

Developing Students' Computational Thinking and Subject Knowledge through Computational Modeling in Secondary Education

A Dissertation

Presented to the Faculty of Arts

Center for Computational Thinking & Design (CCTD)

Aarhus University

Denmark

by

Line Have Musaeus

October, 2021



supported by
it-vest
networking universities

Supervisor:

Prof. Ole Sejer Iversen, CCTD, Aarhus University, Denmark

Co-supervisors:

Prof. Deborah Tatar, Virginia Tech, VA, USA

Prof. Michael E. Caspersen, It-vest – networking universities, Denmark

Assessment Committee:

Louise Mifsud, Oslo Metropolitan University, Norway

Mark Guzdial, University of Michigan, MI, USA

Clemens Nylandsted Klokmose, Aarhus University, Denmark

Financial support:

The work was supported by It-vest – networking universities

(Grant no. AU-2018-40)

List of scientific papers

This dissertation is based on the following five papers:

Paper 1	Computational Thinking in the Danish High School: Learning Coding,
Study I	Modeling, and Content Knowledge with NetLogo
	Musaeus, L. H., & Musaeus, P. (2019)

In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (pp. 913-919)

- Paper 2
Study II A Template for Teaching Computational Modelling: A Design Based Research Study in High School
Musaeus, L. H., Nowack, P., Caspersen, M. E., Musaeus, P.
Submitted to Journal of Education and Information Technologies, October, 2021
- Paper 3
Study III Learning Computing and Biology from Computational Modeling: An Intervention Study in High School
Musaeus, L. H., Tatar, D., Musaeus, P.
Submitted to Journal of Biological Education, August, 2021
- Paper 4
Study IV Computing and Gestures in High School Biology Education
Musaeus, L. H., & Musaeus, P. (2021)
In Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 533-539)
- Paper 5
Study V Empowerment through Computational Thinking: A Literature Survey
Musaeus, L. H., Mechelen, M. V., Caspersen, M. E., Iversen, O. S.
Submitted to Acta Didactica Nordic, October, 2021

Acknowledgements

It is a privilege and a great pleasure to have worked with so many people who helped make this dissertation possible.

A special thank you to:

Ole Sejer Iversen, my supervisor, for being so patient with my background in biology. Also for introducing me to new research areas and to the academic life in an interdisciplinary research center.

Michael E. Caspersen, my co-supervisor, for introducing me to computing education research and suggesting I pursued this work. He has tirelessly inspired, believed, and supported me and my work.

Deborah Tatar, my co-supervisor, for sparking my interest in computational thinking by inviting me to participate in her own work, and for never failing to point me in the right research direction.

Palle Nowack, my close friend and colleague, for inviting me on a winding (road-)trip to the land of computational modeling, and for never taking the easy way back.

Keld Nielsen, my much appreciated colleague, for tireless discussions and for insisting on the relevance of our work.

Peter Musaeus and the rest of my family, for being so kind as to bear with me throughout this process.

Numerous people have contributed to my research. Ole Sejer Iversen, Michael E. Caspersen, Deborah Tatar, Palle Nowack, Peter Musaeus, Maarten P. van Mechelen, and Hermes Arthur Hjort are coauthors of one or more of my publications.

Thanks to all my colleagues at CCTD for providing an atmosphere of friendship, support, trust, and professional ambitions.

Hundreds of students and teachers have volunteered as subjects in various studies during this work. For their help, I am truly grateful.

Last but not least, thank you to It-vest – networking universities for funding my doctoral studies.

*Line Have Musaeus
October, 2021*

Table of Contents

List of scientific papers	3
Acknowledgements	4
Abstract	6
Danish resumé	7

1 Introduction	8
2 Background	21
3 Method	34
4 Findings	56
5 Discussion	65
6 Conclusion	73
7 Dissemination	74
References	75
List of appendix	86

Abstract

Computational thinking has become a core element in many countries' attempts to integrate computing into curricula at various levels of education. Computational modeling should be conceived of as part of computational thinking in K-12 education. Computational modeling has been shown to improve students' content knowledge in existing subjects in K-12 education. However, there is a research knowledge gap in terms of how computational modeling and coding is improving students' content knowledge of existing subjects in high school.

The aim of the present dissertation is to investigate how computational modeling integrated into an existing subject in high school can foster students' skills and competences in computing.

Computer models and teaching activities were developed and applied in different high school classes in Denmark. More than 350 students and 100 teachers participated in the studies which were designed as both intervention and processual studies.

The PhD study contributes to research in the field of computing education. More specifically it touches the areas of students and teachers, schools, and society and addresses aspects of computing education research such as: what is and how can computing education be established in high school education, and what do students' gain from participating in computing education in high school.

This PhD study offers five contributions to the research field. First, an applicable framework for designing and teaching computational modeling and for students' to tinker with learning computational thinking. Second, a template for evaluating teaching activities in computational modeling has been developed as a viable way for predicting the efficacy of teacher-designed activities. Third, a taxonomy of computational thinking and gestures for students' learning of computational thinking have been developed and successfully applied in a case study. Fourth, an assessment of students' learning in both computational modeling, coding, and subject matter have been performed and potential synergetic effects investigated. Statistically significant improvements in students' learning have been identified. Finally, a literature survey of the use of empowerment as a term in relation to computational thinking in education have been conducted and shortcomings made visible.

Danish resumé

'Computational thinking' er blevet et centralt element i mange landes forsøg på at integrere computerundervisning, kaldet 'computing', i læseplanerne på forskellige uddannelsesniveauer. Computermodellering bør opfattes som en del af computational thinking i K-12-undervisningen. Det er blevet påvist, at computermodellering forbedrer elevernes viden om indholdet i eksisterende fag i K-12-undervisningen. Der er imidlertid et område i forskningen der ikke er fuldt belyst med hensyn til, hvordan computermodellering og programmering forbedrer elevernes viden om indholdet af eksisterende fag i gymnasiet.

Formålet med denne afhandling var at undersøge, hvordan computermodellering integreret i et eksisterende gymnasiefag kan fremme elevernes færdigheder og kompetencer inden for computing.

Computermodeller og undervisningsaktiviteter blev udviklet og anvendt i forskellige gymnasieklasser i Danmark. Mere end 350 elever og 100 lærere deltog i undersøgelserne, der var udformet som både interventions- og procesundersøgelser.

Denne afhandling bidrager til forskning indenfor undervisning i computing. Mere specifikt berører den områder aktuelle for elever og lærere, skoler og samfund. Den belyser aspekter der omhandler forskning i computing såsom: hvad er og hvordan kan computing uddannelse etableres i gymnasiet, og hvad får eleverne ud af at deltage i computing i gymnasiet.

Denne afhandling tilbyder fem bidrag til forskningsfeltet, idet den først har udviklet en ramme for udformning og undervisning i computermodellering og for elevernes arbejde med at lære computational thinking. For det andet er der udviklet en skabelon til evaluering af undervisningsaktiviteter i computationel modellering som en brugbar måde at forudsige effektiviteten af lærerudformede aktiviteter på. For det tredje er der udviklet en taksonomi af computational thinking og gestus for elevernes læring af computational thinking, som med succes er blevet anvendt i et casestudie. For det fjerde er der foretaget en vurdering af elevernes læring inden for både computermodellering, kodning og faglige emner, og eventuelle synergieffekter. I den forbindelse er der identificeret statistisk signifikante forbedringer i elevernes læring. Endelig er der foretaget en litteraturundersøgelse af brugen af empowerment som et begreb i forbindelse med computational thinking i undervisningen, og mangler er blevet synliggjort.

1 Introduction

Computing education is gaining momentum, and is being implemented into school curricula worldwide. Since computing was introduced in education there have been discussions of how to assure a high quality and a democratic control of the digitalization of the world, and the importance of general empowerment was already discussed by Turing Laureate Peter Naur in 1967 "This [computing in general education] is a necessary condition for humankind's supremacy over computers and for ensuring that their use do not become a matter for a small group of experts, but become a usual democratic matter, and thus through the democratic system will lie where it should, with all of us" (Naur, 1967). However, computing education

research is still needed in order to ensure a high quality and democratic perspective of computing education. Hence, to address the questions of how and what to teach in computing education and when to teach what.

The ability to think computationally is an important element of computing education.

Computational modeling concerns the ability to develop, use, and modify computer models and is a fundament for computational thinking (CT). Computational modeling is often related to computer models representing real world problems and connected to domain specific knowledge traditionally taught in existing subjects in education. Hence, computational modeling in education is a process where students model a domain specific phenomenon and should be a part of CT in K-12 education (Lee et al., 2020a). However, previous research has not explicated the links between students' content knowledge in existing subjects in K-12 education and computational modeling and a research knowledge gap of if and how the part of computational modeling concerning programming and coding is improving students' content knowledge of existing subjects exist.

The PhD study is the product of an interdisciplinary approach drawing on content knowledge from biology and biotechnology, integrated with computing and interaction design in order to best address the questions of how and what computational modeling can contribute to computing and existing education. The dissertation presents a first step toward filling the research gaps related to these questions. The approach has been to first develop an applicable framework for designing and integrating learning activities in CT and computational modeling, as an important addition to existing subjects in high school. Second, to apply the framework for designing sustainable professional development courses for in-service high school teachers on CT and computational modeling, and to develop a template for evaluating teaching activities developed by teachers participating in the courses. Third, the approach was then to apply the framework and template to design learning activities for students in existing high school biotechnology education. This was followed by interventions and assessments of students' learning of CT, computational modeling, and of biotechnology subject matter when participating in the learning activities. Students' learning of CT was assessed by two mixed methods studies, providing insights into students embodied learning and CT. Fourth, the PhD study then focused on how to classify aspects of potential empowerment by CT in education. This resulted in the final study of the PhD study. Hence, the PhD study is situated in computing education research and address how computing can be established in high school education, and why computing education should be offered to

students as part of their education. Specifically, the PhD study focuses on the integration of computing education into biotechnology.

1.1 Overall research question and contributions

1.1.1 Research question

The PhD study focusses on integrating coding and computational modeling, as core elements of CT, into existing subjects in Danish high schools by design-based interventions using predesigned computational models and learning activities. Hence, the PhD study has educational and scientific relevance by studying important aspects of computing education. This leads to the following overarching research question:

What and how do students learn by integrating computational modeling and coding into Biotechnology?

1.1.2 Contributions

The main impetus for the PhD study was to research how computational modeling can be combined with coding and specific content knowledge in meaningful learning activities for high school students and teachers. The computing aspect of the PhD study is highly inspired by, and focused on, computational modeling as a central part of CT (Nowack & Caspersen, 2014; Weintrop et al., 2016; Guzdia et al., 2019) using an agent-based modeling and programming environment. The research focused on what and how students learn in regards to both computing, in the form of coding and computational modeling, and biotechnology by this approach. Finally, the PhD study researched the affordances of integrating CT in education.

The contributions of the five studies are described in five papers. The contributions are related to how to integrate computational modeling in existing curricula in the form of a framework for integrating coding and computational modeling with content knowledge (CMC), and a template for developing learning activities in computational modeling relating to subject matter in a curriculum in paper 1 and 2, respectively. The contributions are also related to what students learn by integrating computational modeling into education, and to what elements of computational modeling can successfully be integrated. These contributions

are in the form of assessments of students' learning of both computational modeling, coding and content knowledge in paper 3. Students' learning is investigated by traditional tools, such as a content inventory, but investigations into the embodied learning of computing by students are also described in paper 4. Building on the contributions described above, the fifth study investigates the term empowerment in relation to CT education in research literature. The study present contributions related to why computing should be integrated into education in a broader perspective in the form of a literature survey. By applying an existing characterization tool, the distribution of the use of 'empowerment' in connection to computing education is investigated and differences in the use and focus of empowerment identified.

The contributions from the five studies fall into two main categories: 1. CT and computational modeling in education and 2. Students' learning of CT and subject matter. These are summarized in the following (See Table 1).

Paper (see also appendix A, B, C, D, E)	1: <u>Computational Thinking in the Danish High School</u> (Published, SIGCSE2019).	2: A Template for Teaching Computational Modelling: A Design Based Research Study in High School (Submitted, Journal of Education and Information Technologies).	3: Learning Computing and Biology from Computational Modeling: An Intervention Study in High School (Submitted, Journal of Biological Education).	4: Computing and Gestures in High School Biology Education (Published, ITICSE2021).	5: Empowerment through Computational Thinking: A Literature Survey (Submitted, Journal of Acta Didactica Nordic).
Contribution					
CT and computational modeling in education: the <i>how</i> and <i>why</i> in Figure 1:					
Framework (CMC)	X				
Template (learning activities)		X			
Design of professional development for teachers	X	X			
Characterization of empowerment (literature survey)					X
Students' learning of CT and subject matter: the <i>what</i> in Figure 1:					
Assessment (CT integrated into subject)	X		X	X	
Synergy effect (CT and subject)			X		
Taxonomy (of CT and gestures)				X	
Students' embodied learning of computing				X	

Table 1. Contributions to the overall research question from the five papers included in this dissertation in relation to Figure 1.

1.1.3 Additional hypotheses and research questions

From the research question posed in section 1.1.1, other hypotheses (H) and research questions (RQ) can be derived. Hence, below four hypotheses (H) and research questions (RQ) are presented. These are by no means all the research questions that can be derived from the overarching research question in section 1.1.1. The research questions posed below will be discussed in section 5 after the findings of this PhD study have been presented.

The first two (A and B) relates to *how* in the overall research question, while C and D relates to *what* in the overall research question. These will be discussed later in the dissertation.

H(A):

Students understand and discuss the relevant phenomenon at a higher level of abstraction when participating in learning activities designed for study I through IV.

RQ(A):

How do integrating computational modeling into existing subjects change the progression of *what* can be taught to students *when* in regard to existing subjects?

H(B):

The framework, taxonomy, and template developed during this PhD study are not developed specifically to the subject of biotechnology and can be applied within other subjects in high school education, as findings from study two indicates.

RQ(B):

Where is it valuable to integrate computational modeling into which subject in high school education?

H(C):

Thinking computationally can be applied by students in other subjects and outside school, and empower students to understand and meaningfully participate in computational practices in different settings.

RQ(C):

Do students obtain competences in computational modeling that transfer to other subjects or even to situations outside the classroom?

H(D):

Integrating CT into an existing curriculum can be accomplished by relative short interventions, as indicated in study I through IV, and do not have to sacrifice or reduce elements of the existing curriculum.

RQ(D):

How can teaching computational modeling in existing subjects in high school take place within an existing curriculum?

Hypothesis A and B are related to paper 1 in which a fruitful framework for how to integrate computational modeling is presented and to paper 2 which contributes with the development of a template for designing learning activities within the framework presented in paper 1. Hypothesis C and D are related to paper 3 and 4 which assess students' learning of computational modeling and thinking and of subject matter. Hypothesis D is also related to paper 1 in which short interventions were designed and shown to be fruitful for students learning of computational modeling. Hypothesis C is concerned with what competences students gain by participating in learning activities integrating computational modeling into existing subjects in education and with how these competences might transfer to other subjects and situations of students' life leading to students' computational empowerment. Hence, hypothesis C is also related to paper 5 which contributes with a survey of the term empowerment as it is used in research literature concerning CT in education.

1.2 Papers

Figure 1 illustrates the area of research for this PhD study, by positioning students, schools (including teachers), and society as three main elements of the research trajectories of this PhD study.

Figure 1 also illustrates how the contributions are distributed as described in the five papers (number 1 through 5) and in relation to students, teachers, schools, and society (see Figure 1). The contributions are within the field of computing education. The work is described in five papers (Table 1). The papers represent the research process of unraveling some of the underpinnings regarding how and what elements of CT and modeling can be introduced into existing high school education.

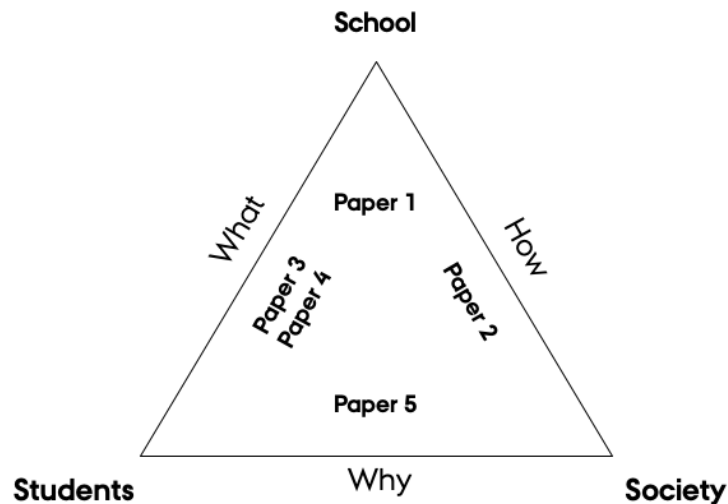


Figure 1. Positions of and contributions from the five papers of this dissertation.

Figure 1 illustrates the research trajectories of the PhD study and how the papers included in the dissertation are positioned within these trajectories. Paper 1 address the question of ‘what’ computing education can be in a high school setting and ‘how’ to integrate it into high school education. The paper is positioned at the top of Figure 1. Paper 2 is concerned with how to foster teachers’ understanding of computational modeling in relation to their own teaching and as such is positioned between school and society on the ‘how’ axis in Figure 1. Papers 3 and 4 describe work on investigating students’ learning of computational modeling in the classroom. Both traditional and more novel methods are use, as are both qualitative and quantitative methods. Both papers are positioned between students and school on the ‘what’ axis in Figure 1. Paper 5 presents a survey on existing literature about CT in education and its relation to computational empowerment. The paper concerns the ‘why’ (see Figure 1), i.e. the societal rationale for studying CT in education.

Paper 1 address both the question of what students learn by participating in coding and computational modeling activities and how to integrate these activities into existing subjects in high school curricula. The paper establishes a framework for these challenges which place the paper between the ‘what’ and ‘how’ in Figure 1.

There is a growing interest in society in general of how to educate future students in computing. Paper 2 is concerned with exploring further how the integration of coding and computational modeling can be integrated into existing high school curricula, and find the education of teachers fundamental for this question. Hence, the paper addresses how to

educate in-service high school teachers in coding and computational modeling by designing PD courses and is positioned accordingly in Figure 1.

Paper 3 and 4 relate to the ‘what’ (see Figure 1). The papers address what students learn of both computing and biotechnology while participating in learning activities of coding and computational modeling. The methods of assessments of students’ learning are different in the two papers in order to accommodate different perspectives on learning. Paper 3 uses traditional assessment tools for assessing learning in both CT and modeling and in biotechnology. Paper 4 takes an embodied learning perspective and explores the gestures students produce while coding and modeling computationally.

The final paper (Paper 5) addresses the question of why students and teachers should be educated in computing. It investigates if and how computing education researchers, as a community, uses the term empowerment in the literature. The paper is positioned accordingly, between student and society and in relation to ‘why’, in Figure 1.

Paper 1. Computational Thinking in the Danish High School: Learning Coding, Modeling, and Content Knowledge with NetLogo.

Paper 1 is reporting on the development of a framework for integrating coding, modeling and content into existing subjects in high school education. This framework is called the CMC framework and have been evaluated successfully by teachers and students in high schools in Denmark. The paper poses the hypotheses:

H1.1 Students’ acquisition of CT and content knowledge can be facilitated by the CMC framework.

H1.2 Students acquire new tools for exploring the phenomenon within the existing subject by participating in learning activities design by the CMC framework.

