A Template for Teaching Computational Modelling in High School

Line Have Musaeus*
lh@cs.au.dk
Center for Computational Thinking & Design, Aarhus University
Aarhus, Denmark

Michael E. Caspersen
Center for Computational Thinking & Design, Aarhus University
Aarhus, Denmark

Peter Musaeus
Centre for Educational Development, Aarhus University
Aarhus, Denmark

ABSTRACT

Computing education is becoming increasingly important in high schools. Computational modelling is important in computing and many sciences, but there is a lack of research on how teachers should teach computational modelling in high schools. This study was a design-based research study with 86 teachers teaching 12 different subjects at 44 Danish high schools. The study aimed to develop a template to help design and classify didactical questions on computational modelling. Teachers participated in one of two courses on computational modelling. The intervention group (Prog+) included an introduction to agent-based modelling and programming in NetLogo. The comparison group (Prog-) included a general introduction to agent-based modelling. A template consisting of 16 modelling parameters was developed with teachers. Results showed that the template was helpful for teachers to design didactical questions and for the research team to classify the taxonomical levels of these questions. A total of 51 teaching activities were developed by teachers and didactical questions were derived. The strength of this design based research study was that it included a control group and inspired teachers to design and evaluate didactical questions in computational modelling in a wide range of high school subjects. Future studies are needed to evaluate the validity of the template.

CCS CONCEPTS

• **Social and professional topics → Computational thinking**; **K-12 education**.

KEYWORDS

Computational modelling, Computational thinking, High school education, K-12 education, Professional development, Design-based research

ACM Reference Format:
Line Have Musaeus, Michael E. Caspersen, and Peter Musaeus. 2023. A Template for Teaching Computational Modelling in High School. In *The 18th WiPSCE Conference on Primary and Secondary Computing Education Research (WiPSCE '23), September 27–29, 2023, Cambridge, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3605468.3609754

*Corresponding author

WiPSCE '23, September 27–29, 2023, Cambridge, United Kingdom

ACM ISBN 979-8-4007-0851-0/23/09…$15.00
https://doi.org/10.1145/3605468.3609754

1 INTRODUCTION

Computational modelling is an important scientific activity that allows scientists to make educated guesses such as forecasting the weather, predicting elections, or tracking contagious diseases. Not only scientists, but also science educators use computational models to build abstractions and visualize scientific phenomena [34, 44, 54]. Teachers need to be introduced to computational modelling as an educational and scientific activity. Further training is necessary for teachers to acquire the skills needed to design teaching activities that stimulate students' learning of computational modelling.

Computational modelling is associated with computational thinking [22, 32, 64]. Computational thinking can be defined as a person's thought processes in formulating and preparing a problem for a computational solution [14, 15, 21]. Research is emerging on K-12 students' learning of computational thinking in different subjects is emerging [14, 15, 21], including computer science [6, 50], physics [25], biology [28, 55], and the liberal arts [16, 29]. Arguably, educational researchers have focused more on computational thinking than computational modelling [63].

There are two challenges in teaching computational modelling in high school. First, there is no consensus on how computational modelling should be taught [32]. However, Sengupta and Wilensky [51] showed that agent-based models can foster students' computational thinking and modelling [51]. They noted the lack of consensus in science education on how teachers should use agent-based computer models in their classrooms. Second, high-school teachers might not be well-trained in computing. However, researchers have found agent-based modelling environments helpful for teaching students who are not familiar with computing [2, 66]. Specifically, agent-based modelling environments have been shown to support students' learning of subject matter [7].

The present study was a design based research study exploring teaching activities. We developed a template to classify and stimulate teachers' design of didactical questions. A template is a rubric that can help in the design and evaluation of teaching activities. It is generic in the sense that it uses a summarized format to guide teaching not only in specific subjects, but in general. The template was developed in collaboration with 86 teachers in 12 different high school subjects including the natural and social sciences, and the humanities. The participants came from 44 Danish high schools. Given the lack of research on teaching resources, we deemed it relevant to develop a template using a design based research design. Further studies will need to evaluate the validity and effectiveness of the template.

The study was guided by two research questions. First, what parameters should be included in a template aimed to help teachers design teaching activities about computational modelling? Second, can the developed template classify teachers' didactical questions about computational modelling?

In the paper, we begin by presenting the methodology and design of the template. Subsequently, we discuss how the parameters in the template were operationalized. Next, we evaluate the template. Finally, we discuss how it can be used to evaluate and guide the design of new teaching activities.

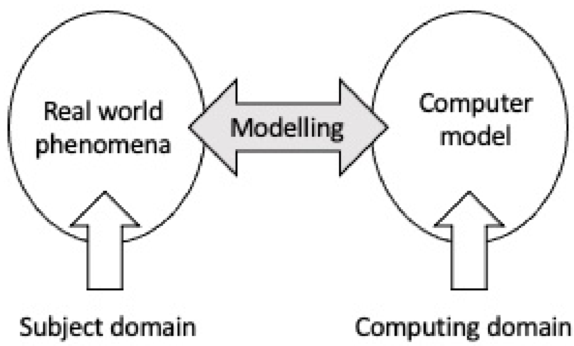


Figure 1: Computational modelling of a subject and a computing domain.

2 BACKGROUND

Researchers have made major contributions to different levels of understanding models and modeling in science [18, 40]. Also, there is important research on students' understanding of models including work on mental models [17, 60]. As pointed out by Krell, Upmeier zu Belzen, and Krüger [30] some studies suggest the existence of global levels of understanding models [20], others propose aspect-dependent levels [12]. According to Krell, Upmeier zu Belzen, and Krüger [30], research supports the idea of considering different aspects of models and modeling and propose the model of model competence as a theoretical framework. Inspired by Crawford and Culling [12], Krell et al. [30] derive at five aspects: nature of models, multiple models, purpose of models, testing models, and changing models, with three levels of understanding for each aspect. This research on complexity in understanding can be used in the classroom as part of an action cycle where students and teachers generate, evaluate and then modify models [9, 30]. At every step of this cycle, the science teacher can ask students for clarifications.

