Gravemeijer, K. (2008). Experimenting to support and understand learning processes. In A. E. Kelly, R. A. Lesh, & J. Y. Baek (Eds.), *Handbook of design research methods in education: Innovations in science, technology, engineering, and mathematics learning and teaching* (pp. 68–95). Routledge.

- diSessa, A., & Cobb, P. (2004). Ontological innovation and the role of theory in design experiments. *The Journal of the Learning Sciences*, 13(1), 77–103. https://doi.org/10.1207/s15327809jls1301_4
- Doorman, M., Drijvers, P., Gravemeijer, K., Boon, P., & Reed, H. (2013). Design research in mathematics education: The case of an ICT-rich learning arrangement for the concept of function. In T. Plomp & N. Nieveen (Eds.), *Educational Design Research—Part B: Illustrative Cases* (pp. 425–446). SLO: Netherlands institute for curriculum development.
- Fishman, B. J., Penuel, W. R., Allen, A.-R., & Sabelli, N. (2013). Design-based implementation research: An emerging model for transforming the relationship of research and practice. *National Society for the Study of Education*, 112(2), 136–156.
- Herheim, R., & Johnsen-Haines, M. (2020). A culture perspective on students' programming in mathematics. *Journal of Mathematics and Culture*, 14(2), 91–110.
- Jankvist, U. T., Aguilar, M. S., Misfeldt, M., & Koichu, B. (2021). Launching Implementation and Replication Studies in Mathematics Education (IRME). *Implementation and Replication Studies in Mathematics Education*, 1(1), 1–19. <https://doi.org/10.1163/26670127-01010001>
- Kilhamn, C. & Bråting, K. (2019). Algebraic thinking in the shadow of programming. In U. T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (Eds.), *Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education* (pp. 566–573). Freudenthal Group & Freudenthal Institute; Utrecht University; ERME.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37. <https://doi.org/10.1145/1929887.1929902>
- Misfeldt, M., Szabo, A., & Helenius, O. (2019). Surveying teachers' conception of programming as a mathematics topic following the implementation of a new mathematics curriculum. In U. T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (Eds.), *Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education* (pp. 2713–2720). Freudenthal Group & Freudenthal Institute; Utrecht University; ERME.
- Pérez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education*, 49(4), 424–461. <https://doi.org/10.5951/jresmetheduc.49.4.0424>
- Smith, R. C., Bossen, C., Dindler, C., & Sejer Iversen, O. (2020). When participatory design becomes policy: Technology comprehension in Danish education. *Proceedings of the 16th Participatory Design Conference 2020 - Participation(s) Otherwise*, 1, 148–158. <https://doi.org/10.1145/3385010.3385011>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>