Integrating modeling and content has been addressed before (Schwarz & White, 2005; Passmore et al., 2009; 2017). But integrating coding, modeling, and content is new. The NSF funded project C+STEM (Gautam et al., 2020) is one of the relatively few studies exploring ways of working with code in relation to subject matter. However, the study presents only pieces of code to the students and not a complete program. Also, the study addresses primary school children and not high school students as is the case in this PhD study.

The research question in paper 1 is:

RQ1.1

Can the CMC (coding, modeling, and content) approach aid researchers, developers, and teachers to produce learning environments for students' acquisition of CT and content knowledge?

The paper evaluates the findings from an open-ended questionnaire with all participating students (n=210) and semi-structured interviews with all teachers (n=15). Thematic analysis was applied to categorize the qualitative data. The framework was successfully tested in high schools in Denmark and the paper concludes that students gained knowledge in CT and content knowledge in the existing subject through working with computer models of phenomena in different subjects. The study also showed that by letting students tinker with computer models they were able to integrate both coding, modeling and content knowledge.

Paper 2. A Template for Teaching Computational Modeling: A Design Based Research Study in High School.

Given that a successful framework for implementing coding and computational modeling and content knowledge in high school education (described in paper 1) have been established, paper 2 addresses the question of what computational modeling looks like in a high school setting. An evaluation template for evaluating teaching activities was developed by formulating parameters of computational modeling. The paper poses the hypotheses:

H2.1 A template can evaluate teaching activities in computational modeling developed by high school teachers.

H2.2 By teaching core elements of computational modeling, including programming, high school teachers can produce high quality teaching activities within the CMC framework.

Again, modeling in science education has been addressed before, and even the use of computational models (Tisue & Wilensky, 2004; Sengupta et al., 2018; Basu, 2014; 2016a; 2016b) but given that the coding aspect of computational modeling is new to this research area, so is the professional education (PD) of teachers in this field. The computational modeling parameters were used to evaluate learning activities produced by high school teachers in PD courses on computational modeling, producing the following research question:

RQ 2.1

How can questions stimulate teaching activities for high school students in agent-based computational?

The paper reports on evaluation of two professional development courses on computational modeling - one emphasizes programming, one does not. The study covers 12 high school subjects, 44 high schools, 51 learning activities, and 86 teachers. The paper concludes that the developed template proved useful as shown by findings of independent researchers' coding of teaching activities produced by high school teachers. The evaluation template highlights similarities and differences in teaching activities related to parameters of computational modeling.

Paper 3. Learning Computing and Biology from Computational Modeling: An Intervention Study in High School.

Paper 3 researches if students learn both CT and content knowledge in an existing subject when participating in learning activities designed by the CMC framework, and if they do, do the students learn the existing subject matter better? The design was a two-group randomized intervention study using mixed methods. The purpose was to integrate computational modeling in high school biology. A two-group intervention study of computer models in biology was designed. Participants were 118 students (17 to 19 years of age) enrolled in first and second year of Danish High School. The intervention group (n = 81) received teaching in both biology and computational modeling while the comparison group (n = 37) received teaching in biology using textbook models. Both groups received two sessions, each of approximately 120 minutes with 15 minutes of instruction about either computational models or biology textbook models.

The paper poses the hypothesis:

H3.1 Students learn threshold concepts within an existing subject better by computational modeling learning activities designed by the CMC framework than when conventional learning activities are applied.

Previous studies have attempted to assess if students learned coding and CT skills in primary education mostly (Brennan & Resnick, 2012; Werner et al., 2012). This paper looks at if students draw on their ability to think computationally when learning another subject, e.g. biotechnology? The research question being:

RQ3.1

To what extent do learning activities around computational modeling foster students' understanding of biology and vice versa?

Implementing coding with computational modeling and content knowledge is new, as elaborated earlier. This study used active control groups and experiment groups, and findings

showed that the experiment groups performed better when measured by ‘Biology Content Inventory’ questions. The paper argues that students use CT concepts when describing a biotechnological phenomenon and furthermore, they perform better in assessments of biotechnological knowledge measured by validated questions from a Biology Content Inventory. Students even use knowledge of biotechnology to understand computational modeling.

Paper 4. Computing and Gestures in High School Biology Education.

Paper 4 addresses the role of gestures produced by students while participating in CMC learning activities regarding CT. Learning is an embodied experience, which has been gaining more attention in education areas recently. Especially in the STEM (science, technology, engineering, mathematics) subjects in both primary, secondary, and tertiary education (Abrahamson et al, 2020; Roth, 2020; Macrine & Fugate, 2020). Given the synergetic effect, described in paper 3, of learning CT and learning biotechnology simultaneously among students in a classroom indicates that perhaps we, as computing education researchers, should embrace alternative signs of learning when it comes to CT. The design was a concurrent mixed-methods study. The study was situated in a constructivist and embodied cognitive perspective. This video-study explored the types of gestures used by students as they engage in learning activities of CT and modeling in Biology. Participants were twenty-eight students (twenty female, eight male) randomly sampled from five different High School Biology classes.

The paper poses the hypothesis:

H4.1 Embodied learning appears when students work with computational modeling of a phenomenon in a high school subject, i.e. Biology.

The traditional methods of measuring CT and programming skills may not be appropriate for measuring learning among all students, as students can have diverse backgrounds and many different learning strategies. Hence, in paper 4 a more embodied approach to measuring learning and to answer the research question was used:

RQ4.1

How are different types of student gestures distributed across core elements of CT?

The area of embodied cognition is emerging in STEM education and in math education in particular (Abrahamson et al., 2020; Flood et al., 2020; Alibali & Nathan, 2012), but the area of gestures in CT education is under-researched (Solomon et al., 2018; 2020). The overarching research objective of this paper was to develop and evaluate a taxonomy table as

a way of classifying and coding gestures versus CT. The paper concludes that the study is the first to propose a taxonomy table of high-school students' use of gestures when engaging in learning activities in computing and that students use gestures adaptively in order to learn core elements of CT.

Paper 5. Empowerment through Computational Thinking: A Literature Survey.

CT potentially contains empowering elements for all learners, and paper 5 takes a broader perspective on CT in education and investigates the field of computational empowerment described in the literature in relation to CT in education. The paper accounts for a literature survey categorizing different strands of empowerment as it unfolds in computing education research concerning CT from 2012 onto and including 2020. A total of 210 papers were included in the final survey.

The paper poses the hypotheses:

H5.1 The community of computing education researchers focus increasingly on empowerment but the community don't have a uniform definition of what empowerment means in relation to CT in education.

H5.2 The use of the term empowerment varies within the computing education research community with respect to geography.

The paper seeks to elicit the landscape of current literature on computing education in relation to the term empowerment. This leads to the research question:

RQ5.1

How does current literature on computing education articulate empowerment in relation to CT?

By applying an existing categorization tool, defining the use of empowerment in relation to five interpretations categorization guidelines were developed. The guidelines served to investigate the landscape of current research on CT in computing education. The paper reports on the understanding of empowerment which varies substantially based on geographical origin meaning that empowerment as an end-goal in CT related education might differ significantly from region to region. Also, the study found that the critical and mainstream empowerment categories were under-represented in the overall international CT literature, but more prevalent in research deriving from the Nordic countries.

1.3 Positioning the work of this PhD study

The work described in this dissertation falls within the scope of computing education research and has taken inspiration from the argument that computing can be perceived as an educational basic skill alongside the educational ‘three Rs’: reading, writing, and arithmetic. Computing is seen as the fourth language following spoken, written, and mathematical languages, and as a useful and fundamental language within all other subjects.

The CMC approach (integrating learning activities in coding, modeling, and content matter in learning activities) presented in paper 1, integrates computational modeling into a STEM and social science context. The rationale is that it is important for students to be able to apply and think critically about computational models to understand their capabilities and limitations. Hence, the CMC approach contributes to students’ computational literacy and empowerment. Paper 2 relate to computational modeling being a key aspect of CT with adjacent competences, skills, and knowledge drawn from computer science. Paper 3 explores the effect of the CMC approach on students’ learning in both computing and biology education. Paper 4 focusses on students’ learning of CT, but takes a new approach to investigate this aspect of CER. By identifying core elements of CT, in a computational modeling context, a taxonomy of CT elements and gestures is presented in the paper. Finally, computational empowerment in computing education is investigated as a central focus by conducting a literature survey in paper 5. The paper identifies a knowledge deficit in computing education research papers concerning the potential of computational empowerment.

The five papers explore the essence of CT and computational modeling in education. The findings and contributions from the papers fall into two main research areas: 1. students’ learning and 2. educational development in a broad perspective. These are summarized in the following (See Table 2).

Together the papers present an answer to the overall research question presented in section 1.1.1 “*What and how do students learn by integrating computational modeling and coding into Biotechnology?*” by proposing a strategy for integrating computing into existing subjects in high school education and presenting both a fruitful practical approach based on a framework and an evaluation template and investigations into the affordances of this approach in relation to both students’ learning and computational competences and empowerment.

Paper	Students’ learning: of CT and subject matter	Educational development: CT and computational modeling in education
-------	----------------------------------------------	---------------------------------------------------------------------

1 <i>Computational Thinking in the Danish High School: Learning Coding, Modeling, and Content Knowledge with NetLogo.</i>	- Students learn both subject matter, in the form of biotechnology, and computational modeling when engaged in learning activities design by the CMC approach.	- The CMC framework represents a fruitful way for teachers to design and teach and for students to tinker with learning CT.
2 <i>A Template for Teaching Computational Modeling: A Design Based Research Study in High School.</i>		- Development of a template to evaluate the quality of teaching activities for students developed by high school teachers in 12 different subjects.
3 <i>Learning Computing and Biology from Computational Modeling: An Intervention Study in High School.</i>	- There is a synergistical effect of learning both subject matter and computational modeling in biotechnology.	
4 <i>Computing and Gestures in High School Biology Education.</i>	- Students use gestures adaptively in order to learn concepts of CT. - Development of a taxonomy table of gestures versus elements of CT.	
5 <i>Empowerment through Computational Thinking: A Literature Survey.</i>		- The understanding of computational empowerment varies substantially based on geographical origin. - Computational empowerment with a critical and mainstream management perspective is under-represented in the CT literature.

Table 2. Findings and contributions, from the five papers included in this dissertation.

2 Background

The following provides an overview of existing literature on CT or computational modeling in relation to education. This section present by no means an exhaustive review of CT education nor of computer science education, but serves merely as a background for positioning the research findings and contributions presented in this dissertation.

The concept of computational modeling is presented as an important element of CT and computer science education is described as a related but separate concept to CT. Computing education research is outlined as the field of research to which CT, computational modeling, and computing are related. Finally, rationale and strategies for computing education are presented as a perspective on the importance of computing education research.

2.1 Computational Thinking

The term computational thinking (CT) was developed by Jeanette Wing after Seymour Papert (1980). In her influential paper from 2006 she argues that CT represents a universally attitude and skill set everyone would be eager to learn and use (Wing, 2006). This paper led to a

virtual boom in research into computing into education at all levels e.g. Computer Science Teachers Association (CSTA, 12) and Standards for Computer Science Educators (ISTE, 2014).

However, while CT has been very influential, related terms have been proposed. The term ‘computational literacy’ has served as a particular inspiration for this PhD study. Computational literacy was described by diSessa (2001; 2018) as key conceptual ideas (e.g. representing cell growth or protein synthesis in programs) and associated activities and learning goals. Computational literacy is an important end goal of computing education and dependent on both the representational form (computer programs) and the affordances and limitations this form provide, and a complex of social and cultural matter within the education system. I, as well as diSessa (2018), have during this work experienced what can be done with computers in education with almost no preparation and how computational models can function to achieve surprising learning.

CT, computing education, and computational literacy are all 21st century skills often mentioned as important for our students (Lee et al., 2020a; diSessa, 2018; Yadav et al., 2016; Wing, 2009). However, there are no clear answers as to how these phenomena relate to how we, as researchers, can provide and reflect upon the pedagogies and tools needed for students to obtain these skills.

In educating students to learn biotechnology, or any other existing subject, concepts, relations and practices within the subject are described as well as learning goals for students’ learning of both content and identity of the subject. This should also be the case in computing education. It is not enough to obtain a casual acquaintance with computers and computational tools. Instead educators have to present concepts, relations, and practices of computing to students and relate them to explicit learning goals. And we, as computing education researchers, have to provide insights into how best to do this, as well as tools for doing so.

When Papert described CT, he referred to the affordances, as implied by Gibson (1979) as something that refers to both the computational representations and the ability to express powerful ideas (Papert, 1980). Papert recognized the far reaching benefits of the skills learned through programming, he argued that “computer presence could contribute to mental processes not only instrumentally but in more essential conceptual ways, influencing how people think even when they are far removed from physical contact with a computer” (Papert, 1980, p. 4).

Wing (2006) has proposed an alternative and more recent definition of CT in which she emphasizes the contribution of computer science to human endeavor and describes CT as

“ involving problem solving, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (Wing, 2006, p. 33). Although a number of definitions for CT have been proposed since Wing’s definition, no consensus has yet emerged (Grover & Pea, 2013; 2018, Denning, 2017; Aho, 2012; Guzdial, 2019a, 2019b). Wing’s initial description of CT was very much aligned with computer science education and the concepts and practices within this field. But in later publications Wing broadened the definition of CT and proposed possibly the most commonly used definition today: “Informally, CT describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines.” (Wing, 2011).

Hence, CT is not a skill primarily relevant to computer scientists or confined to computer science education as nearly every field of research has seen the growth of a computational counterpart in the last few decades. CT is a cognitive skill rooted in the human ability to conceive of meaningful, information-based representations that can be effectively manipulated by a computer and is becoming a cross-disciplinary notion spanning both the humanities (e.g. disciplines around studying computer models as signs, human-computer interaction studying human thinking related to CT, and computational methods such as data mining to discover patterns, links, and trends) and the natural sciences (e.g. computer simulations related to physical processes in nature) as J. Wing points out when she described CT as influencing research in these disciplines (Wing, 2008). There is no need to demarcate these research fields, CT influences all research fields and hence, one might add the social sciences and health to these fields of research. By doing so, it becomes clear that CT is multi-faceted and goes beyond computer science (Li, et al., 2020a; 2020b; Grover and Pea, 2013; Shute et al., 2017). To many scientists, the use of computing is not only a tool, but also a way of making discoveries, involving problem solving, design, and model building, as for example, computational modeling used in biology research (Brodland, 2015). Also, as Denning pointed out, CT has long been used and commonly talked about in many professional fields, such as physics, chemistry, and biology, even without the participation of computer scientists (Denning, 2009). In other words, CT is a natural part of much STEM research, however, the significance of CT in STEM fields have not automatically translated into school instructions (Li et al., 2020a).

Many researchers now see CT as part of an even wider call for ‘computing education’, e.g. in 2017 The Royal Society proposed that CT should be offered to all students in all schools at all levels (The Royal Society, 2017). The concept ‘computing education’ is

gaining momentum and represent much of what can be understood as input to, and outcome of CT (Guzdial, 2015). Amy Ko formulates it like this while describing her own research: “Computing education it not just about programming, or even just about computer science, but also about all the phenomena surrounding computing, including data, modeling, and society’s view on computing. So computing education covers far more than just learning to code.” (Ko, 2020).

Andrea diSessa has been instrumental in combining computational tools and STEM education by introducing the concept of computational literacy in 2001 (diSessa, 2001). Di Sessa emphasizes the affordances and limits of computing in STEM education, and the concept of computational literacy describes the improved ability of humans to understand and solve problems that would otherwise be impossible to solve, with specifically designed technology for education. (diSessa, 2001; 2018; Li et al., 2020a).

The relationship between subject matter and CT is intricate and expand the learning of content matter and the context in which CT can be applied (Wilensky & Reisman, 2006; Wilensky & Stroup, 2002; Weintrop et al., 2016; Grover & Pea, 2018; Burke & Kafai, 2010). This relationship is essential for the motivation to bring CT and subject matter together. Another motivation is to engage a more diverse group of students in computing education. Research activities in computing education increasingly engage the entire K-12 range. However, little is known about validated practices, but the importance of CT have been recognized (Parker et al., 2016; Cooper et al., 2014; Grover & Pea, 2018).

There is a lack of research on students’ learning of content matter and CT simultaneously in high school, i.e. the synergetic effect of bringing these together. Bortz et al. (2019) have shown how argument driven scientific practices can act as a synergetic tool for students’ learning of CT and chemistry ‘in tandem’. The research involved eight grade students, and required chemistry knowledge by the students of a low complexity.

Research of a possible synergetic effect requires integrated assessments of both students development of CT and subject knowledge. However, CT assessments are often focused on specific CT skills such as programming and computing (see section 2.2) , and assessments for high school students are scarce (Tang et al., 2020). Brennan & Resnick (2012) have proposed a practical framework for studying and assessing students development of CT concepts, practices, and perspectives while working with a block-based programming environment. The framework uses both qualitative and quantitative methods for assessing students development of CT. But the framework is based on young children creating projects outside of a formal educational setting and not related to specific subjects or content matter.

Hence, there is a lack of research on investigating the synergetic effect of students learning both CT, computational modeling, and subject matter simultaneously.

Peter Denning (2017) highlights how computing education researchers and educators struggle defining CT. In a critique of the appeal for CT in other fields outside CS, in which the description of CT can sometimes be vague and confusing, Denning argues that questions like: What is CT? How can CT be assessed? Is it good for everyone? are still somewhat unanswered (Denning, 2017). The work in this PhD study aims at providing answers, or pieces of answers, to questions like those suggested by Denning (2017).

A working definition of CT as the concept “thinking with machines” was the starting point for this PhD study. It includes a set of practices used in other contexts than computer science and draws on content, real-world applicability, and authenticity of CT (diSessa, 2018; Weintrop et al., 2016; Blikstein, 2013). The concept includes looking at symbolic interactions including gestures. It borrows from Aho’s description of the concept of CT as becoming a research activity that includes inventing appropriate new models of computation (Aho, 2012). In relation to applicability in and outside of school it echoes claims about CT being a thinking tool for understanding our technology-infused world (Denning & Tedre, 2019). In terms of authenticity Weintrop et al.’s description of CT as grounded in authentic meaningful computational practices essential for students to master have been an inspiration (Weintrop et al., 2016).

2.2 Computer Science Education

There is more to computer science education than computer programming, as Wing (2011) identified. She goes on to identify fundamental concepts of computer science that is included as perspectives on computational thinking in a National Academy’s workshop report (Wing, 2011; NRC10, 2010). In this PhD study, I conceive computer science education as part of a broader education in computing, in which CT is also included, hence the two concepts are closely related.

Computer science education has seen a significant growth in interest worldwide, and has been introduced into education at all levels. Perhaps not surprisingly, there are some unclarity in terminology concerning computer science education. Courses within this area for K-12 students are being called both “Computer Science”, “Informatics” or “Computing” in the US, EU, and in the UK, respectively (Computer Science Education, 2020; Informatics Education, 2021; Computing at School, 2021). The term “Informatics” is perhaps the best

title for the part of computing education overlapping with computer science education in K-12 education.

The positioning of computer science education in schools can be challenging, and an obvious interest has been on programming or coding as a main ingredient in the computer science curricula. Researchers have highlighted the contradictions in describing computer science education as, on one hand, CT supposed to be the essence of computer science, and on the other hand, as the mere action of coding and programming, an extraction of computers science that is represented in computer science education research. Many of the assessments designed for assessing CT competences among students, are mostly addressing students' programming and coding competences which are competences often associated with computer science education (Armoni, 2016; Caspersen et al., 2018a).