Clement [9] used the term discrepant question to refer to a thought-provoking question or problem that challenges students' existing scientific beliefs. Questions can be posed by the teacher in order to create a cognitive conflict between students' preconceptions versus the scientific models being taught by the teacher. Questions are important in teaching and doing science. Lehrer and Schauble [35] found in the context of teaching mathematics and science in K12 that data modeling can be taught through a chain of inquiry fueled by posing questions. Questions help the student and teacher deliberately selecting and amplifying particular attributes for further study. Passmore and Svoboda [45] argue that modelling in science is inherently an argumentative act. They formulate a model where inquiry-based science teaching evolves around what in a scientific community are worthwhile questions. Furthermore, in the field of computing education, Li et al. [36] argue that CT is more about thinking than computing and involves incrementally improving strategies for searching for ways of processing information. This involves asking questions about what achieves the most elaborate or efficient abstraction or computational model.

The research on models in education is interdisciplinary, and there is no easy way of informing teachers on the basis of the above literature on how to teach computational modelling. Empirical and theoretical research is needed to bridge these areas. Collaborative work is needed to help teachers in formulating an overview of questions that might lead to student inquiry into computational modelling. In summary, with this study, we aimed to design a template that served two purposes. First, for high school teachers to see examples of didactic questions that they could use in lessons and second for researchers to have a tool to classify teachers' questions.

Teachers might find computational thinking to be an important set of competences for students to gain. Sentance and Csizmadia [52] found that computational thinking was one of the five most frequently mentioned themes, that constitutes a successful strategy for teaching computing, by more than 300 K-12 computing teachers' evaluations. However, even if teachers perceive a need to teach computational thinking, teachers have different competencies. Yadav et al. [68] investigated pre-service teachers' perception of their ability to teach computational thinking. The study found that while teachers believe that computational thinking is an essential skill for students to learn, only one third of the teachers were confident they could teach computational thinking. There is no agreement amongst educational scientists on whether computational thinking should be taught solely in computing education classes or integrated into different high school subjects [58]. This challenge become even more complex when we consider both computational modelling and computational thinking, which are closely aligned but not identical. Hence, there is a need to teach teachers how to teach computational thinking and computational modelling as part of computational thinking.

But why and what should teachers teach? There is evidence that teaching students about modelling might aid the students to master other fields. A study by Schwarz and White [48] suggests that students' appropriation of knowledge about modelling can produce learning transfer. Furthermore, Chandrasekharan and Nersessian [8] found that computer models facilitate student thinking about natural phenomena and thus lead to new scientific insights. In essence, modelling seems to help students construe basic structures in their ways of making sense of both the everyday and scientific world [24]. A study by Jackson et al. [26] describes modelling as a process of making and using models to describe, explain, predict, and design natural phenomena. Competence in computational modelling draws upon proficiency in scientific modelling and computing [48, 54].

Through computational modelling, students learn to identify real-world phenomena and to find proper representations of these phenomena via concepts and abstractions embedded in computational models (see Figure 1). By relating one domain to another, students are able to understand a computational model [41, 44]. Figure 1 depicts computational modelling into two categories of domain: subject-specific and computer specific. The computational modelling process is an iterative process between a subject domain and a computing domain, and teachers' didactical questions related to the two domains can foster students' computational modelling process. A study by Sins et al. [53] found that students reasoned most efficiently during scientific modelling when they were able to focus on both parts and wholes of computer models. This finding led us to stipulate that teachers should present both parts and wholes of computer models. Research [43] have found that high school teachers can effectively teach computational modelling without first having to introduce complex computing concepts that underpin modelling. It motivates students to learn to grasp complex phenomena and when computational modelling allows students to visualize and manipulate complex phenomena, this is indeed reported by students as stimulating. Furthermore, in the secondary classroom, a study by Jacobson et al. [27] investigated learning designs involving different sequences for the structure of problem-based activities with agent-based computer models of the physics of electricity. They found that students participating in a low-to-high structure learning sequence scored higher on a post-test assessment of student conceptual understanding. In summary, these results concerning the structure and introduction of teaching activities can guide the design of teaching activities for students in computational modelling in the intricate relationship between computational thinking and computational modelling.

2.1 This study

In this study, we focused on didactical questions in order to help teachers to stimulate student problem-solving (see Table 3). Didactical questions refers vc to queries intended to challenge or probe student understanding. This is sometimes in the literature called Socratic questioning [46]. In this study, didactical questions and a computer model form the basis for a teaching activity. Teaching activities enable students to develop knowledge and skills required to achieve the desired educational outcome. Teaching activities can be a variety of activities including lectures, tutorials, concept mapping, and questioning [3]. The teaching activities in this study helped guide students to explore and modify computer models. The computer models represented subject-specific phenomena, which students then gained insight into. The didactical questions were formulated by teachers, who asked the students to relate the subject domain to the computing domain (see Figure 1).

In collaboration with high school teachers, we developed a general template with didactical questions relating to computational modelling in different high school subjects (see Table 3). The teaching template had four advantages from the perspective of teachers. First, it helped the teacher to plan the class. When teachers focused on a subset of the 16 parameters, the template was perceived as relatively easy to fill out. This helped the teacher to formulate a concrete plan for teaching activities in advance of teaching. Second, the template was a design tool that helped the teachers diagnose students' problems had in understanding scientific and computational models. Third, the identification of the problem helped the teacher to identify a teaching activity to ameliorate the problem. For instance, one teacher reported that the template helped the teacher identify a student, who struggled with the mental act of decomposing, by asking the student to identify the elements included in the computer model (see Table 3). Finally, the template helped high school teachers formulate didactic questions concerning modelling at different taxonomic levels.

3 METHOD

3.1 Research design

This study was inspired by design-based research, but contrary to most design-based research, the study included a comparison group. The Design-Based Research Collective [10] conceived of this type of research as both a methodology to study learning and improve educational practice. Design-based research has gained attention in computing education research [37] and in STEM education, for instance in biology [49]. This could be because of its breadth of scope, integrating the scientific study of student learning with a focus on teaching innovation where both learning and teaching are mediated by a novel design or technology.