But computer science education is about more than teaching programming. It includes understanding the fundamental principles of computer science and at K-12 levels engaging children and students in being creative with technology and developing a computational literacy (Sentance, 2018; Caspersen et al., 2018a; Caspersen, 2021a). From the above perspective on computer science education stems a link to the work of the computing education research community in aspects of expanding and diversifying the population that takes computer science courses and of improving teacher training, teaching methods, and teaching technologies. Finally, although computer science is important in many, almost every, discipline, very little is known about what computer science material to teach and when, in K-12 education, as well as how to teach computer science for K-12. Computing education research addresses these questions too (Ko et al., 2020; Nelson & Ko, 2019).

Informatics for grades 9-12 has been characterized as a scientific practice where five methods are at play: formal, experimental, construction, process and modeling (Caspersen & Nowack, 2013a; 2013b). Building on the fact that modeling is a central method to, and even a tradition in, computer science, several researcher have identified computational modeling as an important part of computing education as a whole (Guzdial et al., 2019; Denning, 2018; Lee et al., 2020b; Grover & Pea, 2013; 2018; Denning & Tedre, 2019; Weintrop et al, 2016).

2.3 Computational Modeling

As described in the introduction (section 1) computational modeling is an important part of CT. Computational modeling is also a central part of all STEM education and essential in representing an authentic version of STEM subjects (Nersessian, 2008; Schwarz & White,

2005; Jacobson et al., 2015; Chandrasekharan & Nersessian, 2015; Gouvea & Passmore, 2017). In biotechnology computational modeling is used as a tool for cutting edge research to gain further insight into phenomena such as evolution, bioinformatics, the spread of viruses, molecular behavior, population growth etc. The integration of computational modeling in biotechnology has resulted in a 'third leg' of biotechnology called 'In silico biology' aligned with 'in vitro' and 'in vivo' biology (Di Ventura et al., 2006). However, little is known of how to integrate computational modeling into the subject of biotechnology and research into new and effective pedagogical and didactical approaches in relation to computational modeling are needed (Li et al., 2020a; Yadav et al., 2014; ISTE, 2014; National Research Council, 2011).

Researchers have pleaded for embedding computational modeling into existing STEM education, including biotechnology, as a way of increasing the interest and appeal of CT and modeling among students, hereby attracting and retaining students in computational sciences (Jona et al., 2014; Sengupta et al., 2018; Shute et al., 2017; Bortz et al., 2019). Computational modeling supports a pedagogical approach of students exploring existing models, but also in enabling students to assess their own modified models within a subject (Brady et al. 2015; Gilbert, 2004; Penner, 2000; Wilensky, 2003; Wilensky and Reisman, 2006; Wilkerson-Jerde et al., 2015). Computational models enable students to investigate problems that would otherwise be too dangerous, difficult, or expensive to carry out (National Research Council, 2011; Weintrop et al., 2016). Nowack & Caspersen (2014) presented a practice for computational modeling in education as a top-down approach for CT and modeling. The researchers defined computational modeling as the activity of building or modifying a model system based on a referent system (Nowack & Caspersen, 2014, pp. 148). They motivated the approach by emphasizing how it benefits from a strong tradition in computing and build bridges to a wide range of other disciplines.

Although the above mentioned research is concerned with how to bring computational modeling into existing subjects in education at primary and secondary levels, there is a lack of research on how to integrate coding with computational modeling in existing subjects, and a lack of research on which elements of computational modeling is relevant for teachers to integrate with coding and subject matter in high school.

2.3.1 A constructionist didactical approach

A constructionist approach to learning that involved students tinkering with the computer models' interface and code have been used in this PhD study (Papert & Harel, 1991;

Vygotsky, 1980). This involved students playfully manipulating the code of a computational model to generate and pursue questions in relation to the model, much as described in Wagh et al. (2017). This is especially important for students who are novices in computing and programming, as are the participants of the studies described in this dissertation (Nowack & Caspersen, 2014). Students were encouraged to use, modify, and create the models and code by giving them specific tasks to perform. Thus, the computer models are what Vygotsky conceived of as a semiotic sign meaning a representation for someone that could become a mediated activity for the student. Hereby the models, when appropriated by the student, results in a regulatory change in the students' behavior, understanding etc. (Vygotsky, 1980).

Computational modeling being an important part of CT, was the primary focus for the learning activities in the studies. The learning activities used computer models to address a domain specific phenomenon starting from outside the modelled phenomenon, looking at a simulation and an interface, and moving inside, looking at a computer program producing the simulation (Nowack & Caspersen, 2014).

2.3.2 Educational psychology and pedagogical approach

The inspiration for the pedagogy and didactic approach in this PhD study comes from important educational psychologists, primarily Jean Piaget, Jerome S. Bruner, and Lev S. Vygotsky. Piaget has contributed with his formal descriptions of the nature of the knowledge children exhibit at each stage of development, are especially important and beneficial contributions for many educational research studies and also for this research project. The learning activities and student assessments designed for the studies in this PhD work are all based on the notion of balance between what students can do through the activities and what the activities can do to the students. This balance is described by Piaget (1972) as a condition for adaptation by which the student becomes capable of acquiring and understanding the relevant subject matter. Bruner describe it as 'patterns of growth', how much of humans perception involves going beyond the information given through reliance on a model of the world of events that makes possible interpolation, extrapolation, and prediction (Bruner, 1966, page 2). In Bruner's early work he talks about how a learner conserves past experience in a model of the world, and what the rules are that govern storage and retrieval of information from this model (Bruner, 1966). He introduced three ways that humans accomplish this, by: action, visual or sensory (perceptual) organization, and in words or language. Hence, by enactive, iconic, and symbolic representations. The learning activities designed for this research all strive to activate these three ways of learning by acquiring the

students to actively explore the computer models, present visual representations of the domain specific phenomena, and by asking the students to verbally and in writing, explain both computational and domain specific phenomena.

The approach of computational modeling, integrated into specific content knowledge by coding, offers an opportunity to elicit students mental models of a phenomenon before they are able to represent the phenomenon symbolically and verbally. The learning activities are based on the developed of the CMC framework, which integrates learning activities in coding, modeling, and content knowledge of a subject, in this case biotechnology.

By offering students the opportunity to act and visualize the phenomenon through the learning activities, the students primitive representations of the phenomenon are elicited (Bruner, 1990). This is true for the two domain specific phenomena ('cell growth' and 'protein synthesis') chosen for this PhD study. And especially true for the computational phenomena ('agents', 'procedures' etc.), since students are expected to have even less symbolic representations for the computational phenomena, being novice programmers. The learning activities designed for these studies should lead to a computational empowerment enabling students to approach computing activities critically in their own education and in their lives in general by activating an ability to symbolically represent their computational wishes and beliefs.

The didactic approach included principles of tinkering (Maloney & Resnick, 2010) and of 'use-modify-create' (Lee et al., 2011). Phases like use, modify, create, and tinkering can represent different phases of students' cognitive and practical activities in CT (Lee et al., 2011). Jacobson et al.'s (2015) description of how the progression from low structured learning activities to learning activity with a high degree of structure is a fruitful didactical approach in relation to students' learning. These principles have been an inspiration for the design of the learning activities used in the two interventions.

The learning activities included tasks in which the students related to both the referent system, e.g. a biotechnological phenomenon, and the computer model system. They also included tasks related to students' concept formation and abstraction processes (classification, aggregation, generalization, finding proper representations) and to students having discussions about the referent system in terms of the model system, especially as part of students' computational modeling process (Nowack & Caspersen, 2014).

2.4 Computing Education Research

Computing education include both CT and computational modeling. The field of ‘computing education research’ is relatively newly established (Daniels & Pears, 2012). A look at the rise of the supporting organizations, movements, and national programs of the field , e.g. the Computer Science Teachers Association (CSTA), FabLabs, and CS4All in the US and Informatics for All in EU, shows that these are also all relatively newly established (CSTA in 2004, Fablabs in 2013, CS4All in 2016. and Informatics for all, 2018) (Blikstein & Moghadam, 2019; Caspersen et al., 2019). As another example, CSEdGrad (CS Education Graduates, 2021), a new initiative exploring the pathways of future computing education researchers and their current needs as a community, have been supported by the NSF, and is currently led by Michigan State University and Purdue University. A survey conducted among the members of the CSEdGrad community in early 2021 received a strong response and revealed an interest among the members in face-to-face gatherings, special interest groups, and access to rich resources. These activities all indicate a need for new researchers in the field to establish a foundation for the further work of the research field.

Computing Education Research seeks to build a theoretical understanding of how students learn computing concepts and processes and draws on both educational psychology and learning sciences. The computing education research community have been described as commonly devising new constructs by use of grounded theory, phenomenography, and various statistical models (Malmi et al., 2019), and with a research focus on improving practices, improving the understanding of practices, and improving the situation in which the practices takes place (Daniels & Pears, 2012).

Computing education researchers might often create educational technologies to support the learning and teaching of computing (Fincher et al., 2019). But computing education research is not explicitly concerned with the broader use of technology in learning, teaching, and education. It's specifically concerned with the learning and teaching of computing in particular. Computing education research is also not just about programming, or even just about computer science education, but also about all of the phenomena surrounding computing, including CT and modeling, computational literacy, data, information, empowerment, ethics, and sociocultural and sociopolitical views of computing in society (Nelson & Ko, 2019). Hence, computing education research is positioned to address questions about how to integrate computing into other disciplines. Furthermore, how

to broaden and retain students in computing education, deliver computing education, and offer computational literacy equally to all (Cooper et al., 2014; 2016). Research related to teachers professional development, evaluation of students' learning using mixed methods, and literature reviews that can identify areas for future research are of particular interest (Fincher et al., 2019). The contributions from the papers of this dissertation is in line with both the research call on students' learning of computing, in particular computational modeling and subject matter, and on teachers professional development by providing a template for developing and evaluating teaching activities. Paper 5 presents a literature survey on how research papers on CT in education articulates empowerment as an end-goal for CT in education. As Weintrop et al. (2020) point out, it is time to begin considering how computing education can have alternative endpoints, as not necessarily an education of future professional software developers, for learners that better reflect the ways computing is impacting their lives.

2.5 Computing in Education

There are several rationales for justifying computing education as proposed by Blikstein & Moghadam (2019). These include, but are not limited to, the CT rationale, the computational literacy rationale, and the empowerment and equity of participation rationale. As described in section 2.1 Wing argued that CT “represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.” (Wing, 2006, p.33). Although most people will probably agree that CT is a desirable skill or competence worth learning, it is an over-ambitious statement. One could ask why e.g. artists, politicians, and blue-workers would want to learn CT. The quote by Wing (2006) reflects a rapidly growing interest in learning to use computing in a broad array of professional activities, particular in fields such as biology, physics, social science, finance, and mathematics (Li et al., 2020a). Naturally, this growing interest is related to the design of educational systems, as not only professional activities, but the increasingly digital nature of our world in general requires that all learners feel empowered to understand and participate in computational activities and have access to computing education. Hence, researchers have focused on the effect of computing education and the alignment of computing education. In an attempt to align computing education, in a statewide setting in the US, a survey on computing education pathways was presented by Guzdia et al. (2012). Results, gathered from students

participating in introductory computing classes, showed that having computing experience from high school may have played a role in students' choice on participating in an introductory computing course in college. Kafura and Tatar (2011) investigated the effect of computing education and found that a course on CT and modeling had deepened and broadened college students' reflection and understanding of computer science. These studies indicate that courses on CT, in high school and college, can spark students' interest in, and understanding of, computing.

Therefore, access to computing education for all students is important. This can be ensured by offering specific courses on CT and computer science, but also by integrating CT into existing subjects. By integrating CT into educational systems, a discussion of how to foster and align students' CT competences, skills, and practices is necessary. But just as importantly, a discussion of what students will gain from achieving these CT competences, skill, and practices, leading to why CT should be introduced into the educational system is also necessary.

2.5.1 Computing education: a two tier strategy

In Europe, several countries have focused on implementing CT into existing school curricula or developing new curricula for CT (Caspersen et al., 2019; Caspersen, 2018c; Heintz et al., 2015, Lamprou & Repenning et al., 2018). Slovakia was one of the first countries in Europe to implement CT into education by introducing 'educational programming' into primary schools in 2008 (Gujberova & Kalas, 2013). England was also among the first countries in Europe to build and implement CT into primary and secondary school systems, and have already harvested the first solid experiences in this field. Since the publication of an evaluation report, aligning computing education across England, was published in 2017 research into how to improve the opportunities for both researchers, developers, teachers and students have begun (Jones, 2013; The Royal Society, 2017).

In Denmark computing education became a major concern not only for educators, but also for curriculum developers and policy makers within the past decade. In 2017, Danmarks Vækstråd (The Danish Council for Growth), an advisory board for the Danish Government since 2006, recommended that CT should be integrated into, and compulsory at, all levels in the Danish educational system. The implementation of a new subject "*Technology Comprehension*" into primary and lower secondary schools in Denmark was developed and is being tested and evaluated at the moment, from 2019 to 2021, as part of an initiative by the

Danish Ministry of Education (Thukala et al., 2019; Caspersen, 2021b). Also, in 2017 a whole chapter of the textbook, used throughout Denmark for educating high school teachers in general pedagogy and didactics, was dedicated to CT in high school education (Caspersen, 2017). Approaches to integrate CT into education also received attention from funding bodies, e.g. The Villum Foundation and Central Denmark Region. A brief was commissioned by The Villum Foundation to cast light on the research area of CT in education internationally and in Denmark particularly (Caspersen et al., 2018a), and The Central Region of Jutland funded one of the first projects to implement CT into high school education in Denmark as part of The Central Jutland Technology Pact (The Central Jutland Technology Pact, 2018; Computational Thinking in Secondary Education, 2018)

This indicates that there is a two tier strategy as to how computing can be implemented into education: ‘CT as a subject’ or ‘CT in existing subjects’. These two strategies have been described in the research literature (Caspersen et al., 2018b; Caspersen et al., 2019), and phrased by Shuchi Grover in “A tale of two CTs” in 2018: ‘CT as a thinking skill for computer science classrooms’ and ‘CT as a thinking skill/problem solving approach in non-CS settings’ (Grover, 2018). But as pointed out by Weintrop et al. (2020) many national efforts tend to focus on economic concerns and concerns related to personal creativity and empowerment. However, computing education programs focus on students learning the skills and practices of professional software developers. Weintrop et al. argue for taking the “For All” (in “CS For All”, <https://www.csforall.org/>) seriously and consider alternative endpoints for computing education by having computing education targeting all students regardless of school, age, and interests (Weintrop et al., 2020). In Denmark the subject in computing education for primary and lower secondary education, named “Technology Comprehension” (Caspersen, 2021b), is perhaps one of the only and newest attempts to follow this advice. While computing curricula of many countries essentially have taken university curricula and adapted them for secondary and even primary school levels, the curriculum of the Danish subject stands out in the sense that the approach is much thorough, broader and considers the social aspects of computing, hereby rethinking the needs of the pupil population to infuse general digital competences for all pupils (Kölling, 2021; Guzdial, 2021). While the work in this PhD study is aligned with and welcomes the effort of Weintrop et al. (2020), there seems to be a need for further research into the empowerment aspect of both the national efforts and of the computing education programs (Iversen et al., 2018). Computational empowerment has been suggested as a lever for building educational programs that include both the approach to engage students in developing personal

meaningful applications that have an impact on the real world, and the approach to have students create impactful technology (Tissenbaum et al.; 2017).

However, more research is needed into how CT should be integrated into existing education, especially in primary and secondary education (Guzdial, 2015; Guzdial et al, 2019; Lee et al., 2020a). And in particular, relatively few research studies have been conducted towards creating frameworks and guidelines for integrating and assessing CT into existing high school education (Grover & Pea, 2013; Tang et al., 2020).

In the future, computing education research will strengthen the field of computing. Understanding how students think about computing and how computing activities can improve their learning will have impacts on how to design the interface between humans and computers. Teachers will have access to validated and established pedagogical instruments and assessment tools. The training of K-12 teachers will follow established guidelines, principles, and methods. Departments with faculty interested in computing education research or interested in hiring in this area, will realize that a number of models for successful appointments and collaborations exist (Voogt et al., 2015; Guzdial, 2015).

In summary, the concepts of CT and computing education have gained increasing interest in the research literature within this century, as a quick search of the ACM digital library database supports. The number of research papers containing ‘CT’ in the title increased nearly tenfold from 4 in year 2000 to 35 papers in 2020, as well as the number of papers including ‘computing education’ in the title tripled from 32 in year 2000 to 102 papers in 2020. In the same time interval the number of papers including ‘computer’ in the title doubled from 489 to 1127 papers. However, there are still under-researched areas of CT and computing education. The PhD study focusses especially on questions of how to introduce coding and computational modeling, as part of existing education in high school, and what students learn by participating in such teaching activities integrating coding, modeling and content matter. Given that these questions are answered they give rise to the question of what aspects of computational empowerment can students obtain from learning computational modeling and coding.

3 Method

The overall research question for this PhD study was” What and how do students learn by integrating computational modeling and coding into Biotechnology?”. To answer this question the research project made use of both quantitative and qualitative methods to

research aspects of computing education such as students' learning and teachers assimilation of computing into existing subjects, which is relevant for both students, teachers, and high schools.

3.1 Research design

The research project used a concurrent mixed methods design. It was longitudinal and consisted of iterative cycles of design, enactment, analysis, and redesign of sharable design strategies (Design-Based Research Collective, 2003). The design-based research project consisted of five studies. The first took place in late 2018, the second during 2020 and 2021. The third study was initiated in 2019 and ran into 2020. The fourth study took place during autumn 2019 and spring 2020, and the fifth study was initiated in 2020 and finished in 2021 (see Table 3).

The didactical framework, CMC, was developed in the first study (study I) and applied in the next three studies (study II, III, IV). The CMC framework integrates coding activities (C) with computational modeling (M) and specific content matter (C) related to the phenomenon represented in the computer model and to an existing subject.

In study I, six modules consisting of learning activities and computer models in participation with three high school teachers were designed. The phenomena represented in the computer models were taken from the subjects of biotechnology, chemistry, biology, and social science. Each of the six modules centered around one specific phenomenon in biotechnology, chemistry, biology, and social science respectively. Social science represented by three different modules and phenomena. The six modules were tested by 15 high school teachers from nine different high schools in their classroom, without researchers and developers being present. 210 students participated in the learning activities. All students answered an online questionnaire containing domain specific questions and identical questions throughout the six modules regarding CT and modeling.

For the second study, two nearly identical professional development courses were designed and offered to 86 randomly chosen in-service high school teachers from 44 high schools and representing 12 different subjects. The two courses differed on one parameter. One course included agent-based modeling in NetLogo the other course did not. Otherwise the two courses provided the participants with comparable training and lasted the same amount of time.

For study III and IV, a two-module curriculum was designed including both content, modeling, and coding activities and centered around cell growth and protein synthesis. The curriculum was progressing in complexity. For the third study the design was a two-group intervention study using mixed methods and comparison and intervention groups. The comparison group in the third study was given traditional biology teaching mainly consisting of blackboard teaching focusing on the biology models presented in the traditional textbook (Egebo et al., 2017). In contrast, the intervention group was exposed to the same biological concepts in the same amount of time as the comparison group but used computer models designed for the interventions. This was done both through similar learning activities to those in the comparison group and via interaction with a computational model developed in NetLogo (Wilensky, 1999). Students in the intervention group were offered a 120-minute intervention on cell growth, including 15 minutes of direct instruction about relevant biology and computing. Data were collected before, during and after the interventions, as well as two weeks after the intervention. Subsequently, after the third study, all students were pooled into one group. This group participated in the fourth study focused on the phenomenon of protein synthesis using a separate computational model and biological curriculum. The same 118 students from four different high schools participated in study III and IV.