We were inspired by the pioneering work by Collins et al. [11] on design research where researchers and educators collaborate in all phases of the research process from idea to implementation. This speaks to the fluidity and adaptability of this type of research as "Design-based research is not so much an approach as it is a series of approaches, with the intent of producing new theories, artefacts, and practices that account for and potentially impact learning and teaching in naturalistic settings" [1]. In designing this study, we took note of a review by Warr and Mishra [62] arguing that teachers should be viewed as designers or collaborators in the design process. Arguably, the teacher has been overlooked in design-based research [57]

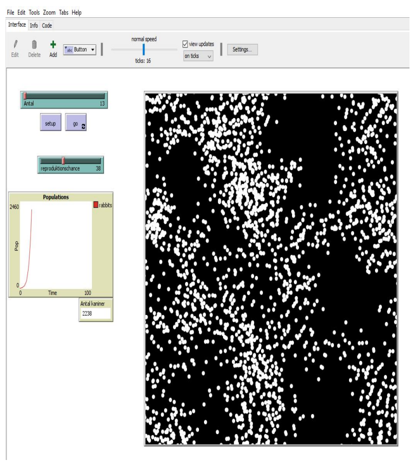
The teaching template was a design, which could only be developed, implemented, and evaluated in collaboration. Other design-based research studies have focused on teachers' professional development [61, 69]. Thus, the study focused on developing a template as part of teachers' professional development. The teachers in this study participated together with a research team of four experts in computing education, computer science, and educational psychology.

The design process involved four researchers and 86 teachers. The reason for using design based research to develop a template for teaching activities, rather than assessing effectiveness, was the lack of research on computational modelling in high school education.

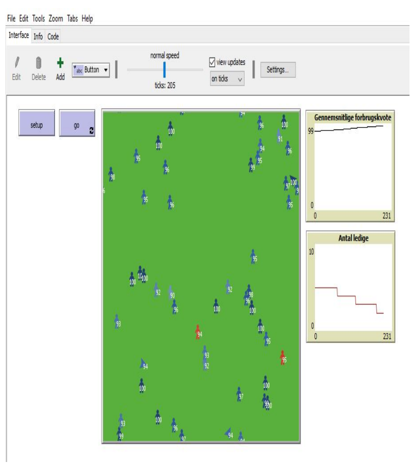
NetLogo [65] was used in two courses. NetLogo is a widely used agent-based programming environment developed specifically for educational purposes. Examples of computer models developed in the NetLogo programming environment and used by teachers can be seen in Table 1. According to Blikstein et al. [5], NetLogo has a "low threshold and high ceiling" (p. 1), meaning is it easy for novice programmers to work with. Furthermore, NetLogo is freely available online and regularly maintained by a group of experts. Research has demonstrated that NetLogo might facilitate students

Table 1: Examples of computer models with interface and code: Top: developed by a participant in the Prog+ group. Bottom: developed by an instructor in the Prog- group.

Interface	Code




```
breed [ employees employee ]
breed [ unemployees unemployee ]
turtles-own [ forbrugskvote ]
to go
if ticks >= 1000 [ stop ]
ask turtles [
move
set label (forbrugskvote) ]
ask unemployees [
update-color-unemployee ]
ask employees [
update-color-employee ]
tick
end
to setup
clear-all
ask patches [
set pcolor green ]
create-employees 45 [
set shape "person"
set color blue
set size 1.5
set forbrugskvote 90
setxy random-xcor random-ycor ]
create-unemployees 5 [
set shape "person"
set size 1.5
set forbrugskvote 90
setxy random-xcor random-ycor ]
reset-ticks
end
to move
right random 360
forward 1
end
to update-color-unemployee
set color scale-color red (forbrugskvote) 110 80
end
to update-color-employee
set color scale-color blue (forbrugskvote) 110 80
end
```




```
breed [rabbits rabbit]
to setup
clear-all
set-default-shape rabbits "circle"
create-rabbits Antal [
set color white
setxy random-xcor random-ycor ]
reset-ticks
end
to go
if count rabbits >= 10000
[ stop ]
ask rabbits
[ move
reproduce ]
tick
end
to move
right random 50
left random 50
forward 1
end
to reproduce
if random 100 < reproduktionschance
[ hatch 1 [ forward 3 ] ]
end
```

Table 2: Details of participants in group Prog- and Prog+.

Number of	Prog-	Prog+	Total
Teachers	22	64	86
High schools	11	33	44
Learning activities	12	39	51
Subject domains*	6	6	12

in the activity of modelling complex real-world phenomena [13, 66, 67].

The design-based research study consisted of four phases as following Reeves [47] and Herrington and Reeves ([23].

Phase 1: Problem analysis and exploration. Teachers and researchers identified relevant phenomena within the high school subjects that could be represented by an agent-based computer model. The researchers used an educational didactical approach called the CMC approach, which integrates coding (C) and modelling (M) activities with content matter (C). The CMC approach is a didactic framework to be used when high school teachers and researchers want to collaborate to design teaching activities in computational modelling [41, 42]. Teachers, in collaboration with researchers, then designed and developed computer models representing the subject phenomena. Initial teaching activities were designed by the teachers to engage students in the computational modelling process.

Phase 2: Development of template. The authors developed the template in collaboration with 86 high school teachers. This development fell into two phases: a.) Initial development of the template based on the teaching activities designed by teachers and researchers in phase 1, and on existing literature on modelling in high school science subjects [8, 19, 44, 48]. Researchers extracted "elements" from the literature, resulting in a list of potential parameters of the template. In coming up with the parameters, we used research on model-based thinking in science [31, 54] and research on what models are representations of versus what models can be used for in classrooms [19]. The elements and themes from the learning activities designed in phase 1 were discussed within the research team and the most prevalent themes and elements were identified, resulting in parameters 1, 4, 5, 6, 7, 11, 12, and 14 of the template. b.) Iterative refinement of the template. The template was presented to the teachers as a tool for designing and evaluating teaching activities. More parameters were added and rephrased in the form of didactical questions by the teachers in collaboration with researchers. The teachers used the template for designing teaching activities. The usefulness of the template was discussed among teachers and researchers. Observations were recorded by the researchers during the discussion. Two researchers then adjusted the template according to the discussion by introducing a total of 16 parameters and the division of the parameters into the subject domain and the computing domain (Table 3).

Phase 3: Implementation and evaluation in iterative cycles. The research team reflected on the temporary results from the previous phases to produce the final template. Specifically, this was developed by implementing teachers' suggestions for didactic questions and having teachers discuss and evaluate the usefulness of these questions after they had tested them in their classrooms. This phase produced the final formulation of questions and coding within the final version of the parameters (see Table 3).

Phase 4: Reflection to produce design principles. Once the template had been implemented, evaluated, and refined, the design was described (see Table 3 and the rubric in Table 4).

3.2 Evaluation of the template

To evaluate the template, this study used data from two professional development courses (Prog- and Prog+). The participants were in-service high school teachers assigned by the teachers' principals to either of the courses before the study. Thus, the assignment of participants was not random, but beyond the researchers' control. The two courses had identical duration and spanned an entire school year, with four seminar days approximately two months apart. The teachers participating in the Prog+ course were given an introduction to programming in NetLogo [65] for developing computer models. Prog- participants were not given this introduction but provided input on subject-specific phenomena, which the instructors used to program the computer models. During this process, teachers completed a questionnaire about the representation of the phenomena to help instructors develop the computer models. Items included for instance: Which agents should the model include, or can you describe the properties and behaviours of agents in the model?

3.3 Course instructions

There were four seminar days of teacher training:

Day 1: Developing computer models. On the first seminar day of both courses, teachers were taught computational modelling, specifically agent-based modelling. In the Prog- group, teachers, worked in groups of two to three, to identify a phenomenon within their subject domain (see Figure 1) suitable for a computer representation. Two instructors, skilled in programming and with similar teaching experience, were assigned to each course. In the Prog- group, teachers' identification of a subject specific domain and phenomenon provided input to instructors programming the computer models in NetLogo. The teachers completed a questionnaire about requirements for their model. This part lasted for two hours. The instructors then developed the computer models. In the Prog+ group, teachers learned to program computer models in NetLogo. A short introduction to NetLogo and design principles [66], which also lasted for two hours (on the first seminar day), was given to the Prog+ group. Apart from the two hours, the instructions were identical in content for the two groups. The first seminar day had the same duration for both groups. Teachers in both groups met with instructors to refine computer models in NetLogo for a 1.5 hours physical meeting between the first and second seminar days. The computer models were similar in terms of interface and code complexity. The code complexity was evaluated as the number of elements included in the interface and the lines and procedures in the code. Table 1 shows examples of computer models produced by either a participant or an instructor from each of the two courses.

Day 2: Developing teaching activities. Teachers were introduced to the CMC approach, didactical principles related to the principles of use-modify-create [33], guidance [27], and tinkering ([59]. The initially developed template was presented and discussed, and subsequently the teachers developed teaching activities during the second seminar day. In the time between the second and third seminar days, teachers applied the teaching activities and computer models in their teaching.

Day 3: Improving teaching activities. At this stage, we asked teachers to reflect on their experiences and reiterate their teaching activities in relation to the template. This way, teachers were encouraged to redesign the teaching activities. In the time between the third and fourth seminar days, teachers applied these redesigned teaching activities to a new class of students.

Day 4: Presenting computer models and teaching activities. On the fourth seminar day, participants presented and shared computer models and teaching activities with each other. We collected the teaching activities for analysis. The teaching activities and computer models can be accessed here: https:// https://library.ct-denmark.org/.

3.4 Participants

Participants were 86 in-service high school teachers (26 female, 60 male) who volunteered to participate. They came from 44 high schools sampled randomly from the whole of Denmark. The teachers taught 12 different subjects. Participants were assigned to either computational modelling and programming (Prog+) or modelling (Prog-) conditions (see Table 2). Teachers collaborated in groups of two to three to produce a total of 51 teaching activities. Less than 10% of the participating teachers in both courses reported having any programming experience before participating.

3.5 Classification of the didactical questions

Teachers were asked to formulate didactic questions that would facilitate students' understanding of computational modelling. The teachers' names were anonymized before teaching activities were coded. First, the first and the second author coded the same five randomly selected teaching activities independently, by referring each didactical question to a parameter in the template (Table 3). Second, the two researchers met, discussed disagreements, and refined the coding procedure. Third, the two researchers proceeded to independently code ten more teaching activities by referring each didactical question to the relevant modelling parameters of the template. The level of agreement between the two coders was recorded, and an Inter-Rater Reliability (IRR) across the two coders was established. We obtained an IRR of 0.87 (Cohen's d: 0.74, 0.95 confidence interval), indicating substantial agreement. Fourth, the first author coded the remaining teaching activities. A binary score (zero or one) was assigned to each parameter, depending on whether the teaching activity contained didactical questions relating to this specific parameter. To compare the scores of the teaching activities from the two different courses, the percentage of teaching activities containing didactical questions related to each parameter was calculated for each course. The frequencies of the 16 parameters were normally distributed, ascertained numerical by skewness and kurtosis [39]. A one-way chi-square analysis and one-way ANOVA (with one source of variability) analysis were performed to compare teaching activities from the two courses and to compare effects within subjects. We calculated the effect sizes, based on Cohen's d, and performed a one-way ANOVA analysis to examine the possible within-subjects effects. Subsequently, we selected representative didactical questions from the teachers' learning activities for each taxonomical level and parameter by two members of the research team and compared. One question for each parameter and taxonomical level was chosen by the researchers and presented in the template (Table 3).

In summary, teachers participated in either of two courses (Prog+, Prog-) and developed and applied computer models (in Prog+) or applied computer models (in Prog-) together with producing teaching activities on computational modelling by designing didactical questions.

4 RESULTS

4.1 Design of the template

The first research question was: What parameters should be included in a template aimed to help teachers design teaching activities about computational modelling? To answer this question, we developed a teaching template consisting of 16 parameters relating to the computational modelling process (Table 3).

The parameters classify teaching activities in terms of how well they might foster students' skills in computational modelling. The 16 parameters are divided into two categories: Subject and computing domain. The template was introduced to the teachers participating in this study, and feedback from the teachers incorporated into the final design of the template. All parameters of the initial template were perceived as useful by the teachers, but teachers reported a need for more parameters and for relating each parameter to either of the two domains described in Figure 1. This led to the result seen in Table 3.

The first 11 parameters related to the subject domain (see Figure 1). In formulating these parameters, we were inspired by Schwarz and White [48] who analysed modelling activities in science education. These activities were included to help students visualize complex concepts and test their conceptual understanding.

The last five parameters, numbers 12 to 16 (see Table 3), related to the computing domain and students' competencies in computational modelling (see Figure 1). Here we build on work done by Nowack and Caspersen [44] on computational model-based thinking and practice training students in identifying and relating concepts. Furthermore, we were inspired by a study by Lee et al. [33] prescribing a progression from student exploration and use of the computer model to modifying the model and finally creating a new or improved computer model.