For study V 210 papers were analyzed and rated by independent researchers for the use of the term empowerment.

The activities of the PhD study took place as visualized in the timeline below (Table 3).

Timeline	Autumn '18	Spring '19	Autumn '19	Spring '20	Autumn '20	Spring '21	Autumn '21
Study	I	III	III, IV	III, IV, V	II, V	II, V	V
Study: Research objectives	Study I: Integrate CT and computational modeling into existing subjects in high school education	Study III: Assessing students' learning of computational modeling and subject matter	Study IV: Assessing students' embodied learning of CT + study IV	Study V: Survey CT and empowerment in computing education literature + study III, V	Study II: Developing in-service high school teachers' understanding of CT and computational modeling + study V	Study II + study V	Study V

Table 3: Timeline, studies, and research objectives.

The PhD study presents a first step towards a generic approach to integrating CT into subjects, in which the specific subject is subordinate. Therefore, the research design took its onset in four different high school subjects represented in study I by 15 teachers and 210 students, to investigate if the framework was fruitful for different subjects. The research design then increased the number of teachers participating in study II compared to study I, from 15 to 86 teachers, and the number of represented subject from four to 12 respectively, in order to investigate if the development of a template for evaluating teaching activities on computational modeling was applicable to high school teachers and subjects in general. Having established a fruitful framework and an applicable template for designing and evaluating teaching activities on CT and computational modeling, the research design was designed to focus on investigating how and what students learn by participating in learning activities on computational modeling in study III. The number of students were reduced compared to study I and included 118 students for study III. Of the 118 students participating in study III, 28 students were selected for study IV. Study IV represented a new approach to investigating students' learning of CT and used videorecording for analyzing students' gestures. Hence this study used a smaller number of students compared to study I and III, as each student produced a large number of data for analysis. Finally, the research design changed from intervention studies to a literature survey for study V. The study included 210 papers from the computing education research community on empowerment in relation to CT in education. This study was conducted as a narrative review in order to capture the overall tendencies in how empowerment is articulated in recent research literature.

Although the specific subject can be almost all subjects present in the Danish high schools, for the third and fourth study described in this dissertation the particular subject chosen was biotechnology. Biotechnology is a subject introduced into the Danish high school education approximately a decade ago in 2007 (Egebo et al., 2017), and only offered as an A-level subject. Biotechnology combines biology and chemistry with an emphasis on authentic techniques and methods used in the field. The author holds a master degree in biology, a supplement in chemistry and biotechnology and knowledge of computer science from courses in "Introduction to programming"¹, "Fundamentals of NetLogo"², "ICER2019 Doctoral Consortium"³ and "Research in Computing Education"⁴. The author does not have an academic background in the learning sciences, but has over ten years of experience as a high school teacher in biotechnology, biology, and chemistry, and have played the leading role in introducing biotechnology as a new subject into the largest high school in Denmark, thus, the

subject of Biotechnology was a natural choice for this PhD study. For the purpose of paper 1 through 5, the term biotechnology has been replaced by the term biology. This is in order to frame the work described in the papers so that a computing education researcher from any geographical area can relate to the designation of the subject. The term Biotechnology is only being used as a term describing a subject in high schools ('gymnasium') in Germany and Denmark to the knowledge of the author.

1 <https://kursuskatalog.au.dk/da/course/91950/Introduktion-til-programmering> (visited 15.10.2021)

2 <https://www.complexityexplorer.org/courses/84-fundamentals-of-netlogo> (visited 15.10.2021)

3 <https://archive.icer.acm.org/icer-2019/doctoral-consortium/> (visited 15.10.2021)

4 <https://www.ntnu.edu/excited/call-for-participants-excited-summer-school-2019> (visited 15.10.2021)

3.1.1 Participants

The first study included 210 students and 15 teachers. the students and teachers came from nine different high schools throughout central Denmark. The students attended either biotechnology, chemistry, biology, or social science and were either first, second or third year students in high school.

The participants in the second study were 86 in-service high school teachers, 26 female and 60 male, from 44 different high schools throughout Denmark. The teachers taught 12 different subjects including both the natural sciences, the social sciences, and the humanities. Less than 10% of all participating teachers identified themselves as having any experience with programming before participating in the study.

The third and fourth study included 118 students, 70 % female and 30 % male. The students came from five different biotechnology classes in four different high schools distributed in Jutland, Denmark. The students' age range was between 16 and 19 years of age. More than 63% of the students identified them self as novice and only 3% as having comprehensive experience with programming. All students were taught by the same textbook (Egebo et al., 2017). All students had chosen Biotechnology A as their main subject throughout the three years of their high school education. Apart from the main subject the students chose a smaller subject as an elective during the three years. Their choices, as reported by them self, were very diverse, ranging from music to astronomy, representing 16 elective subjects in total. Five teachers participated in the interventions. One for each class of students. Four teachers were female and one male. The teachers age ranged from 38 to 61 years of age. All teachers had more than 10 years of experience in teaching Biology and Biotechnology. Four school participated in the two studies. The schools had different

socioeconomic range, but all were within the range of 90% of all high schools in Denmark (Secondary Educations, 2020). The high schools were located in three different regions of Jutland, Denmark. The total number of students attending each school ranged from approximately 700 to 1500 students.

The fifth study was a literature survey by coding 210 papers. Four researchers participated in conducting the survey and analyzing the results.

Table 4 presents an overview of the interventions and the participants.

Study	Domain specific theme	Number of students or teachers	Design	Computer models	Instruments
I	Biotechnology: Cell membrane Chemistry: Equilibrium Biology: Heart Social Science: Intervention, welfare, economic circuit	210 students	Design-based. Mixed methods. Quantitative and qualitative data.	Six models	Questionnaires Log files Semi-structured interviews
II	12 different high school subjects	86 teachers	Design-based. Mixed methods. Quantitative and qualitative data.	51 models (see: https://library.ct-denmark.org/)	Questionnaire Teacher produced learning activities
III	Biotechnology: Cell growth	N=81 students in intervention group N= 37 students in comparison group	Intervention study. Mixed methods. Quantitative and qualitative data.	Model 1(see appendix F)	Concept maps Log files BCI Questionnaire
IV	Biotechnology: Protein synthesis: transcription and translation	N=118 students in intervention group	Intervention study. Mixed methods. Quantitative and qualitative data.	Model 2a Model 2b (see appendix G and H)	ScreenCast-O-Matic Video recording
V	Papers concerned with CT in an educational context and empowerment	210 papers	Extracting sentences and paragraphs containing the term “*mpower” Coding the sentences according to an existing taxonomy.		Python script for extracting sentences. Existing taxonomy of empowerment

Table 4. Overview of the studies and participants included in this study.

3.1.2 Learning objectives in CT and Biotechnology

According to Denning (2017) only vague definitions of CT have been produced so far. But learning objectives have been suggested (Computing at School, 2021; Denning, 2017), although they are often abstractly described by terms such as: algorithms, programming, data, hardware, communication, and technology. In this PhD study the learning objectives in CT were inspired by early descriptions of CT competences by Brennan & Resnick (2012) and Wing (2011) such as model understanding, understanding the relation between subject and model, and between representation and code and being able to modify code. The overall goal was for students to understand how the design of a computational model relates to a biotechnological phenomenon and to understand the concept of “stepwise improvement” as a general approach to incremental development of computer models Caspersen & Kölling, 2009; Caspersen & Nowack, 2013c). This includes students understanding program syntax and realizing how a program can be decomposed into smaller parts. It also meant finding patterns in the small parts, realizing that small parts of the program can be reused in other programs and, by this, designing and adjusting algorithmic descriptions.

The learning objectives focused on students’ knowledge of the concept of variables and values, statements, loops, if-sentences, and program structure. More specifically, the students should learn to navigate, use, and modify a NetLogo program including changing values of variables, modifying a procedure, introducing new breeds of agents, and adding procedures by copying and modifying existing procedures. In order to perform the latter, students’ also need to understand the overall structure of a NetLogo program, e.g. understand when to add breeds of agents. Hence, students should be able to understand how biotechnological concepts and sub-phenomena are represented in a NetLogo program (see Figure 2). Figure 2 presents the learning objectives of CT that have been the foundation for designing learning activities in this work. The figure illustrates a progression from students being able to use and interact with a NetLogo model, over the ability to change small elements of code to modify the model for a specific purpose, to introducing new elements and even behaviors to the model by adding new procedures.

Learning goals in CT
Interact with a NetLogo model
Change value of a variable
Modify a procedure

Introducing new breeds of agents
Adding procedures

Figure 2. Learning objectives for Computational Thinking (CT).

Protein synthesis was represented in the two studies each centered on a sub-phenomenon, namely cell growth and protein production. The phenomena were connected by an overarching story about the Danish company Novo Nordisk and the company's production of the protein Insulin for medical purposes by genetical engineered yeast cells and an overall driving question of how to increase the production of Insulin.

In study III and IV the learning objectives for biotechnology were related to the phenomenon of 'The central dogma' which is a concept describing the flow of genetic information within a biological system (See Figure 3). Figure 3 illustrates how genetic information is stored in DNA molecules and can be transcribed to RNA molecules used for producing protein by translating the RNA molecule to protein. Firstly, the focus was on cell growth and on the industrial production of the protein Insulin, using genetically engineered yeast cell as an example. Figure 4 (taken from students' textbook) is a traditional way of illustrating the growth of cells to students in textbooks. The plot in figure 4 is reproduced in computer model 1 used for study III, as an output of running the simulation. This can be seen by comparing figure 3 with the interface of model 1 (Appendix F).

Secondly, the focus of study III and IV was on the molecular production of Insulin and specifically of the subprocesses protein synthesis: transcription and translation (see Figure 5 and 6, also taken from students' textbook). Sub-concepts such as growth factors, gene regulation, and protein structure were addresses in student tasks as questions during the learning activities.

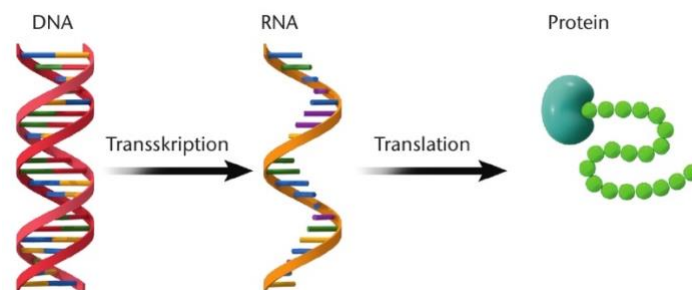


Figure 3. Model of 'the central dogma' from students' textbook (Egebo et al. 2017).

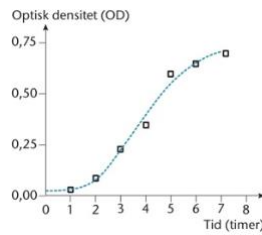


Figure 4. Model of cell growth from students' textbook (Egebo et al., 2017).

Figure 5 and 6 illustrates the concepts of transcription of DNA molecules to RNA molecules (Figure 5) and of translation of RNA molecules to protein (Figure 6). The representation of the DNA molecules as two strands of nucleotides, illustrated by the letters A, T, G, C, in figure 5 is repeated in model 2a, which can be seen by comparing figure 5 with the interface of model 2a (Appendix G). Likewise, the representation of RNA molecules as a one strand molecule of nucleotides in figure 5 can also be seen in model 2a. Figure 6, representing the concept of translation, illustrates the process of producing protein from RNA molecules. The RNA molecule is again illustrated by a single strand of nucleotides, which is also the representation chosen for model 2b, which can be seen by comparing figure 6 with the interface of model 2b (Appendix H). The protein being produced is illustrated by green 'pearls' connected as a string in figure 6. The same way of representing the protein molecules can be seen in model 2b, in which the protein molecule is built of green 'pearls'. The illustrations in figure 5 and 6 also contains elements, named 'RNA-polymerase' and 'ribosom', that are also included in model 2a and 2b.

The representations for model 1, model 2a, and model 2b, of the concepts of cell growth, transcription, and translation were chosen to help students make connections to their own prior content knowledge of these concepts and hence, help students transfer knowledge from the subject domain of biotechnology to the domain of the computer model.

Students often exhibit misconceptions in learning central biotechnological generic concepts of randomness and variation, and feedback mechanisms and the genetic code (Klymkowsky et al., 2010; Anderson & McKenzie, 2010; Duit, 2009; D'Avanzo, 2008; Garvin-Doxas & Klymkowsky, 2008; Klymkowsky & Garvin-Doxas, 2008). These concepts were represented in the code of the computer models (1, 2a, 2b) and addressed in specific tasks for the students to solve during the learning activities. Hence, common misconceptions in biotechnology were taken into consideration when designing computer models and learning activities. An example was asking students to describe "the life of a yeast cell" from

their understanding of the code in model 1. This fostered an understanding among students of the development of a yeast cell as something influenced by randomness and variation.

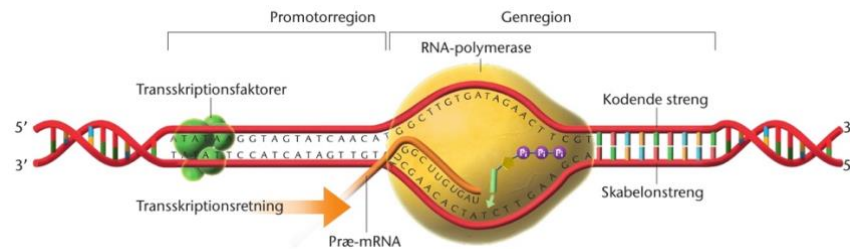


Figure 5. Model of transcription of DNA to mRNA from students' textbook (Egebo et al., 2017).

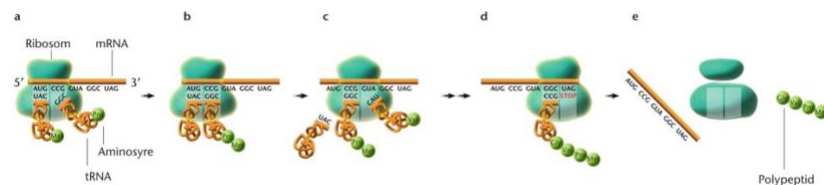


Figure 6. Model of translation of mRNA to protein from students' textbook (Egebo et al., 2017).

3.1.3 Learning activities

The design of the learning activities for study I, II, III, and IV followed the CMC approach (described in Paper 1) with an emphasis on the computational modeling parameters as described in Paper 2. See section 2.3.2 “Educational psychology and pedagogical approach” for a detailed description of the CMC approach.

The learning activities included three phases: a tinkering, an explorative, and a discussion phase (see section 2.3.1 ” A constructionist didactical approach ”) . The latter focusing on students’ perception of using computational models for learning Biotechnology. The activities took a constructionistic approach, taking their onset in a tinkering phase in which students examined, used, and tinkered with a predesigned computational model representing parts of the phenomenon of cell growth or protein synthesis. The tinkering phase was replaced by an explorative phase in which the students identified groups of agents, their appearance and behavior and tested the models for their range of application. Subsequently, students were asked to modify the models by changing the code. Activities in this phase ranged from changing values of a variable, to tasks such as creating code by creating a new

procedure for the models and evaluating models. During the learning activities the students were also asked to answer validated questions from the Biology Concept Inventory (BCI), open-ended questions regarding the representation of the phenomenon in the model (both in the code and in the interface) and verbally explain specific procedures in the code and how the procedures relates to the biotechnological phenomena.

In all four studies, students used personal individual lab top computers for the computational modeling exercises and the comparison exercises.

In study I the teachers lead the session after having received a short two hours introduction to the designed activities and models beforehand.

In study II teachers produced their own teaching activities after having received an introduction to the CMC framework.

For study III and IV the author of this dissertation led the interventions with both the intervention groups and the comparison group in collaboration with the relevant classroom teacher. Every student was asked to create a concept map of cell growth or protein synthesis at the beginning of each intervention. Log files from students in both studies were kept on the students' computers till the end of the interventions and then collected by the author. Open-ended questions, asking students to describe the phenomenon represented in the model and the procedures in the code, were posed to all students of both studies. In study III students worked in groups to create video recordings of maximum of two minutes using the software ScreenCast-O-Matic (Priowijanto, 2013) of themselves explaining specific pieces of code. After instruction, all students filled out an online questionnaire measuring subject knowledge, in biotechnology, by use of questions from the Biology Content Inventory (Klymkowsky & Garvin-Doxas, 2008). At the end of each intervention students created concept maps once again. The concept map exercise was repeated 9-15 days after the interventions.

3.1.3.1 Comparison Group

The comparison groups in study III received an introduction to the biotechnological phenomenon by the author and thereafter performed tasks focused on models of the phenomenon as it was represented in the textbook. The learning activities for the students in the comparison groups were designed to align with the activities for the students in the intervention groups but using the traditional textbook, instead of the computer models. The learning activities of the comparison groups had a duration time similar to the duration of the learning activities of the intervention group.

3.1.3.2 Intervention Group

For the four studies I, II, III, and IV learning activities focused on the computing concepts of variables, values, and conditionals, with a gradual progression into introducing whole procedures and program structure. Domain specific phenomena were represented in a computer model visualizing both a simulation of the specific phenomenon and the code (See section 3.4 for a detailed description of the models used in study III and IV). The students received an introduction to the domain specific phenomenon by the teacher (in study I and II) and by the author (in study III and IV), but no explicit instruction introduced the computational environment. Instead learning was thought to arise from semi-structured interaction with the system. The first task in the learning activities acquired students to explore and tinker with the interface and describe, in as much detail as possible, the phenomenon that was represented by the simulation in the interface. Students were then asked to modify the characteristics of specific agents by introducing changes to the code and asked to argue why they chose to make the specific changes. During the computational interventions in study II and III, students were also asked to comment on specific procedures in the model system and finally asked to identify a major improvement to the model and create a new procedure in the program to accommodate this improvement (in study III).

3.2 Agent-based modeling in education

An agent-based modeling approach was used to design interventions for integrating CT into existing subjects in high school. Agent-based computational modeling uses individual computational entities with appearance and behavior described by simple rules in relation to the domain specific phenomenon they describe and enact. When the code in an agent-based computer model is executed, a simulation, in which the modelled biological phenomenon emerge, is visualized.

The concept behind ABM is the observation of simultaneous and multiple interactions of agents or programmed instructed entities through simulated environments. In developing these simulated environments, predictions of emergent phenomena can be evaluated. With an ABM, macro or high-level system properties can emerge from lower-level system interactions. Moreover, large scale state changes will emerge from micro-scale agent behaviors generated from system-level interactions (Wilensky & Rand, 2015).

There is a significant potential for learning by integrating agent-based modeling in science education. While science education relies heavily on the complex language of mathematics to describe science phenomena, the language of programming simplifies and eliminates much of the complexity when describing the same phenomena. By using appropriate tools in education it becomes possible to focus on a new approach and method in all scientific domains where computing offers its own specific way of thinking to describe and explain scientific phenomena (Caspersen et al., 2018b).