4.2 Guiding the design of teaching activities

By answering the first research question, the aim was to guide researchers and teachers to develop teaching activities in computational modelling. The teaching activities from the two courses included on average 18 questions. Two members of the research team independently evaluated and scored teachers' didactical questions. The two researchers independently categorized the questions with reference to the SOLO taxonomy, which stands for the structure of observed learning outcomes [4]. The questions were anonymized and coded into three taxonomic levels: low, medium, or high (see Table 4 columns 2 to 4). Table 4 was constructed from the above analysis to inform and guide the development of new teaching activities. Table 4 provides examples of didactical questions formulated by teachers for their students. The examples of didactical questions presented in Table 4 are drawn from all 12 subjects represented by the high school teachers participating in the two courses (Prog+, Prog-). The examples can serve as guidelines for how to design

Table 3: Modelling parameters in template. Subject domain modelling (parameters 1-11) and computer domain modelling parameters 12-16)

Number	Modelling parameter
1	Range of the model
2	Explanatory power of the model
3	Causality between the elements in the model
4	Behaviour of the model
5	Elements included in the model
6	Elements not included in the model
7	How the model relates to a real-world phenomenon
8	Formulation of hypothesis that can be tested by the model
9	Results from repeated execution of the model
10	Emergent phenomenon developing over time
11	Use of and interaction with the model
12	Changing a value or variable
13	Adding a variable
14	Changing a procedure
15	Adding a procedure
16	Creating a new model

Table 4: Template with modelling parameters versus taxonomical levels of didactical questions (posed by teachers to students) in teaching activities. [] authors comments.

Taxonomical levels of questions	Low	Medium	High
1 Range of the model	Can the model show temperature?	What can the model predict?	Discuss whether the opportunities to influence input and output are realistic.
2 Explanatory power of the model	Name an element relevant for estimating the temperature of the atmosphere.	Which assumptions have been made as a basis for the model?	What are the implications of using computer models for policy-making?
3 Causality between elements of the model	Identify a water molecule and describe how it interacts with other molecules.	Add carbon dioxide to the atmosphere. What impact does this molecule have on the temperature of the Earth?	How does the effect of energy transfer change when material type and mass are changed?
4. Behaviour of the model	What happens when you press 'go'?	How can we see that green people react, reproduce, move differently than blue people?	Predict what happens to the dependent variables (Crime Index, Sense of Justice) when the independent variables (Length of sentence and number of police officers) are changed?
5 Elements included in the model	What do you see, when you press 'setup'?	How would you describe all elements in the interface, both the agents and the background?	Identify all the elements of the model, from what you see in the interface and can read in the code.
6. Elements not included in the model	Can you name the elements that are missing in the model?	Are there any elements missing in the model, or are there elements behaving differently than you would expect?	What is missing in the model, if any, compared to the experiment you have done?
7. How the model relates to a real-world phenomenon	Describe what the model depicts.	Compare the model to your own subject knowledge about the phenomenon - does it match? Why/why not?	Discuss in what ways the model is a simplification of reality.
8. Formulation of hypothesis that can be tested by the model	How will a water molecule behave when you press 'go'?	Make a prediction based on your model about how a Roman soldier can influence other agents in the model and how the simulation will turn out.	Find out how to set the independent variables and hypothesize what effect this will have on the dependent variables. Test your hypothesis by running the model.
9. Results from repeated execution of the model	Run the simulation five times. Is the result the same?	Change the albedo value [represented by a slider], run the simulation and note the effect. Repeat the 'experiment' many times. What effect does the albedo have?	Run the model a certain number of times and compare the outcome. Try changing the value of a variable and run the model again the same number of times. What is the effect of changing the value of the variable? How sure of the effect are you?
10. Emergent phenomenon developing over time	Describe what happens to the CO2 molecules when you run the model?	How many times, out of 10 runs, does it lead to a fall of the Roman Empire?	Explain whether the simulation always give the same result?
11. Use of and interaction with the model	Can you describe what happens to the reaction rate [visualized by a plot] when you increase the number of reactants [represented by a slider]?	Interact with the model. E.g. add clouds [button] or CO2 [slider] to the model. What happens to the reflection of the energy?	Experiment with the model. Can you evaluate which variables are changeable from the interface and which you can change in the code?
12. Change a value or variable	Find the term 'color' in the code. Can you change the color of the water molecule?	Change the number of persons [turtles] in the initial setting, by exploring the code: How many blue, black, pink persons are being created?	Change the color and shape of the enzyme [turtle] by identifying the relevant turtle described in the code. Why did you choose that particular color and shape?
13. Adding a variable	Add a shape like a square to the enzyme.	Introduce a new shape for 'person'. Do you need a different shape for each person created in the code?	Introduce a variable called birth age and let it be the age for when the rabbits [turtles] are sexually mature. Investigate how many places birth age should be entered in the code.
14. Changing a procedure	Change the movements of the water molecules to include backward movements.	You want to visualize that each year the rabbits give birth to two new cubs instead of one. Can you redesign the code so that this is visualized?	Expand the code, so that the decay of an atom can be one of three possibilities, rather than not just one. Run the program. Can you explain the effect that this has?
15. Adding a procedure	Introduce a limit for how many rabbits [turtles] the simulation can have before the program stops.	Use the 'wait' command. Can you make the ball [turtles] move slower?	Study the procedure for how the inhibitors [turtles] are created and behave. Then create a new type of inhibitor that behave differently.
16. Creating a new model	Create a new model by copying the first part of the code from this model. How much code is needed?	Try to put together all your small program pieces from the three previous models. Can you create a new model?	Create a new model by applying at least one of the rules described in this model. What phenomenon can the new model represent?

didactical questions related to specific parameters presented in the template, as part of the teaching activities.

4.3 Classifying didactical questions

To investigate possible differences in the two groups of participants, each teaching activity was scored. For every parameter in the template, the number of didactical questions related to this parameter was recorded. Figure 3 shows the prevalence of questions from the teaching activities representing each parameter, in percentage. Overall, there was a statistically significant difference didactical questions from the teaching activities between the two groups when tested by a one-way chi-square test (p < .001, level .95, n = 51) with a large effect size (Cohen's d = 0.8), see Figure 2.

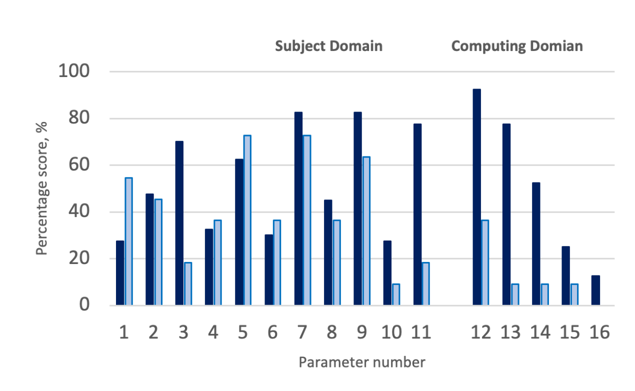


Figure 2: Percentage of teaching activities, containing questions relating to each parameter, in Prog+ (dark) and in Prog- (light). See Table 3 for a description of each parameter.

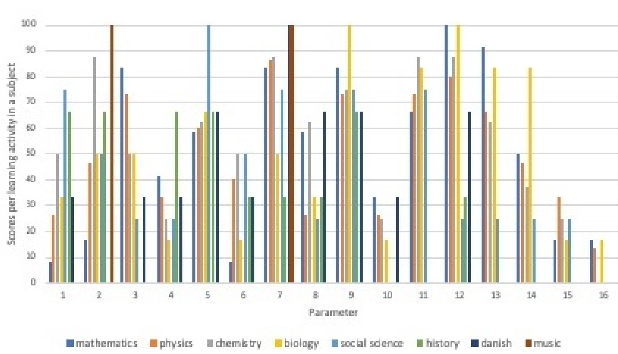


Figure 3: Average scores per parameter according to subject.

For parameters 1 to 11, there was no statistically significant difference in the distribution of parameters between the two groups of participants in course Prog+ and Prog- respectively (one-way chi-square test, p = .946, level .95, n = 51) with a small effect size (Cohen's d = 0.4). This indicated that both courses gave the teachers the ability to produce teaching activities that included didactical questions concerning modelling of the subject domain, as illustrated in Figure 1. However, a statistically significant difference was found between the two courses when comparing the percentages of parameters 12 to 16 by a one-way chi-square test (p = .004, level .95, n = 51) with a very large effect size (Cohen's d = 1.6). This indicated that teachers in the Prog+ group were better at designing teaching activities that entailed high taxonomic level didactical questions regarding the computational model than teachers in the Prog- group (see Figure 2). These parameters (12 to 16) represented questions concerning students' ability to modify and create code in the computer models.

The template was used to classify didactical questions in eight different school subjects. Table 5 shows the number of teaching activities representing each subject. To study effects within school subjects, we performed a one-way ANOVA test, only considering study subjects as the source of variability. Results showed that there were statistically significant differences in means between groups (subjects), (p = 0.001 F-value (F = 3.775) > F critical (F = 2.087)). Figure 3 illustrates the distribution of the average scores per parameter for each subject represented in the teaching activities. In general, teaching activities on computational modelling in music only represented a few parameters of the template, while subjects such as chemistry and biology were addressing all parameters in the didactical questions to a medium degree. Hence, the template could be used for guiding high school teachers in developing teaching activities for computational modelling containing more than one taxonomical level relating to the subject domain. The template also proved useful for evaluating teaching activities. Furthermore, when applying the template for evaluating teaching activities developed by teachers, the results showed differences in the developed teaching activities depending on the teachers' backgrounds.

5 DISCUSSION AND CONCLUSION

With this design-based research study, we aimed at developing a template for teaching activities in computational modelling. The template can be used as a starting point in dialogue between teachers and researchers for designing and conducting teaching activities in computational modelling. It can also be used to evaluate teaching activities and didactic questions in computational modelling. The data analysis suggested that the template could help teachers and researchers evaluate didactic questions developed by high school teachers. Results showed that Prog+ participants were able to produce computational modelling teaching activities with several questions referring to parameters from the template concerning the computer domain (parameters 12 to 16), while the Prog- participants were not. This indicates that achieving knowledge of programming positively affects teachers' ability to produce teaching activities for computational modelling, but raises questions as to what and how much knowledge teachers need for achieving this.

This study was novel in terms of researching computational modelling in high school subjects including physics, mathematics, chemistry, biology, social science, history, Danish, and music. The study contributed on methodological grounds by developing a template for guiding and evaluating high school education [38, 56]. Teachers reported that the template could help them formulate questions at different taxonomical levels appropriate for students who are learning computational modelling.

Table 5: Number of teaching activities produced by teachers in each subject.

Subject	Number of learning activities
Physics	14
Mathematics	12
Chemistry	8
Biology	6
Social Science	4
History	3
Danish	3
Music	1

There were two main limitations to this study. First, relating to the two groups and the fact that there was no control of the allocation of the participants to groups by the research team and the unequal distribution of the participants in each group. The unequal group size would have been problematic, particularly in a traditional intervention study. However, this study was a design based research study and did not aim to measure the intervention effect of computational modelling on student or teacher understanding. Further research studies, such as controlled designs on the learning effects of specific computational modelling environments, are needed. The teaching subjects represented in the two groups were different, although all participants in both groups reported that they were novice programmers. The fact that there were more natural science teachers in the Prog+ group could potentially explain why parameters of the computing domain were more prevalent in this group (Figure 2). A traditional assumption would be that the natural science teachers feel better educated, more interested in, and trained in how to think about computing than the humanities and social science teachers. The need to control for participants' background and knowledge is important for future research.

The second limitation concerns the application of the template. It could be objected that the parameters were not independent, but conceptually overlapping. This could have complicated the coding of the teaching activities, i.e. the assignment of each didactical question in the teaching activities to each parameter in the template. However, since we were able to code all the questions in the teaching activities (and thus refer all the questions to one of the 16 parameters), we saw no need to identify a clearer demarcation of individual parameters.

In spite of these limitations, this study responded to a growing need in research and educational practice for incorporating computational modelling in high school curricula and in a range of different subjects. Future studies could use classroom observations and interviews with teachers and students about the perceived usefulness of the template.

In conclusion, this study is a first step towards developing and applying a template intended for improving teaching activities in computational modelling within various existing high school subjects. It integrates parameters of computational modelling with levels of didactic questions. By delineating computational modelling into 16 parameters, we achieved a comprehensive yet manageable list of identifiable and measurable behavioural criteria. We hope that teachers will be inspired by the template to qualify their own teaching in something as complicated as computational modelling.

ACKNOWLEDGMENTS

The authors wish to thank the participating teachers and our colleagues at Center for Computational Thinking & Design for fruitful discussions. This work was funded by The Velux Foundations by Grant no. 24635 and Region Midt.