The vision of using ABM in education stems from Papert's seminal book 'Mindstorms', in which he emphasized that the essence of the computer was its universality that gave power to the computational thinker to simulate real-world phenomena with the computer (Papert, 1980). He developed an educational programming language called Logo and a physical device, a 'turtle', to be programmed by it. Papert applied 'Constructionism' as a design and learning theory building on the ideas of Jean Piaget's learning theory 'Constructivism'. However, constructionism is focused not only on learning processes, which is the main focus in constructionism, but also on the design processes leading to computational products as something meaningful for each student to show and share among fellow students. Hence, within constructionism the development of adequate computational learning tools for students to use, modify, create, and construct is important (Wilensky & Papert, 2010).

Papert's two doctoral students, Mitchel Resnick and Uri Wilensky carried this vision further by developing agent-based programming environments for educational use in both primary and secondary education, i.e. Scratch and NetLogo, respectively (Resnick, 1992; Resnick, 1996a; Resnick, 1996b; Resnick, 1996c), (Wilensky & Resnick, 1999; Wilensky, 2006; Wilensky & Rand, 2015; Weintrop et al, 2016). Also, Alex Repenning have developed agent-based programming environments for primary education, i.e. AgentSheet and AgentCubes (Repenning, 2000; Ioannidou et al., 2009). These environments have mainly been used to introduce programming through a game-based approach (Basawapatna et al., 2010) but are also now being used as an introduction to computational modeling in courses for teachers. In Switzerland, the future teacher workforce in primary schools are now being introduced to AgentCubes as part of a national curriculum for education (Lamprou & Repenning, 2018).

Other researchers associated with Wilensky and the development of NetLogo are Abrahamson and Blikstein (Blikstein et al., 2005). While Abrahamson have pursued the embodied learning approach and designed computational tools for children to physical

interact with while learning subject matter, such as the Mathematics Imagery Trainer (Abrahamson & Sánchez-García, 2016), Blikstein is currently working on enabling primary and lower secondary students to modify computer models by providing domain specific code blocks and integrating data science elements in the NetLogo environment DeltaTick (Fernandez et al., 2021). Finally, Weintrop, who also contributed to the development of NetLogo, have researched how a definition of CT can be developed from the perspective of practitioners in STEM (Weintrop et al., 2016). Recently Weintrop have argued for a broader approach to integrating CT into educational systems by considering various endpoints of computing education (Weintrop et al., 2020). The work of the above mentioned researchers are important as background and beacons for the research described in this PhD study. Especially, the embodied approach to cognition by Abrahamson has been an inspiration for study IV. And for study I-III, the definition of CT presented by Weintrop et al. (2016) with an emphasis on computational models and simulations has inspired our choices of programming environment and the development of the CMC framework. Blikstein and Weintrops approach to coding activities has served as a background for developing teaching activities in study I-IV, and Weintrop et al.'s description of alternative endpoint for computing education as inspiration for study V.

3.3 NetLogo

The work described in this dissertation is based on the use of NetLogo as an agent-based programming environment. NetLogo, developed by Wilensky, is an open-source multi-agent programming and modeling environment for simulating complex phenomena, developed with multiple audiences in mind (Wilensky, 1999). NetLogo is optimized for the use, modification, and creation of agents by providing a programming language easily understandable for novice programmers. The programming philosophy of NetLogo: “low-threshold, high-ceiling” is inspired by Logo, a programming language designed earlier by S. Papert, who was the doctoral supervisor of U. Wilensky (Blikstein et al., 2005). NetLogo uses four types of agents: turtles, patches, links, and the observer. For the models designed in study II and III, only the first three are used, and primarily the first two. Turtles are agents that can move around in the ‘world’. The ‘world’ is a two dimensional square divided into a grid of patches. Patches are small squares of ground, over which turtles can move. Patches cannot move around, but otherwise have the same properties as turtles. Links are agents that connects two turtles. When using NetLogo models the user can give

instructions to hundreds or thousands of independent “agents” all operating concurrently, and agents can be collected into ‘agent sets’ (Tisue & Wilensky, 2004).

In NetLogo the main window is divided into tabs and you have the choice of viewing a model’s interface, code or info tab. The interface tab is where you watch the model run. You start by pressing ‘setup’ which create a ‘world’ in which agents (turtles, patches, and maybe links) appear. By pressing ‘go’ the model start to run. Usually there are additional plots and sliders accompanying the ‘world’ in the interface view. The code tab is where the code for the model is stored. You can write commands in the code and save them for use later. Hence, next time you press ‘setup’ and ‘go’ in the interface, the model will use the commands you have written. The info tab provides an introduction to the model. It explains what phenomenon is being modeled and how to use the model. It also suggests things to explore in the model and special NetLogo features used in the model

(<https://ccl.northwestern.edu/netlogo/docs/>). Blikstein demonstrated in 2010 (Blikstein, 2010) how computational models build in NetLogo can facilitate learning science in middle school education by integrating theory and computer models with physical models. Fuhrman et al. in 2013 investigated if students should design and, or interact with computer models in science education. Fuhrman et al. define a model as “an abstract, simplified representation of a phenomenon that focuses on one of the phenomenon’s key elements and may be used in the production of an explanation or a prediction of that phenomenon ” (Fuhrman et al., 2013; p.483) and report that students gain metamodeling knowledge if they are engaged in designing the computational models as well as interacting with them (Fuhrman et al., 2013). Fuhrman et al. also concluded that “When students perceive a model as imperfect, they tend to more carefully evaluate the model and better address its limitations, leading to deeper learning about the phenomenon itself.” (Fuhrman et al., 2013; p.486). Other researchers have used agent-based computational modeling and have shown that it improves students’ knowledge acquisition of STEM subjects (Centola et al. 2013; Gautam et al. 2020; Sengupta and Farris, 2012; Wilensky & Reisman, 2006; Bortz et al., 2019).

The studies described in this dissertation explores the relationship between learning subject knowledge and computational modeling, i.e., students’ learning with computer models within the domain of biotechnology. In order to learn computational modeling, students should develop in a progression from basic to advanced skills of computing. This is a progression from basic computing skills such as the ability to read and understand program syntax, to more advanced competences such as modify and comment on the design and program-structure of a computational model. The work focuses on instantiations of

abstractions that are important to understanding subject knowledge, especially biotechnological knowledge, including how students learn to choose between elements that represent structural similarity between the computing and biotechnological concepts.

3.4 Computer models

Three models for study III and IV were designed and developed during the PhD study. The approach was a model-based thinking and practice approach (Nowack & Caspersen, 2014). The models were designed and developed as a collaboration between a professional programmer (Palle Nowack) specializing in computational modeling and the author of this dissertation. The process was an iterative process including cycles of design, enactment, analysis, and redesign of the models.

One model (model 2 used in study IV) contained two sub-models. They were designed to be run simultaneously by NetLogo Level Space, an extension for NetLogo that allows modelers to easily build multilevel agent-based models (Hjorth et al., 2020). Hence, model 2a and 2b are connected and can run simultaneously, with model 2a producing agents (mRNA molecules) that are dynamically transferred on runtime into model 2b and appear in the “world” of model 2b’s interface as they are produced in model 2a. the phenomenon of protein synthesis is visualized as the sub-phenomenon of transcription, in model 2a, in which molecules of mRNA is being produced. The same molecules are then transferred to model 2b and appear as part of the sub-phenomenon of translation in model 2b.

The computer models used in the studies were designed primarily to serve as initial starting points and idea generators for students to tinker with, use and modify. Students could interact with models built as half-baked microworlds. A microworld is defined as a less developed representation of a phenomenon (Kynigos, 2007; Wagh et al., 2017).

Model 1 for instance, represents the phenomenon of “cell growth” (see Figure 7). The phenomenon was presented for the students in an introduction, given by the researcher, focusing on the production of the protein Insulin by yeast cells. The theory of cell growth and growth curves (see Figure 3) was highlighted in the introduction to the students. The same phenomenon was represented in the computer model (model 1). The students tinkered, used, and modified the model by relating it to their content knowledge of cell growth and cell curves. Especially, students modified the code by relating the existing code in the procedure ‘lav-en-ny-turtle’ (‘make-a-new-turtle’) to their knowledge of specific properties of yeast cells, such as reproduction time. Hence, there was a close correlation between the parameters

of the biotechnological phenomenon and the computer model, i.e. between a referent system, the subject domain of cell growth, and the computer system as described in Caspersen & Nowack (2013c). Figure 7 illustrates this correlation between the domain specific system and the model system, using model 1 as an example. The interface of model 1 can be seen in Appendix F and figure 5 can be used for comparison.

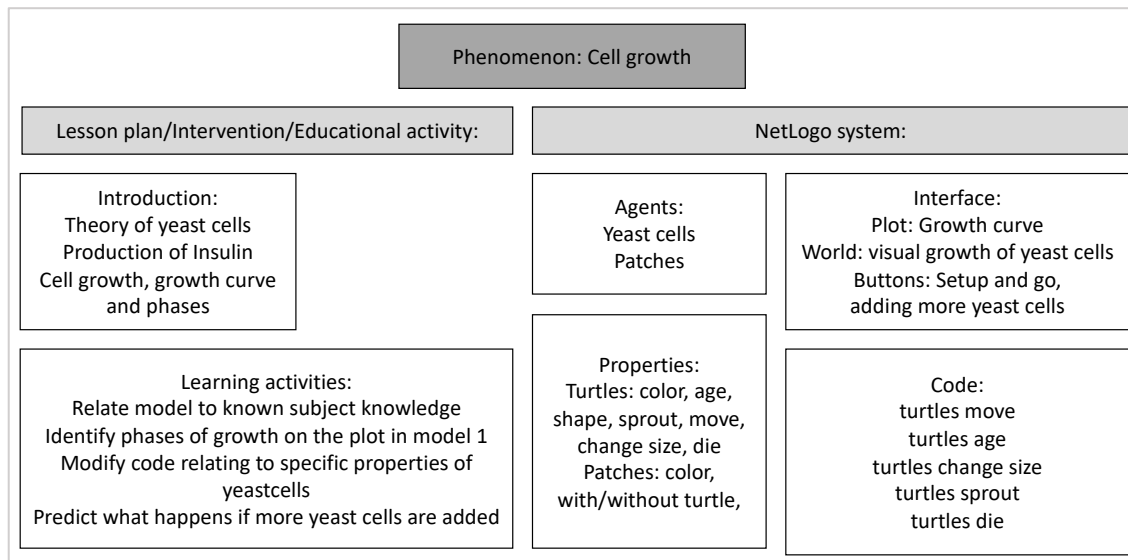


Figure 7. Correlation between biotechnological concepts and computer model 1.

The models developed for study III and IV are purposely simplified in order to motivate the students to modify the models, as also Fuhrman et al. (2013) have argued for. The principle of designing simplified models apply to both the interface and code of the models. See the illustrations from students' textbook (Figure 3, 4, 5, 6) for a comparison to the simplified computer models designed for study III and IV and appended in Appendix F, G, and H. This principle of simplicity, together with the evaluation template described in paper 2, have guided the development of the computer models for study III and IV. The modeling parameters included in the evaluation template are ideally all included in learning activities of computational modeling developed for integration into existing subjects in high school education. Parameter number one to eleven concerns the interface of the models, i.e. the referent system, and number 12 to 16 concern the code of the models, i.e. the model system. The parameters can be seen in Table 5. As can be seen in Table 5, the parameters concerning the model system correlates with the learning objectives for CT described in section 3.1.2. The parameters one to eleven are developed by inspiration of work done on

characterizing the modeling process in science education by Schwartz & White (2005) and Nersessian et al. (2006) and Gouvea and Passmore's (2017) conception of modeling in the classroom and the questions of 'what are models for' versus 'what are models of'.

Number	Modeling parameter
1	The range of the model (limitations and opportunities)
2	The explanatory power of the model
3	Causality between elements in the model
4	The behavior of the model
5	The elements of the model
6	Elements that are 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 phenomena developing over time
11	Use of and interaction with the model
12	Change a value or variable
13	Add a variable
14	Change a procedure
15	Add a procedure
16	Create a new model

Table 5. Computational modeling parameters (from paper 2).

As described in "Research design" (section 3.1), there was a progression of complexity in both the biotechnological content and the code of the models used in study III and IV. Selected features of the computer models designed for the studies can be seen in Table 6. While the code for model 1 contained only one type of turtles, model 2 contained several. Regarding procedures in the code, the 'go' procedure in model 1 called one other procedure, while 'go' in model 2 called several procedures. The number of program lines also increases from model 1 to 2a through to model 2b. The interface of the models all contained some elements that were preserved throughout the models, such as 'setup' and 'go forever' buttons. Model 1 and 2 contained one extra button of either 'go once' or 'add more yeast cells'. Model 1 and 2 contained plots in the interface. The number of plots were increased from one to two in model 1 and 2a and 2b, respectively. The plots visualized and produced data as an outcome of running the models, and quantified the phenomenon

represented by the model. Overall, this increases the complexity of the model and also of the potential investigations that students can do using the models without examining the code. Important features of the models used in study III and IV can be seen in Table 6.

Model	Properties	Number of agenttypes	Number of procedures called by the 'go' procedure	Number of program lines	Interface elements	Button types
1		1 turtle 1 patch	1	36	1 plot 3 buttons "world"	Setup Go forever Add more yeast cells
2a		3 turtles 2 patches 1 link	3	239	2 plots 3 buttons "world"	Setup Go once Go forever
2b		5 turtles 2 patches 1 link	3	256	2 plots 3 buttons "world"	Setup Go once Go forever

Table 6. Features of the computer models number 1, 2a, 2b.

All computer models for study III and IV (Model 1, 2a, 2b) can be seen in Appendix F, G, and H.

3.5 Instruments

During the work presented in this dissertation many instruments for investigating students' learning and teachers' teaching activities have been used. The following sections describe the instruments and how they have been applied in the relevant studies.

3.5.1 Concept maps

Novak and coworkers developed concept maps as part of research into children's changes in knowledge of science (Novak & Musonda, 1991). Concept maps have been used in science education, and in Biology education in particular. Tripto et al. used concept maps to assess students' system knowledge and development of such in relation to a new Biology curriculum emphasizing the role of homeostasis, a complex biological system. Results proved concept maps useful in assessing students' biological knowledge especially at the lower

levels of system thinking (Tripto et al., 2013). But relatively few studies report on using concept maps for assessing students' computing knowledge (Wei & Yue, 2017). Some studies report on the use of concept maps for assessing student knowledge of specific computer science concepts such as input/output topic in computer architecture (Larrazza-Mendiluze & Garay-Vitoria, 2012) or in teaching object-oriented modeling (Sien, 2011). Two studies have identified students' misconceptions and misunderstandings in learning database and information systems concepts (Moen, 2009; Wei & Yue, 2016). Concept maps are graphical tools for organizing and representing relationships between concepts indicated by a connecting line linking two concepts. Lines and words specify the relationship between two concepts. Concepts and propositions are usually organized hierarchically, from most general, most inclusive to most specific. It is best to construct concept maps with reference to some domain of knowledge that is familiar to the person constructing the map, and to define the area of knowledge to be mapped (Novak & Cañas, 2007).

In study II and III, the biotechnological phenomena (cell growth and protein synthesis) were the references for constructing concept maps. Concept map is an easy to learn tool and available online in several editions. For the two studies the independent software company's Bubbl.us online tool was used (<https://bubbl.us/about>).

The concept maps produced by students gave an insight to students' structure of knowledge and their quality of understanding, when analyzed by applying a scoring system developed by Markham et al. (1994) and recommended by Sentance et al. (2018) Examples of concept maps produced by students in study II and III can be seen in Appendix I.

3.5.2 Log files

Log file analysis can shed light upon students' ability to modify computer programs and solve problems computationally (Ericson et al., 2019; Zoanetti & Griffin, 2017). The modifications can be simple (change the value of a variable) or complex (create a new procedure for the program).

The interventions of study I, III, and IV all used collection of log files as a mean to assess students' computing skills and competences. The initial analysis of students' log files excluded all programs that were not executable. Log files from students in study I, III, and IV were analyzed to identify the modifications the students had made to the computer models. The log files were analyzed according to the description of CT concepts developed by Brennan & Resnick (2012). It involves a progression running from changing a value of a

variable, to introducing a new variable, modifying a procedure to commenting on a selected piece of code, creating a new agent, and identifying, copying, and adjusting procedures from one program to another.

3.5.3 Biology Concept Inventory

The work by Hestenes et al. (1992) in developing the Force Concept Inventory, a testing instrument designed to measure student comprehension of the Newtonian concept of force, sparked the development of a Biology Concept Inventory (BCI) by Klymkowsky & Garvin-Doxas (2008). The goal of concept inventories is to identify student misconceptions, and track students' gains in understanding over time in order to lead students to conceptual mastery.

In study III and IV multiple-choice questions from the BCI were translated into Danish and posed to students during the interventions. To establish if students were comfortable with answering BCI questions and if their answers were as expected compared to answers from other students reported in the literature (Klymkowsky et al. 2010), three questions from the BCI were posed to all students in the end of the first intervention in study III and compared with result reported in the literature. Hereafter, students' answers to BCI questions in the intervention and comparison groups in study III and IV were analyzed and compared. BCI questions were used in study III and IV for assessing students' domain specific knowledge, i.e. knowledge of biotechnological phenomena.

3.5.4 Open-ended questions

Open-ended questions were posed to all students during learning activities in all studies by online questionnaires. The open-ended questions were used to prompt students to describe the computer models or the textbook models in relation to the relevant domain specific phenomena.

Examples of open-ended questions posed to students:

- *Try to press "setup" and "go". Describe in your own words what the model represent.*
- *Look at the code. Which names are being used for the elements in the model?*
- *What rules are described in the code for how each element should behave?*

The open-ended questions were analyzed for students' use of domain specific concepts, both biotechnological and computing concepts.

3.5.5 Screen Casts

Screencasts are most often used as an engaging and student-centered activity produced by teachers for their students (Morris & Chikwa, 2013). However, in study IV students were asked to work in groups of three to create screencasts of their computer screens while showing relevant code in the computer model. Students were asked to explain specific pieces of code (maximum of two minutes) using the software ScreenCast-O-Matic (Priowijanto, 2013) recording both computer screen and audio. Hence, the screencast software was used to record and document students' learning of computational modeling.

The recordings were coded for biological and computing concepts, by the author, and the number of times biotechnological and computing concepts were used by the students to explain the code were noted.

3.5.6 Videorecording

Video recordings of students, in study III and IV producing gestures while working in pairs were performed. A gesture is the visible movements of the hands and arms, specifically the preparation, stroke, and retraction phase, which students used while talking (Kendon, 2004). Gestures are important for students and teachers in communication, meaning making, and learning. Students' gestures have been shown to be positively associated with students' learning, and especially students' gestures in relation to math education have been researched (Shapiro & Stolz, 2019; Walkington et al., 2019; Cook et al., 2008; Alibali & Nathan, 2012). However, students' gestures in relation to computing education integrated in STEM education in a high school setting have not yet been investigated (Sheu & Chen, 2014).

Different types of student gestures, derived from video data of interaction between students and teacher who model protein synthesis in the second intervention, were coded and analyzed. Students' gestures were coded in terms of David McNeill's classification of four types gestures: deictic, iconic, metaphoric and beats (McNeill, 1992). Students' CT were classified according to five CT concepts introduced by Grover and Pea in 2018: logical thinking, algorithmic thinking, pattern recognition, abstraction, evaluation and analytical thinking (Grover & Pea, 2018), and compared to the gestures the students produced at the same time.