REFERENCES

[1] Sasha Barab and Kurt Squire. 2004. Design-based research: Putting a stake in the ground. *The journal of the learning sciences* 13, 1 (2004), 1–14.
[2] Todd K BenDor and Jürgen Scheffran. 2018. *Agent-based modeling of environmental conflict and cooperation.* CRC Press.
[3] John Biggs. 1999. What the student does: Teaching for enhanced learning. *Higher education research & development* 18, 1 (1999), 57–75.
[4] John B Biggs and Kevin F Collis. 2014. *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome).* Academic Press.
[5] Paulo Blikstein, Dor Abrahamson, and Uri Wilensky. 2005. Netlogo: Where we are, where we're going. In *Proceedings of the annual meeting of Interaction Design and Children, Press*, Vol. 23.
[6] Paulo Blikstein and Uri Wilensky. 2009. An atom is known by the company it keeps: A constructionist learning environment for materials science using agent-based modeling. *International Journal of Computers for Mathematical Learning* 14 (2009), 81–119.
[7] Whitney Wall Bortz, Aakash Gautam, Kemper Lipscomb, and Deborah Tatar. 2019. Integration Computational Thinking into Middle School Science: A search for Synergistic Pedagogy. In *ASEE Southeastern Section Conference.*
[8] Sanjay Chandrasekharan and Nancy J Nersessian. 2015. Building cognition: The construction of computational representations for scientific discovery. *Cognitive science* 39, 8 (2015), 1727–1763.
[9] John Clement. 2008. Six levels of organization for curriculum design and teaching. *Model based learning and instruction in science* (2008), 255–272.
[10] Design-Based Research Collective. 2003. Design-based research: An emerging paradigm for educational inquiry. *Educational researcher* 32, 1 (2003), 5–8.
[11] Allan Collins, Diana Joseph, and Katerine Bielaczyc. 2004. Design research: Theoretical and methodological issues. *The Journal of the learning sciences* 13, 1 (2004), 15–42.
[12] Barbara A Crawford and Michael J Cullin. 2004. Supporting prospective teachers' conceptions of modelling in science. *International Journal of Science Education* 26, 11 (2004), 1379–1401.
[13] Damian Dalle Nogare and Ajay B Chitnis. 2020. NetLogo agent-based models as tools for understanding the self-organization of cell fate, morphogenesis and collective migration of the zebrafish posterior Lateral Line primordium. In *Seminars in cell & developmental biology*, Vol. 100. Elsevier, 186–198.
[14] Peter J Denning. 2017. Remaining trouble spots with computational thinking. *Commun. ACM* 60, 6 (2017), 33–39.
[15] Peter J Denning and Matti Tedre. 2019. *Computational thinking.* Mit Press.
[16] Rachele Dominguez and Benjamin Huff. 2015. The role of computational physics in the liberal arts curriculum. In *Journal of Physics: Conference Series*, Vol. 640. IOP Publishing, 012061.
[17] Dedre Gentner and Albert L Stevens. 2014. *Mental models.* Psychology Press.
[18] Ronald N Giere. 1992. *Cognitive models of science.* Vol. 15. U of Minnesota Press.
[19] Julia Gouvea and Cynthia Passmore. 2017. 'Models of'versus 'Models for' Toward an Agent-Based Conception of Modeling in the Science Classroom. *Science &*