3.5.7 Teaching activities

Two types of courses for in-service high school teachers on CT and computational modeling were developed by the author in collaboration with researchers in teachers professional development (Keld Nielsen) and software development (Palle Nowack) and conducted as part of study II. The teachers participating in the courses worked in groups of two or three to develop teaching activities on computational modeling for use in their own teaching. The teaching activities were collected by the author and analyzed using a template for evaluating teaching activities. The template was developed by the author and other researchers on the basis of modeling practices in STEM education described by Schwarz and White (2005) and Nersessian et al. (2006; 2008) and on descriptions of computational modeling by Nowack & Caspersen (2014).

4 Findings

This section summarizes the main findings of the five studies reported in the five papers included in this dissertation.

4.1 Learning Coding, Modeling, and Content Knowledge - Study I

This study reports on the development of the CMC framework integrating coding and modeling activities with content matter by use of computer models. a initial teaching experiment where Researchers, developers, and high school teachers collaborated to develop computer models using the programming environment NetLogo. The models were then applied in high school teaching by 15 teachers representing four different subjects.

By applying the CMC framework, it was possible to take a first step towards filling the research gap described in section 2.3 of how to integrate coding and computational modeling into traditional high school education. Both the students and teachers participating in the study were programming novices. The framework presented an opportunity for the students and teachers to gain knowledge of the process of modeling and hands-on experiences with computer programming.

Findings falls into two main themes:

1. Findings showed that students were able to use, modify, and create code in NetLogo that enabled them to develop CT and content knowledge. Figure 8 illustrates how many percentage of students were able to perform tasks relating to five categories computational modeling by gaining understanding: of subject matter, of a representation in the form of a computer model, of the relation between the subject matter and the model representation, of the relation between the interface and the code, and of modifying the code of a computer model. Finally, students' perspective on the role of computational modeling as a learning activity in existing high school education was assessed pre and post the activities and an improvement was seen, illustrated by the 'CT Baseline' in figure 8.
2. The CMC framework represents a fruitful way for teachers to design and teach for students to tinker with learning CT.

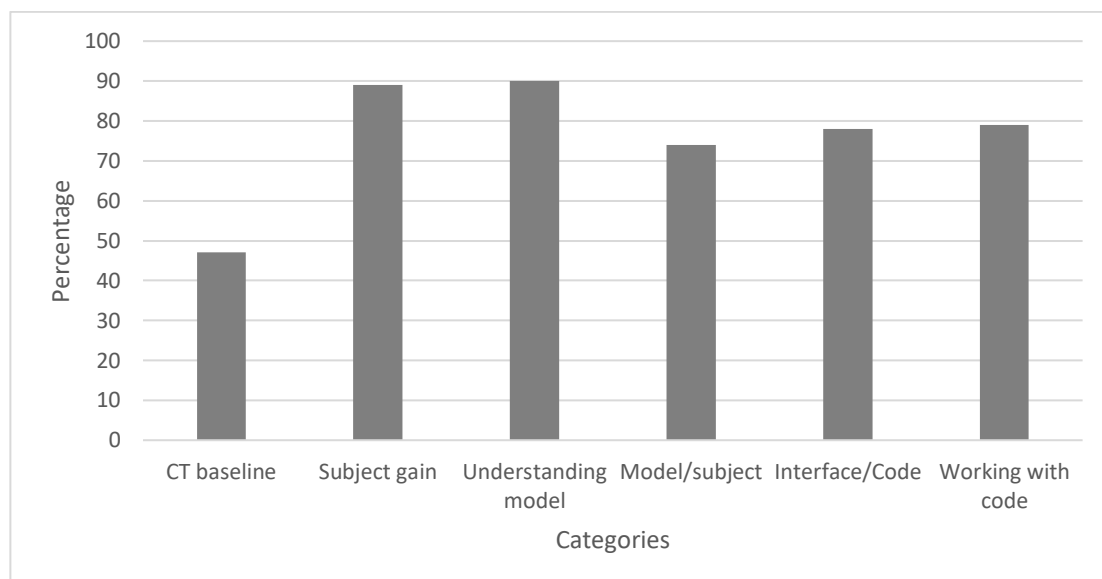


Figure 8: Percentage of students that successfully solved tasks in category.

The study provides details of the underlying CMC framework, which integrates: Coding, Modeling, and Content, and for how to adopt the computer models.

4.2 A Template for Teaching Computational Modeling - Study II

To develop a sustainable professional development of high school teachers education in CT and computational modeling it was necessary to design courses for teachers in how to

develop high quality teaching materials in computational modeling. The study sought to develop an evaluation template as an instrument for evaluating teaching activities produced by high school teachers and as an initial step toward developing teacher education on computational modeling. The template itself is a first iteration on presenting a set of guidelines for developing teaching materials in computational modeling.

The template contributes to filling the research gap described in section 2.3 on which elements of computational modeling is relevant for teachers to integrate with coding and subject matter in high school. The study uses teaching activities produced during two professional development courses for in-service high school teachers. In one course the participants were introduced to programming in NetLogo, in the other course participants were not. The study provides details for others to adopt the evaluation template. The study was performed parallel to the other studies described in this dissertation.

Two major findings are presented in the study:

1. The template (see Figure 9) represent a novel and fruitful way for predicting the efficacy of teacher-designed activities for learning computational modeling and for students to tinker with learning CT and computational modeling. Figure 9 present 16 parameters, each relating to either a domain specific system of e.g. biotechnology (parameter one to eleven) or to a computer model system of e.g. a NetLogo model (parameter 12 -16). By subdividing the template into two systems, the process of computational modeling, as a process of relating a domain specific phenomenon to a representation by a computer model, is visualized in the template.
2. By applying the evaluation template on teaching activities developed by teachers, it was shown that the template represent a novel and fruitful way for predicting the efficacy of teacher-designed activities for learning computational modeling and for students to tinker with learning CT. Figure 10 shows findings from evaluating teaching activities developed by teachers participating in two types of courses, named Prog+ or Prog-. Prog+ participants were given an introduction to agent-based programming, while Prog- participants were not. Otherwise the courses were identical. Figure 10 illustrates how many percentages of all teaching activities, contained questions addressing each of the parameters in the evaluation template. Teachers participating in course Prog+ developed teaching activities in the form of questions representing significantly more parameters than participants in the Prog-course.

- Domain specific system**
1. Range of the model (limitations and opportunities)
 2. Explanatory power of the model
 3. Causality between elements in the model
 4. Behavior of the model
 5. Elements of the model
 6. Elements that are 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 phenomena developing over time
 11. Use of and interaction with the model
- Computer model system**
12. Change a value or variable
 13. Add a variable
 14. Change a procedure
 15. Add a procedure
 16. Create a new model

Figure 9: Evaluation parameters included in the evaluation instrument.

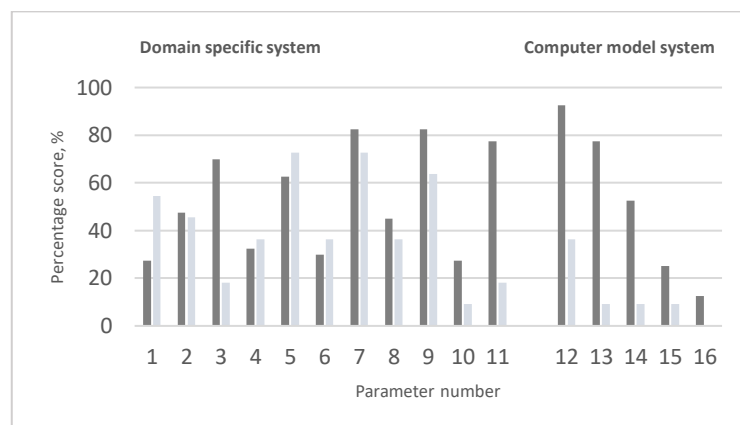


Figure 10: Percent of teaching activities, containing questions relating to each parameter of the template, in Prog+ (dark) and in Prog- (light). See Figure 9 for a description of each parameter.

4.3 Computational Modeling in High School Biology - Study III and IV

After having developed a framework and an instrument to address the part of the overarching research question concerning how students can learn by participating in activities on computational modeling, a natural research focus was to investigate what students learn by participating in learning activities designed by the CMC framework. Assessing students learning is a complex task. To ensure a thorough investigation, various methods, both qualitative and quantitative were used in study III and IV. Study III sought to fill part of the

research gap described in section 2.1 concerning the possible synergetic effect of integrating computational modeling and coding with subject matter. The synergetic effect found, in study III, of integrating computational modeling into biotechnology is an initial stepping stone for investigating the possible transfer of computational modeling skills and competences to other high school subjects, and eventually arguing for computing education for all students independent of students' various educational interests.

The main purpose of the two intervention-control study (study III) was to investigate how skills in computational modeling are associated with knowledge acquisition in biotechnology. The study used intervention and comparison groups giving each group an initial short introduction to computer models or textbook models, respectively. Students performed online learning activities designed by the CMC framework, described in the first study, and by the evaluation parameters developed in the second study.

Findings showed that participants in the intervention group gained statistically significant improvements in both their computational modeling skills and biotechnology knowledge. The findings fall into two themes:

1. An increase in students' skills and understanding of computing concepts. Figure 11 capture the effectiveness of the learning activities, specifically in relation to the CT learning objectives described in section 3.1.2. The figure illustrates how many of students' logfiles were: executable, modified, and commented. 75% or more of all participating students were able to accommodate the CT learning objectives of interacting with, changing, and modifying a computer model. This finding confirms that the students achieved the learning goals for CT and computational modeling, and that they were not too complex.
2. An increase in students' understanding of biotechnology was found when comparing the intervention and comparison group. Figure 12 shows the percentage of students answering correctly on three validated Biology Concept Inventory (BCI) questions. The percentage of students answering correctly from the intervention group, having participated in the learning activities on computational modeling, was significantly higher than the percentage of students answering correctly from a comparison group. This finding strongly indicate an effect of integrating computational modeling into high school biotechnology education as it will foster students' learning of biotechnology subject matter.
- 3.

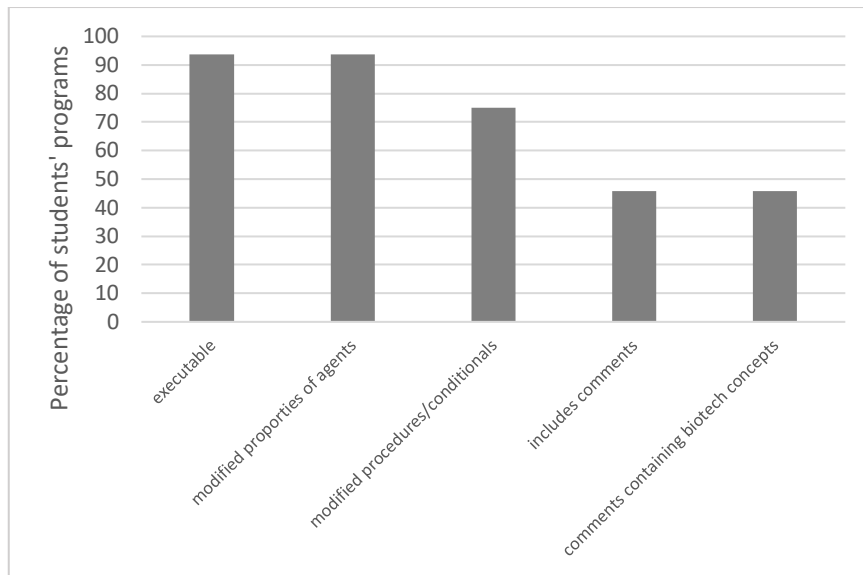


Figure 11: Percentage of students' computer programs containing a specific modification (horizontal axis).

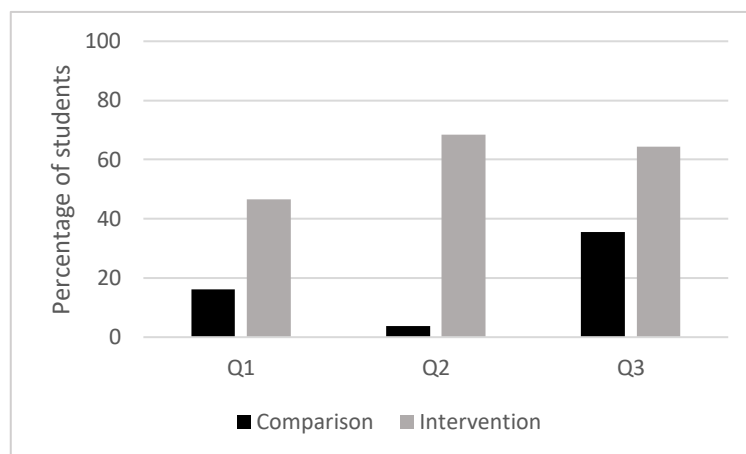


Figure 12: Percentage of students from a comparison and intervention group answering correctly on three BCI questions (Q1, Q2, Q3).

Study III also identify concepts that are learned better from each of the subjects (biotechnology and computing) and suggests that biotechnology and computational modeling should be integrated in high school education.

4.4 Computing and Gestures in High School Biology Education - Study IV

This video-study explored the types of gestures used by students as they engage in such learning activities of CT and modeling in biotechnology. Hence, this study IV sought to fill another part of the research gap described in section 2.1, by investigating how students learned CT and expressed their understanding of CT through gestures, while participating in learning activities designed by the CMC framework. The objective of study IV was to develop a taxonomy table of gestures versus elements of CT.

The awareness of embodied cognition in computing education is dawning, and study IV is unique in its presentation of a taxonomy on CT versus gestures, and on the investigation of students' adaptive use of gestures when learning CT. The taxonomy is a first step towards linking CT concepts with gestures as a means of better understanding student's engagement with computing. These findings are a first step toward a higher awareness of computing education being an embodied experience for students, and toward alternative ways for assessing students CT.

Two major findings are presented in the study:

1. A taxonomy of elements of CT versus gestures, as a first step towards linking CT concepts with gestures as a means of better understanding student's engagement with computing as gestures are indicative of embodied learning (Table 7). As the findings shows, gestures accompany students' efforts to perform computational modeling and thinking, and gestures can be operationalized and coded.
2. A significant difference in the distribution of students' gestures across five concepts of CT were found. Figure 13 illustrates how students use more deictic gestures when engaged in logical and algorithmic thinking and more metaphoric gestures when abstracting on and evaluating the computer models, indicating how gestures represent embodied cognitive strategies. Hence, students used gestures adaptively in order to learn concepts of CT, and in particularly these initial findings suggest a difference students' use of deictic and metaphoric gestures according to the CT concept they are engaging with. This taxonomy table is a way to help educators interpret student gestures in connection with CT.

<i>Gesture type</i>	<i>Deictic</i>	<i>Iconic</i>	<i>Metaphoric</i>	<i>Beat</i>	<i>Total</i>
<i>CT concept</i>					
<i>Logical thinking</i>	17 (18%)	5 (5%)	1 (1%)	0	23 (25%)

<i>Algorithmic thinking</i>	12 (13%)	4 (4%)	3 (3%)	2 (2%)	21 (23%)
<i>Pattern recognition</i>	10 (11%)	2 (2%)	5(5%)	0	17 (18%)
<i>Abstraction</i>	4 (4%)	4 (4%)	12 (13%)	0	20 (22%)
<i>Evaluation and analytical thinking</i>	2 (2%)	2 (2%)	6 (6%)	2 (2%)	12 (13%)
<i>Total</i>	45 (48%)	17 (18%)	27 (29%)	4 (4%)	93 (100%)

Table 7: Gesture type versus CT concepts.

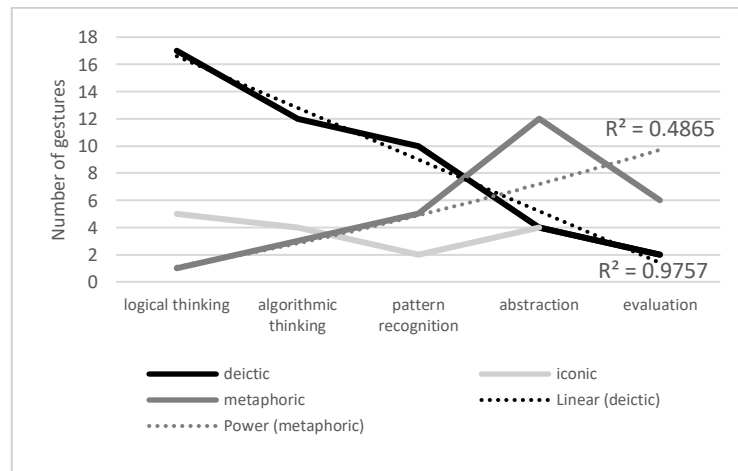


Figure 13: Number of gestures in CT concept groups.

4.5 Empowerment through Computational Thinking - Study V

Concurrent with the four studies described above, a fifth study, a literature survey categorizing different strands of empowerment as it unfolds in computing education research concerning CT was performed. This study represents research necessary for moving forward in investigating the impact of integrating CT and computational modeling into existing education. Empowerment is often described as an end-goal of interventions involving CT in education. However, as the study identify, there is a lack of definition of the term empowerment, and hence, of how CT can be empowering and to whom. In order to be able to discuss the impact of research in computing education and argue for a possible ‘computing for all students’ approach in educational systems, it is important to be apparent and distinct in the mediation of the relevant research results from the computing education research community to fellow researchers, developer, and decisionmakers within educational policy. Study V represent a first step toward obtaining this accuracy and distinctness.

A comprehensive search yielded a total of 244 papers, of which 210 were coded by two independent researchers for the use of the term empowerment in relation to CT in an educational setting. The papers' use of 'empowerment' were categorized, by coding each sentence in which 'empowerment' occurred, according to five categories: mainstream & management, educational, functional, critical, and democratic empowerment. The study discusses the possible challenges of using an ill-defined empowerment term to justify or promote computing education on the basis of the findings.

The findings illustrate two main themes:

1. A lack of a definition of the term empowerment in the included literature. When the included papers were examined for the use of empowerment, only one paper was found to provide a definition of 'empowerment'.
2. Differences in the geographical distribution was found, when comparing North America, Europe, and the Nordic countries in particular. Figure 14 illustrates the distribution of the five categories of empowerment used in coding the surveyed papers. The functional aspects of empowerment is strong in North American research on CT in education, as is also the educational aspect of empowerment. However, when the research papers from the Nordic countries are separated, the educational aspect of empowerment is not as strong in the Nordic countries as in the rest of Europe and in North America. However, the democratic and mainstream/management aspects of empowerment are most strong in the included research literature from the Nordic countries compared to research reported from the rest of Europe and North America. These findings indicate a tendency for different focuses on empowerment and on end-goals of CT in education in the Nordic countries compared to the research reported from North America and other countries in Europe.

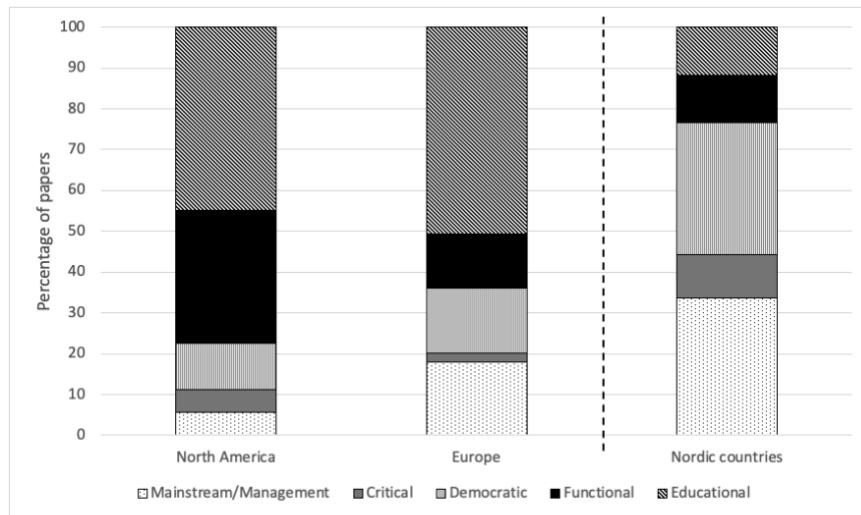


Figure 14: Percentage of papers representing each category of empowerment in the North American, European, and Nordic regions.

4.6 Overall findings

All the findings presented above are partial answers to the overall research question of this PhD study “What and how do students learn by integrating computational modeling and coding into Biotechnology?”. Together the findings indicate trajectories for how to integrate computational modeling and coding into existing subjects in high school education, exemplified by the subject of biotechnology, and for what the affordances of such an integration might be.

Detailed descriptions of the findings, described above, are presented in each corresponding paper. Additional findings and more detailed presentations can be found in the appended papers 1-5 (Appendix A to E).

5 Discussion

In this section a discussion of the findings from the five interrelated studies of this PhD study will be presented. The contributions of new knowledge in relation to the research gaps described in section 2 as a result of the findings and their limitations will be described.

The main findings of the PhD study falls into two categories. First, study I, II, and III support the hypothesis that the CMC approach present a unique approach to integrating computing education into existing subjects in high school education which combine

computational modeling with coding and subject matter. The approach is successful for both students and teachers. The CMC approach fosters students' learning of subject matter and computing. The CMC approach also fosters high school teachers in designing learning activities in computational modeling, and even computer models, given the right introduction to the programming environment, as shown in study II. Second, study IV illustrates that coding and computing education is not merely an 'one student, one program' activity. Students communicate their thoughts to others and themselves by gesturing. As computing education researchers we need to be more aware of how embodied learning and understanding of computing is taken place.

The findings presented in section 4.1, concerning how to integrate computational modeling and coding with existing subjects in high school, contributes to answer part of the overall research question: *What and how do students learn by integrating computational modeling and coding into Biotechnology?* The findings relate to how students learn when coding and computational modeling is integrated with subject matter.

The findings presented in section 4.2 uses the knowledge from section 4.1 and paper 1, about students' learning and teachers ability to teach computational modeling, to develop a template for teachers to foster students' learning of computational modeling. Hereby, further contributing to answer the 'how' of the overall research question by focusing on teachers role in applying the framework and developing teaching materials for computational modeling. Together, the findings from study 1 and 2 contributes to the research gap described in section 2.3 of how to integrate computational modeling and subject matter with coding.

The integration of computational modeling and subject matter have been investigated before by e.g. Wilensky and Reisman (2006) in biology and Sengupta et al. (2013) in physics and biology. However, research is lacking on how to integrate coding with computational modeling and subject matter as described in section 2.3. Wilensky and Reisman (2006) introduced code-snippets of text based programming to students in biology. Brady and colleagues (2015) investigated how working with an agent-based environment can improve students' model-based learning. The environment used in the specific study was a redesign of NetLogo and used what the authors called 'construction primitives' and not text based program code. Sengupta et al. (2013) presented findings from a study integrating programming activities into middle school biology curriculum. The researchers used alternatives to text based programming environments, i.e. a visual programming environments in which students construct programs using graphical objects as drag and drop elements of the interface. The researchers argued that since previous work by Sherin et al.

(1993) have shown that students in high school struggle to construct programs in the LOGO language, even after weeks of instructions, a visual programming environment would be more appropriate for the study reported by Sengupta et al. (2013). The findings presented in paper 1, 3, and 4 contributes to the research gap of research into new and effective pedagogical and didactical approaches in relation to integrating coding and computational modeling with subject matter in education (Li et al., 2020a), as described in section 2.3. Study I, III, and IV, included in this dissertation, build on the work of Wilensky and Reisman (2006), Brady et al. (2015), and Sengupta et al. (2013) and represent a further step towards researching how the integration of coding activities can influence students' learning of computational modeling and subject matter simultaneously.

A further development of the concept and elements of computational modeling is presented in paper 2 as a template. By applying the template it was found useful for evaluating teaching activities developed by high school teachers. Hence, these findings contributes with new knowledge of how students can learn by integrating computational modeling and coding into existing subjects. Especially two aspects are important research contributions: first, how complex the learning activities in coding and computational modeling can be in order to be successfully integrated into existing subjects in secondary education, and second, a template for teachers to develop teaching materials for computational modeling. Although the findings from study I and II are important contributions, further research into the trajectories of teaching high school students coding and computational modeling is needed.

The findings presented in section 4.3 and 4.4 from study III and IV, regarding students' learning of computational modeling, coding, and biotechnology, are products of different research methods. Especially in order to investigate the synergetic effect of learning both subject matter and computational modeling, it was necessary to apply numerous integrated assessment instruments for obtaining the findings described in paper 3 and 4.

The contributions of the two studies seek to fill the research gap, described in section 2.1, of research lacking on high school students' learning of content matter and CT simultaneously, i.e. the synergetic effect of bringing these together in learning activities and assessing students' knowledge of CT and subject matter. Brennan and Resnick (2012) have proposed assessments of students' knowledge of CT concepts, practices, and perspectives. However the study was not concerned with students' learning of subject matter and focused on younger children in a non-formal setting, compared to the studies described in this dissertation. Bortz et al. (2019) investigated eight-grade middle school students' CT and

chemistry knowledge drawing mostly on qualitative assessments designed specifically for the reported study. Hence, study III and IV, described in this dissertation, build on the work by Brennan and Resnick (2012) and Bortz et al. (2019) by presenting findings that establish computational modeling as an embodied activity, and as a mean to learn both coding and domain specific subject matter investigated by established methods in a formal high school setting of biotechnology education. The findings have contributed new knowledge to the research gap of assessing what students learn by integrating computational modeling, coding, and subject knowledge in learning activities in high school education.

Findings, presented in section 4.5 from study V, are a continuation of research into the importance of CT in education (Parker et al., 2016; Cooper et al., 2014; Grover & Pea, 2018). The findings, described in paper 5, points to a lack of definitions of computational empowerment in research literature, as well as an emphasis on the functional and educational aspects of empowerment in North America compared to the Nordic countries. The aspect of managerial empowerment is emphasized the most in literature originating from the Nordic countries indicating a potential focus on how teachers, schools, and policy makers can integrate CT into curricula. However, there is a research gap on how, and by what aspects, CT can empower students, teachers, and school. Paper 5 provides a categorization tool for reflecting on how our investigations, as computing education researchers, can empower in various ways. In particular for this dissertation, how the findings presented in paper 1 to 4 can scale and be sustainable for students, educators, and policy makers.

Although elements of CT is a natural part of most research these days, particularly in the natural sciences, but increasingly also in social science and the humanities, a discussion of what implications a potential computational empowerment of students, in the form of CT, can have for our society as a whole, and for our educational systems in particular, have not automatically been translated into school instructions and validated practices (Li et al., 2020a). However, there are signs of an increasing convergence toward a common understanding of the importance of students' and teachers' computing competences (McGarr et al., 2021)

Although, there have been significant progress in our understanding of how to develop effective teaching environments for students in education, our understanding of how to develop effective methods and approaches to integrate computing into curricula is less developed. The work by Anders Malthe-Sørensen and colleagues at Center for Computing in Science Education at University of Oslo, Norway is concerned with how the integration of computing into existing subjects can present a practice perspective of learning and open for

creative and collaborative learning strategies in which students address real-world problems relevant for their future studies and research (Odden & Malthé-Sørensen, 2021). Hence, integrating computing including computational modeling into curricula can lead to students' computational literacy, and serve as a driver for disciplinary renewal. However, effective methods and approaches to ensure an effective fostering of students' computational literacy and empowerment are scarce.

This PhD study contributes to the gap of research on how to integrate computational modeling and coding into subjects (Li et al., 2020a; Yadav et al., 2014; ISTE, 2014; National Research Council, 2011) by offering new knowledge of effective pedagogical and didactical approaches and of students' learning in relation to computational modeling and thinking in education.

The main strengths of these studies were the design of computer models and learning activities, applied in relatively short interventions, that provided students with computing experiences including coding activities. The learning activities and computer models represented a progressing in complexity that was successfully tested with students. Another strength was the application of mixed methods to measure how students' learning of subject matter statistically significant improved. The working definition of CT as "thinking with machines" (see section 2.1) proved useful as a starting point for all five studies. (see section 2.1) and could be considered a candidate for defining CT in education in multiple settings and maybe even multiple educational levels.

5.1 Limitations

The studies presented in this dissertation of course have their limitations; the major ones are summarized here.

It was not possible to investigate the effect of including programming in teachers professional development (PD) courses, as it was not possible to select the participants in the two PD courses and hence insure a unified distribution of teacher in the two courses.

The number and diversity of participants in the studies is also a limitation. Especially in study IV the number of students participating in the study ($n=28$) is small and the would be needed to warrant more valid inferences about generalizability. Study II, IV, and V make use of trained independent raters. However, internal validity would be higher with more than two raters as is the case with all three studies.

Also, for this PhD study the focus has been on computational modeling in biotechnology education, but modeling is just one aspect of CT and computing education (Fincher et al. 2019; Grover and Pea, 2018; Weintrop et al. 2016). However, computational modeling is considered an important and central element of computing education (Aho, 2012; Guzdial et al. 2019), particularly appropriate for biotechnology and an element that has proven fruitful when integrated into existing subjects (Gautam et al. 2020; Hutchins et al., 2020).

Finally, the presence of a researcher, and in study IV a camera, in the classroom might have made some students behave different than they normally would. Subsequently the internal validity of the study decreased.

5.2 Further Research

The five papers and their contributions to computing education research presented here, form trajectories that runs within the research area shown in Figure 1. Although the contributions of the papers are necessary and important for further investigations, they represent but a small portion of the necessary research for future work. The additional research questions proposed in section 1.1.3 presents possible pathways for such further research.

First, study IV touched upon the question regarding the progression of *what* can be taught to students *when* in regard to existing subjects (see section 1.1.2. RQ(A)) by interviewing teachers at the end of the study. Teachers reported students being more engaged in initiating and participating in discussions about the computer models and the relevant phenomenon at higher taxonomical levels measured by both the Bloom and SOLO taxonomy (Bloom et al., 1984; Biggs & Collis, 2014), than were the case when using a traditional textbook (Egebo et al., 2017). These observations are backed up by Caballero & Hjorth-Jensen (2018) who argue that computing in existing education shifts the focus to problem definition, visualization and “what if” discussions as well as allows for solving more realistic problems and enhancing the understanding of abstractions and generalizations. Sand et al. (2018) also points out how computation can provide fertile ground for student engaging in sensemaking and critical thinking. The findings call for further research into e.g. learning trajectories of CT. Work on CT trajectories is already initiated by Diane Franklin and her research group (Rich et al., 2018).

Second, students were reported by their teachers to pursue activities such as “programming summer camps” at universities and afterschool coding activities after having

participated in study III and IV. Such findings, although only preliminary, provide initial answers to the research question regarding students' obtained competences in computational modeling and whether they transfer to other subjects or even to other situations outside the classroom (see section 1.1.2 RQ(C)). However, longitudinal studies are required to investigate the exact effects of integrating computational modeling into existing subjects in high school.

Third, the CMC framework, the taxonomy of CT and computational modeling, and the evaluation template developed during this research project are not developed specifically to biotechnology and can productively be applied within other subjects in high school education. Hence, study II relates and indicates a positive answer to the research question regarding whether the CMC approach is generic and if it is valuable to integrate computational modeling into any subject in high school education (see section 1.1.2 RQ(B)). Going back to the quote from Wing (2006, p.33) in section 2.5, stating that everyone would be eager to learn and use CT, study II, described in this dissertation, indicates that at least high school teachers from many different subjects are eager to learn and use CT and computational modeling in their teaching.

Study I, II, III, and IV all relate to the research question of scalability of this approach e.g. if integrating computing into existing education will reduce time and resources for the existing subject and specific elements of the existing curriculum (see section 1.1.2 RQ(D)). The interventions designed for study I, II, III, and IV are relatively short but have a significant effect on students' learning and even after the first intervention students exhibited signs of achieving skills in both computational modeling and subject matter. The interventions designed for study III and IV relate specifically to existing elements in the Danish curriculum for teaching biotechnology, hence the existing curriculum will not have to be reduced in order to apply these learning activities as they are already aligned with the curriculum. Findings from study II indicate that teachers who have gained programming competences are more prone to develop learning activities for computational modeling of higher taxonomic levels, than teachers who are not familiar with programming. These findings can have an impact on the scalability of, not only the CMC approach, but on the scalability of computing education overall. More research is needed to investigate the effect of professional development of teachers at all educational levels.

Study V relates to research questions concerning scalability and transferability of students' CT skills and competences and why teaching computational modeling is important for students and i.e. society as a whole. Paper 5 provides a categorization tool for reflecting on

how our investigations, as computing education researchers, can empower students in various ways. In particular for this dissertation, how the findings presented in paper 1 to 4 can scale and be sustainable for students, educators, and policy makers.

5.3 Recommendations

With this PhD study, students, teachers, educational developers, and policy makers will hopefully become more aware of the affordances by integrating computational modeling into existing subjects in high school.

Students in Danish high schools have the opportunity to affect their own education by participating in choosing a sub-content of each subject taught in high school. It is the aim that students, having participated in interventions like the ones presented in this dissertation, will actively choose computational modeling as a natural part of their existing high school education.

Teachers and educational developers have the opportunity of, to a certain degree, decide which didactics, learning activities, and assessments they will apply in Danish high schools. I will recommend that they actively seek to participate in professional development courses on computational modeling and apply the CMC approach, the evaluation template, and the CT taxonomy to design and evaluate teaching activities and strategies for their own teaching and for high schools in general.

Danish policy makers are currently evaluating the subject called technology comprehension in Denmark and will eventually decide if the subject becomes mandatory in primary and lower secondary education. This decision will inform the development of computing education in high schools in Denmark. Findings, presented in this dissertation, can serve as inspiration for policy makers on national policies for computing education at multiple educational levels and specifically for establishing computational modeling as an essential element, not only for biotechnology and STEM subjects, but for most other subjects in Danish high schools.

Last, but not least, computing education researchers can find the findings presented here useful as contributions to:

<p><i>how</i> to design computing education that work for everyone, <i>why</i> should computing education be offered to everyone,</p>

<i>what</i> can be taught in computing education for everyone.

6 Conclusion

The aim of this PhD study was to answer the research question: What and how do students learn by integrating computational modeling and coding into biotechnology? Paper 1 and 2 present a framework for how to integrate coding and computational modeling with content matter, e.g. biotechnology. Also, paper 2 present a template for developing and evaluating teaching activities, designed by the framework described in paper 1, for teaching computational modeling. Paper 3 and 4 present evidence for students' learning of both computational modeling and content knowledge, and of what they learn, by applying the framework and template described in paper 1 and 2, to teaching activities in biotechnology. Findings, obtained by mixed methods in paper 3, suggest a synergetic effect on students' learning by integrating computational modeling into the subject of biotechnology contributing to the 'what' aspect of the overarching research question. Paper 4 present a taxonomy of CT and classify students' gestures accordingly, in order to investigate students' embodied learning, and if students use gestures adaptively, while participating in computing activities. The findings of paper 4 provide additional answers to the 'what' aspect of the research question for this PhD study. Together, paper 1 and 2 answers the 'how' of the above research question, while paper 3 and 4 answer the 'what' of the research question. Going back to figure 1 in section 1.2 of this dissertation, the final paper included in the dissertation, paper 5, represent a beginning of the unraveling of *why* computing education should be integrated into existing education. The paper investigates how we as researchers position our work in relation to the empowering potential computing education have on both students, teachers, schools, and society at large.

Seen together, the papers of this dissertation represents partial answers to the overall research question by presenting findings of what elements of computational modeling and subject knowledge students learn by integrating computational modeling into existing subjects in high school education and how students' learning of computational modeling and coding is synergetic and an embodied activity.

7 Dissemination

See “List of Scientific Papers”

7.1 Additional papers

Van Mechelen, M. P. P., Musaeus, L. H., Iversen, O. S., Dindler, C., Hjort, A. (2021). A Systematic Review of Empowerment in Child-Computer Interaction Research. *Interaction Design and Children*, 119-130

Van Mechelen, M. P. P., Dindler, C., Iversen, O. S., Musaeus, L. H., Kaspersen, M. H., Ericsson, E., Fog, B. V., Schmidt, C. R., Bilstrup, K. K., Baykal, G. E., Hjort, H. A., Petersen, M. G., Leiva, G., Wagner, M., Bouvin, N. O., Kanstrup, K. H., Klokmose, C. N. (2020). Digital Fabrication in Education: Pursuing a Second Wave Agenda for FabLearn Research. *International Journal of Child-Computer Interaction*. FabLearn.

Musaeus, L. H. (2019). Developing Student's Computational Thinking through Agent-Based Modeling in Secondary Education. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 345-346).

7.2 Supplement materials

7.2.1 Computer models and learning activities

The computer models and learning activities developed for study III and IV will be made available at: <https://library.ct-denmark.org/>

7.2.2 Danish publications

Apart from the five papers presented in this dissertation, the research project has also produced the above mentioned publications (in section 7) and the following publications in Danish:

Musaeus, L. H. (2020). Modelleringskompetence i naturfag–hvad er det?. *MONA-Matematik-og Naturfagsdidaktik*, 5-5.

Caprani, O., Musaeus, L. H. (2020). Computational tankegang i teknologiforståelsesfaget. In: Andersen, I. H., Krarup, M., Philipps, M. R., Gulmann, M. (Eds.), *Teknologiforståelse på skemaet*. Unge Pædagoger, nr. 1

Caspersen, M. E., Iversen, O. S., Nielsen, M., Hjorth, A., & Musaeus, L. H. (2018). CT. In: Dolin, G. Holten Ingerslev & Hanne Sparholt Jørgensen (Red.), *Gymnasiepædagogik. En grundbog*. København: Hans Reitzels, 470-478.

7.2.3 Podcast from It-vest networking universities

The author of this dissertation has participated in a podcast presenting the CMC approach and discussing preliminary findings of the research project:

<https://www.it-vest.dk/aktiviteter/podcast-om-computational-thinking/episode-10-klimamodeller-og-cellemembraner-kode-og-faglighed-i-smuk-balance>

7.2.4 Video presentation

The author of this dissertation has participated in a video presentation of the CMC approach and discussing preliminary findings of the research project:

<https://www.youtube.com/watch?v=9R9r5w2Gr-w>

References

- Abrahamson, D., & Sánchez-García, R. (2016). Learning is moving in new ways: The ecological dynamics of mathematics education. *Journal of the Learning Sciences*, 25(2), 203-239.
- Abrahamson, D., Nathan, M. J., Williams-Pierce, C., Walkington, C., Ottmar, E. R., Soto, H., & Alibali, M. W. (2020). The future of embodied design for mathematics teaching and learning. In *Frontiers in Education* (Vol. 5, p. 147). Frontiers.
- Aho, A. V. (2012). Computation and Computational Thinking. *The computer journal*, 55(7), 832-835
- Alibali, M. W., & Nathan, M. J. (2012). Embodiment in mathematics teaching and learning: Evidence from learners' and teachers' gestures. *Journal of the learning sciences*, 21(2), 247-286.
- Anderson, T. and McKenzie, J. (2010). *Conceptual and reasoning difficulties in Sciences*, vol. 2010.
- Armoni, M. (2016). COMPUTING IN SCHOOLS Computer science, Computational Thinking, programming, coding: the anomalies of transitivity in K-12 computer science education. *ACM Inroads*, 7(4), 24-27
- Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010). Using scalable game design to teach computer science from middle school to graduate school. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 224-228).
- Basu, S., Kinnebrew, J. S., & Biswas, G. (2014). Assessing student performance in a computational-thinking based science learning environment. In *International conference on intelligent tutoring systems* (pp. 476-481). Springer, Cham.
- Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016a). Identifying middle school students' challenges in CT-based science learning. *Research and practice in technology enhanced learning*, 11(1), 1-35.
- Basu, S., Biswas, G., & Kinnebrew, J. (2016b). Using multiple representations to simultaneously learn CT and middle school science. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).