Education 26 (2017), 49–63.
[20] Lorraine Grosslight, Christopher Unger, Eileen Jay, and Carol L Smith. 1991. Understanding models and their use in science: Conceptions of middle and high school students and experts. *Journal of Research in Science teaching* 28, 9 (1991), 799–822.
[21] Shuchi Grover and Roy Pea. 2018. Computational thinking: A competency whose time has come. *Computer science education: Perspectives on teaching and learning in school* 19, 1 (2018), 19–38.
[22] Mark Guzdial, Alan Kay, Cathie Norris, and Elliot Soloway. 2019. Computational thinking should just be good thinking. *Commun. ACM* 62, 11 (2019), 28–30.
[23] Jan Herrington and Thomas C Reeves. 2011. Using design principles to improve pedagogical practice and promote student engagement. (2011).
[24] David Hestenes. 2013. Modeling theory for math and science education. *Modeling Students' Mathematical Modeling Competencies: ICTMA 13* (2013), 13–41.
[25] Nicole M Hutchins, Gautam Biswas, Miklós Maróti, Ákos Lédeczi, Shuchi Grover, Rachel Wolf, Kristen Pilner Blair, Doris Chin, Luke Conlin, Satabdi Basu, et al. 2020. C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology* 29 (2020), 83–100.
[26] Jane Jackson, Larry Dukerich, and David Hestenes. 2008. Modeling Instruction: An Effective Model for Science Education. *Science Educator* 17, 1 (2008), 10–17.
[27] Michael J Jacobson, Beaumie Kim, Suneeta Pathak, and BaoHui Zhang. 2015. To guide or not to guide: issues in the sequencing of pedagogical structure in computational model-based learning. *Interactive Learning Environments* 23, 6 (2015), 715–730.
[28] Michael J Jacobson, James A Levin, and Manu Kapur. 2019. Education as a complex system: Conceptual and methodological implications. *Educational researcher* 48, 2 (2019), 112–119.
[29] Dennis Kafura, Austin Cory Bart, and Bushra Chowdhury. 2015. Design and preliminary results from a computational thinking course. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. 63–68.
[30] Moritz Krell, Annette Upmeier zu Belzen, and Dirk Krüger. 2014. Students' levels of understanding models and modelling in biology: Global or aspect-dependent? *Research in science education* 44 (2014), 109–132.
[31] Mark J Lattery. 2016. *Deep Learning in Introductory Physics: Exploratory Studies of Model? Based Reasoning*. IAP.
[32] Irene Lee, Shuchi Grover, Fred Martin, Sarita Pillai, and Joyce Malyn-Smith. 2020. Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology* 29 (2020), 1–8.
[33] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. Computational thinking for youth in practice. *Acm Inroads* 2, 1 (2011), 32–37.
[34] Richard Lehrer and Leona Schauble. 2006. *Cultivating model-based reasoning in science education*. Cambridge University Press.
[35] Richard Lehrer and Leona Schauble. 2013. Modeling in mathematics and science. In *Advances in instructional psychology*. Routledge, 101–159.
[36] Yeping Li, Alan H Schoenfeld, Andrea A diSessa, Arthur C Graesser, Lisa C Benson, Lyn D English, and Richard A Duschl. 2020. Computational thinking is more about thinking than computing. , 18 pages.
[37] Lauren E Margulieux, Brian Dorn, and Kristin A Searle. 2019. *Learning sciences for computing education*. Cambridge: Cambridge University Press.
[38] André Menolli and João Coelho Neto. 2022. Computational thinking in computer science teacher training courses in Brazil: A survey and a research roadmap. *Education and Information Technologies* 27, 2 (2022), 2099–2135.
[39] Prabhaker Mishra, Chandra M Pandey, Uttam Singh, Anshul Gupta, Chinmoy Sahu, and Amit Keshri. 2019. Descriptive statistics and normality tests for statistical data. *Annals of cardiac anaesthesia* 22, 1 (2019), 67.
[40] Mary S Morgan and Margaret Morrison. 1999. *Models as mediators: Perspectives on natural and social science*. Number 52. Cambridge University Press.
[41] Line Have Musaeus and Peter Musaeus. 2019. Computational thinking in the Danish high school: Learning coding, modeling, and content knowledge with NetLogo. In *Proceedings of the 50th ACM technical symposium on computer science education*. 913–919.
[42] Line Have Musaeus and Peter Musaeus. 2021. Computing and gestures in high school biology education. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 533–539.
[43] Line Have Musaeus, Deborah Tatar, and Peter Musaeus. 2022. Computational Modelling in High School Biology: A Teaching Intervention. *Journal of Biological Education* (2022), 1–17.
[44] Palle Nowack and Michael E Caspersen. 2014. Model-based thinking and practice: A top-down approach to computational thinking. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*. 147–151.
[45] Cynthia M Passmore and Julia Svoboda. 2012. Exploring opportunities for argumentation in modelling classrooms. *International Journal of Science Education* 34, 10 (2012), 1535–1554.
[46] Richard Paul and Linda Elder. 2007. Critical thinking: The art of Socratic questioning. *Journal of developmental education* 31, 1 (2007), 36.
[47] Thomas Reeves. 2006. Design research from a technology perspective. In *Educational design research*. Routledge, 64–78.
[48] Christina V Schwarz and Barbara Y White. 2005. Metamodeling knowledge: Developing students' understanding of scientific modeling. *Cognition and instruction* 23, 2 (2005), 165–205.
[49] Emily E Scott, Mary Pat Wenderoth, and Jennifer H Doherty. 2020. Design-based research: a methodology to extend and enrich biology education research. *CBE—Life Sciences Education* 19, 2 (2020), es11.
[50] Pratim Sengupta, John S Kinnebrew, Satabdi Basu, Gautam Biswas, and Douglas Clark. 2013. Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies* 18 (2013), 351–380.
[51] Pratim Sengupta and Uri Wilensky. 2009. Learning electricity with NIELS: Thinking with electrons and thinking in levels. *International Journal of Computers for Mathematical Learning* 14 (2009), 21–50.
[52] Sue Sentance and Andrew Csizmadia. 2017. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies* 22 (2017), 469–495.
[53] Patrick HM Sins, Elwin R Savelsbergh, and Wouter R van Joolingen. 2005. The Difficult Process of Scientific Modelling: An analysis of novices' reasoning during computer-based modelling. *International Journal of Science Education* 27, 14 (2005), 1695–1721.
[54] Yanlong Sun, Wendy Newstetter, and Nancy J Nersessian. 2006. Promoting model-based reasoning in problem-based learning. *Trabajo presentado en la reunión anual de la Cognitive Science Society, Vancouver, Canadá* (2006).
[55] Julia Svoboda and Cynthia Passmore. 2013. The strategies of modeling in biology education. *Science & Education* 22 (2013), 119–142.
[56] Xiaodan Tang, Yue Yin, Qiao Lin, Roxana Hadad, and Xiaoming Zhai. 2020. Assessing computational thinking: A systematic review of empirical studies. *Computers & Education* 148 (2020), 103798.
[57] Jan Van den Akker. 1999. Principles and methods of development research. *Design approaches and tools in education and training* (1999), 1–14.
[58] Joke Voogt, Petra Fisser, Jon Good, Punya Mishra, and Aman Yadav. 2015. Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies* 20 (2015), 715–728.
[59] Aditi Wagh, Kate Cook-Whitt, and Uri Wilensky. 2017. Bridging inquiry-based science and constructionism: Exploring the alignment between students tinkering with code of computational models and goals of inquiry. *Journal of Research in Science Teaching* 54, 5 (2017), 615–641.
[60] Chia-Yu Wang and Lloyd H Barrow. 2011. Characteristics and levels of sophistication: An analysis of chemistry students' ability to think with mental models. *Research in Science Education* 41 (2011), 561–586.
[61] Shiang-Kwei Wang, Hui-Yin Hsu, Thomas C Reeves, and Daniel C Coster. 2014. Professional development to enhance teachers' practices in using information and communication technologies (ICTs) as cognitive tools: Lessons learned from a design-based research study. *Computers & Education* 79 (2014), 101–115.
[62] Melissa Warr and Punya Mishra. 2021. Integrating the discourse on teachers and design: An analysis of ten years of scholarship. *Teaching and Teacher Education* 99 (2021), 103274.
[63] Mary Webb, Niki Davis, Tim Bell, Yaacov J Katz, Nicholas Reynolds, Dianne P Chambers, and Maciej M Sysło. 2017. Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies* 22 (2017), 445–468.
[64] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology* 25 (2016), 127–147.
[65] Uri Wilensky. 1999. *NetLogo*. http://ccl.northwestern.edu/netlogo/
[66] Uri Wilensky and William Rand. 2015. *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. Mit Press.
[67] Uri Wilensky and Kenneth Reisman. 2006. Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and instruction* 24, 2 (2006), 171–209.
[68] Aman Yadav, Elisa Nadire Caeli, Ceren Ocak, and Victoria Macann. 2022. Teacher Education and Computational Thinking: Measuring Pre-service Teacher Conceptions and Attitudes. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. 547–553.
[69] Doron Zinger, Ashley Naranjo, Isabel Amador, Nicole Gilbertson, and Mark Warschauer. 2017. A design-based research approach to improving professional development and teacher knowledge: The case of the Smithsonian learning lab. *Contemporary Issues in Technology and Teacher Education* 17, 3 (2017), 388–410.