Biggs, J. B., & Collis, K. F. (2014). *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. Academic Press.

Blikstein, P., Abrahamson, D., Wilensky, U. (2005). "Netlogo: Where we are, where we're going." In *Proceedings of the annual meeting of Interaction Design and Children, Press*, vol. 23. 2005.

Blikstein, P. (2010). Connecting the science classroom and tangible interfaces: the bifocal modeling framework. In *Proceedings of the 9th International Conference of the Learning Sciences-Volume 2* (pp. 128-130).

Blikstein, P. (2013). Multimodal learning analytics. In *Proceedings of the third international conference on learning analytics and knowledge* (pp. 102-106).

Blikstein, P., & Moghadam, S. H. (2019). Computing education: Literature review and voices from the field. *The Cambridge Handbook of Computing Education Research*, Sally A. Fincher and Anthony V. Robins (Eds.). Cambridge University Press, Cambridge, 56-78.

Bloom, B. S., Krathwohl, D. R., & Masia, B. B. (1984). Bloom taxonomy of educational objectives. In *Allyn and Bacon*. Pearson Education.

Bortz, W. W., Gautam, A., Tatar, D., & Lipscomb, K. (2019). The Availability of Pedagogical Responses and the Integration of Computational Thinking. *Integrating Digital Technology in Education: School-University-Community Collaboration (2019)*, 81.

Brady, C., Holbert, N., Soylu, F., Novak, M., & Wilensky, U. (2015). Sandboxes for model-based inquiry. *Journal of Science Education and Technology*, 24(2-3), 265-286.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of Computational Thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).

Brodland, G. W. (2015). How computational models can help unlock biological systems. In *Seminars in cell & developmental biology* (Vol. 47, pp. 62-73). Academic Press.

Bruner, J. S. (1966). *Toward a theory of instruction*. Belknap/Harvard Press

Bruner, J. S. (1990). *The acts of meaning*. Harvard University Press

Burke, Q., & Kafai, Y. B. (2010). The writers' workshop for youth programmers. In *Proceedings of the 43rd SIGCSE Technical Symposium on Computer Science Education* (pp. 433-38).

Caballero, M. D., & Hjorth-Jensen, M. (2018). Integrating a computational perspective in physics courses. *arXiv preprint arXiv:1802.08871*. <https://arxiv.org/abs/1802.08871#> (visited 27.10.2021)

Caspersen, M. E., & Kolling, M. (2009). STREAM: A first programming process. *ACM Transactions on Computing Education (TOCE)*, 9(1), 1-29.

Caspersen, M. E., & Nowack, P. (2013a). Elementer af it-fagets metoder og didaktik. *Konference om informationsteknologi*, 24-25 april 2013;

Caspersen M. E., Nowack, P. (2013b). It-fagets metoder. *Center for Science Education, Aarhus University*. <https://www.gymnasieforskning.dk/fagets-metoder-version-0-3/#hey-click-me> (visited 27.10.2021)

Caspersen, M. E., & Nowack, P. (2013c). Computational Thinking and practice: A generic approach to computing in Danish high schools. In *Proceedings of the Fifteenth Australasian Computing Education Conference-Volume 136* (pp. 137-143).

Caspersen, M. E. (2017). Computational Thinking. *Dolin, G. Holten Ingerslev & Hanne Sparholt Jørgensen (Red.), Gymnasiepædagogik. En grundbog*. København: Hans Reitzels, 470-478.

Caspersen, M. E., Iversen, O. S., Nielsen, M., Musaeus, L. H., & Hjorth, H. A. (2018a). CT–hvorfør, hvad og hvordan. *Efter opdrag fra Villum Fondens bestyrelse*.

Caspersen, M. E., Gal-Ezer, J., McGettrick, A., & Nardelli, E. (2018b). *Informatics for all the strategy*. ACM.

Caspersen, M. E., Gal-Ezer, J., McGettrick, A., & Nardelli, E. (2019). Informatics as a fundamental discipline for the 21st century. *Communications of the ACM*, 62(4), 58-58.

Caspersen, M. E. (2021a). Fra teknologiforståelse til informatik. *MONA-Matematik-og Naturfagsdidaktik*, 8-8.

Caspersen, M. E. (2021b). Informatics as a Fundamental Discipline in General Education–The Danish Perspective. *DIGHUM Perspectives on Digital Humanism*. <https://dighum.ec.tuwien.ac.at/perspectives-on-digital-humanism/> (visited 27.10.2021)

Centola, D., Wilensky, U., & McKenzie, E. (2013). A hands-on modeling approach to evolution: Learning about the evolution of cooperation and altruism through multi-agent modeling-The EACH Project. In *International Conference of the Learning Sciences* (pp. 178-185). Psychology Press.

Chandrasekharan, S., & Nersessian, N. J. (2015). Building cognition: The construction of computational representations for scientific discovery. *Cognitive science*, 39(8), 1727-1763.

Computing at School (2021). Promoting and supporting excellence in computer science education. <https://www.computingschool.org.uk/> (visited 27.10.2021)

Cook, S. W., Mitchell, Z., & Goldin- Meadow, S. (2008). Gesturing makes learning last. *Cognition*, 106(2), 1047- 1058.

Cooper, S., Grover, S., Guzdial, M., & Simon, B. (2014). A future for computing education research. *Communications of the ACM*, 57(11), 34-36.

Cooper, S., Forbes, J., Fox, A., Hambrusch, S., Ko, A., & Simon, B. (2016). The Importance of Computing Education Research. *Computers and Society arXiv preprint arXiv:1604.03446*. <https://arxiv.org/abs/1604.03446> (visited 27.10.2021)

Computational Thinking in Secondary Education (CT i gymnasiefag), (2018). Region Midt, Center for Computational Thinking & Design, Aarhus University, Denmark. https://cctd.au.dk/fileadmin/user_upload/CT_i_Gymnasiefag-afslutningsrapport-ur.pdf (visited 27.10.2021)

Computer Science Education, CA. (2020). Computer Science Content Standards Development. <https://www.cde.ca.gov/be/st/ss/computerscicontentstds.asp> (visited 27.10.2021)

CS Education Graduates (2021). Exploring pathways of future CSEd researchers. <https://www.csedgrad.org/> (visited 27.10.2021)

CSTA, K. (12). Computer science standards. *Computer Science Teachers Association*, 12.

Daniels, M., & Pears, A. (2012). Models and methods for computing education research. In *Proceedings of the Fourteenth Australasian Computing Education Conference-Volume 123* (pp. 95-102).

D'Avanzo, C. (2008). Biology concept inventories: overview, status, and next steps. *BioScience*, 58(11), 1079-1085.

Denning, P. J. (2009). The profession of IT Beyond Computational Thinking. *Communications of the ACM*, 52(6), 28-30.

Denning, P. J. (2017). Remaining trouble spots with Computational Thinking. *Communications of the ACM*, 60(6), 33-39.

Denning, P. J. (2018). The computing profession. *Communications of the ACM*, 61(3), 33-35.

- Denning, P. J., & Tedre, M. (2019). *Computational Thinking*. MIT Press.
- Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, 32(1), 5-8.
- DiSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy*. Mit Press.
- DiSessa, A. A. (2018). Computational literacy and “the big picture” concerning computers in mathematics education. *Mathematical thinking and learning*, 20(1), 3-31.
- Di Ventura, B., Lemerle, C., Michalodimitrakis, K., & Serrano, L. (2006). From in vivo to in silico biology and back. *Nature*, 443(7111), 527-533.
- Duit, R. (2009). Bibliography–STCSE. *Students' and Teachers' Conceptions and Science Education* http://www.ipn.uni-kiel.de/aktuell/stcse/download_stcse.html (visited 27.10.2021)
- Egebo, L., Johansen, J. S., Jørgensen, F. G., Mantoni, T. S., Wolff, H. (2017). *Bioteknologi A*. Nucleus, Aarhus, Denmark
- Ericson, B. J., YeckehZaare, I., & Guzdial, M. J. (2019). Runestone Interactive Ebooks: A Research Platform for On-line Computer Science Learning. In *Proceedings from ICER '19*, Toronto, ON, Canada.
- Fernandez, C., Fuhrmann, T., de Deus Lopes, R., & Blikstein, P. (2021). Designing domain-specific blocks for diffusion: The dialogue between pedagogical principles and design decisions. In *Interaction Design and Children* (pp. 461-465).
- Fincher, S., Tenenberg, J., Dorn, B., Hundhausen, C., McCartney, R., and Murphy, L. (2019). Computing education research today. *The Cambridge handbook of computing education research*, 1, 40-55.
- Flood, V. J., Shvarts, A., & Abrahamson, D. (2020). Teaching with embodied learning technologies for mathematics: responsive teaching for embodied learning. *ZDM*, 52(7), 1307-1331.
- Fuhrmann, T., Salehi, S., & Blikstein, P. (2013). Meta-modeling knowledge: Comparing model construction and model interaction in bifocal modeling. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 483-486).
- Garvin-Doxas, K. and Klymkowsky, M. W. (2008). Understanding Randomness and its impact on Student Learning: Lessons from the Biology Concept Inventory (BCI). *Life Science Education* 7, 227-233.
- Gautam, A., Bortz, W., & Tatar, D. (2020). Abstraction through multiple representations in an integrated Computational Thinking environment. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 393-399).
- Gibson, J. J. (1979). The ecological approach to visual perception. *Boston: Houghton Mifflin*. Glotzbach, P. A., and Heft, H. (1982). *Ecological and phenomenological approaches to perception* *Nous* 16108–121
- Gilbert, N. (2004). Agent-based social simulation: dealing with complexity. *The Complex Systems Network of Excellence*, 9(25), 1-14.
- Gouvea, J., & Passmore, C. (2017). Models of versus ‘Models for. *Science & Education*, 26(1), 49-63.
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. *Computer science education: Perspectives on teaching and learning in school*, 19.
- Grover, S. (2018). A tale of two CTs (and a revised timeline for Computational Thinking). *Communications of the ACM*. Retrieved from <https://cacm.acm.org/blogs/blog-cacm/232488-a-tale-of-two-cts-and-a-revised-timeline-for-computational-thinking/fulltext>.

Gujberova, M., & Kalas, I. (2013). Designing productive gradations of tasks in primary programming education. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education* (pp. 108-117).

Guzdial, M., Ericson, B. J., McKlin, T., & Engelman, S. (2012). A statewide survey on computing education pathways and influences: factors in broadening participation in computing. In *Proceedings of the ninth annual international conference on International computing education research* (pp. 143-150).

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6), 1-165.

Guzdial, M., Kay, A., Norris, C., & Soloway, E. (2019). Computational Thinking should just be good thinking. *Communications of the ACM*, 62(11), 28-30.

Guzdial, M. (2019b). What is not Computational Thinking? Curly braces, $x = x + 1$, and else. <https://computinged.wordpress.com/2019/05/03/whats-not-computational-thinking-maybe-x-x-1/> (visited 27.10.2021)

Guzdial, M. (2019c). A new definition of Computational Thinking: It's the Friction that we want to Minimize unless it's Generative. <https://computinged.wordpress.com/2019/04/29/what-is-computational-thinking-its-the-friction-that-we-want-to-minimize/> (visited 27.10.2021)

Guzdial, M. (2021). Considering the Danish Informatics Curriculum: Comparing National Computer Science Curricula. *Computing Education Research Blog*. <https://computinged.wordpress.com/2021/07/12/considering-the-danish-informatics-curriculum-comparing-national-computer-science-curricula/> (visited 27.10.2021)

Heintz, F., Mannila, L., Nygård, K., Parnes, P., & Regnell, B. (2015). Computing at school in Sweden—experiences from introducing computer science within existing subjects. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 118-130). Springer, Cham.

Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force concept inventory. *The physics teacher*, 30(3), 141-158.

Hjort, A., Head, B., Brady, C., & Wilensky, U. (2020). LevelSpace: A NetLogo Extension for Multi-Level Agent-Based Modeling. *Journal of Artificial Societies & Social Simulation*, 23(1).

Hutchins, N. M., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., ... & McElhaney, K. (2020). C2STEM: A system for synergistic learning of physics and Computational Thinking. *Journal of Science Education and Technology*, 29(1), 83-100.

Informatics Education (2021). Informatics Europe. <https://www.informatics-europe.org/activities/informatics-education.html> (visited 27.10.2021)

Ioannidou, A., Repenning, A., & Webb, D. C. (2009). AgentCubes: Incremental 3D end-user development. *Journal of Visual Languages & Computing*, 20(4), 236-251.

ISTE Standards for Computer Science Educators (2014). http://www.iste.org/docs/pdfs/2014_ISTE_Standards-CSE_PDF.pdf (visited 27.10.2021)

Iversen, O. S., Smith, R. C., & Dindler, C. (2018). From computational thinking to computational empowerment: a 21st century PD agenda. In *Proceedings of the 15th Participatory Design Conference: Full Papers-Volume 1* (pp. 1-11)

Jacobson, M. J., Kim, B., Pathak, S., & Zhang, B. (2015). To guide or not to guide: issues in the sequencing of pedagogical structure in computational model-based learning. *Interactive Learning Environments*, 23(6), 715-730.

Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014). Embedding Computational Thinking in science, technology, engineering, and math (CT-STEM). In *future directions in computer science education summit meeting, Orlando, FL*.

- Jones, S. P., Mitchell, B., & Humphreys, S. (2013). Computing at school in the UK. *CACM Report*.
- Kafura, D., & Tatar, D. (2011). Initial experience with a computational thinking course for computer science students. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 251-256).
- Kendon, A. (2004). *Gesture: Visible Action as Utterance*. Cambridge, UK, Cambridge University Press.
- Klymkowsky, M. W. and Garvin-Doxas, K. (2008). Recognizing Student Misconceptions through Ed's Tool and the Biology Concept Inventory. *PLoS Biol* 6, e3
- Klymkowsky, M. W., Underwood, S. M., & Garvin-Doxas, R. K. (2010). Biological Concepts Instrument (BCI): A diagnostic tool for revealing student thinking. *arXiv preprint arXiv:1012.4501*. <https://arxiv.org/abs/1012.4501> (visited 27.10.2021)
- Ko, A. 2020. What is computing education research? <https://faculty.washington.edu/ajko/cer> (visited 27.10.2021)
- Ko, A. J., Oleson, A., Ryan, N., Register, Y., Xie, B., Tari, M., ... & Loksa, D. (2020). It is time for more critical CS education. *Communications of the ACM*, 63(11), 31-33.
- Kynigos, C. (2007). Using half-baked microworlds to challenge teacher educators' knowing. *International journal of computers for mathematical learning*, 12(2), 87-111.
- Kölling, M. (2021). Comments on the Danish informatics curriculum. *King's College London*
- Lamprou, A., & Repenning, A. (2018). Teaching how to teach computational thinking. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 69-74).'
- Larrazza-Mendiluze, E., & Garay-Vitoria, N. (2012). A comparison between lecturers' and students' concept maps related to the input/output topic in computer architecture. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research* (pp. 57-66).
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32-37.
- Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020a). 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(1), 1-8.
- Lee, I., & Malyn-Smith, J. (2020b). Computational thinking integration patterns along the framework defining computational thinking from a disciplinary perspective. *Journal of science education and technology*, 29(1), 9-18.
- Li, Y., Schoenfeld, A. H., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020a). On computational thinking and STEM education. *Journal for STEM education Research*, 3, 147-166.
- Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020b). Computational thinking is more about thinking than computing. *Journal for STEM education Research*, 3, 1-18.
- Macrine, S. L., & Fugate, J. M. (2020). Embodied cognition. In *Oxford Research Encyclopedia of Education*.
- Malmi, L., Sheard, J., Kinnunen, P., & Sinclair, J. (2019). Computing Education Theories: What Are They and How Are They Used?. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 187-197).
- Maloney, J., Resnick, M. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing education*. Vol. 10 (4), 1-15.

- Markham, K. M., Mintzes, J. J., & Jones, M. G. (1994). The concept map as a research and evaluation tool: Further evidence of validity. *Journal of research in science teaching*, 31(1), 91-101.
- McGarr, O., Mifsud, L., & Colomer Rubio, J. C. (2021). Digital competence in teacher education: comparing national policies in Norway, Ireland and Spain. *Learning, Media and Technology*, 1-15.
- McNeill, D. (1992). *Hand and Mind: What gestures reveal about thought*. Chicago, IL: University of Chicago Press.
- Moen, P. (2009). Concept maps as a device for learning database concepts. In *7th Workshop on Teaching, Learning and Assessment in Databases, Birmingham, UK, HEA* (pp. 2006-01).
- Morris, C., Chikwa, G. (2013). Screencasts: How effective are they and how do students engage with them? *Active Learning in Higher Education* 15 (1): 25-37
- Naur, P. (1967). *Datamaskinerne og samfundet*. Munksgaard. Denmark
- National Research Council (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking*. National Research Council, 2010 (www8.nationalacademies.org/cp/projectview.aspx?key=48969)
- National Research Council. (2011). *Report of a workshop on the pedagogical aspects of computational thinking*. National Academies Press.
- Nelson, G. L., & Ko, A. J. (2019). John Henry AWARD: on use of theory in computing education research. *ACM Inroads*, 10(1), 74-83.
- Nersessian, N. (2008). Model-based reasoning in scientific practice. In *Teaching scientific inquiry* (pp. 57-79). Brill sense.
- Nersessian, N. J., Newstetter, W., & Sun, Y. (2006). Promoting model-based reasoning in problem-based learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 28, No. 28).
- Novak and Musonda (1991). A twelve-year longitudinal study of science concept learning'. In *American Educational Research Journal*, 28(1), pp117-153.
- Novak, J. D., & Cañas, A. J. (2007). Theoretical origins of concept maps, how to construct them, and uses in education. *Reflecting Education*, 3(1), 29-42.)
- Nowack, P., & Caspersen, M. E. (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* (pp. 147-151).
- Odden, T. O. B., & Malthe-Sørenssen, A. (2020). Using computational essays to scaffold professional physics practice. *European Journal of Physics*, 42(1), 015701.
- Parker, M. C., Guzdial, M., & Engleman, S. (2016). Replication, validation, and use of a language independent CS1 knowledge assessment. In *Proceedings of the 2016 ACM conference on international computing education research* (pp. 93-101).
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., p.VIII
- Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, 36(2), 1-11.
- Passmore, C., Stewart, J., & Cartier, J. (2009). Model-based inquiry and school science: Creating connections. *School Science and Mathematics*, 109(7), 394-402.
- Passmore, C., Schwarz, C. V., & Mankowski, J. (2017). Developing and using models. *Helping students make sense of the world using next generation science and engineering practices*, 109-134.

Penner, D. E. (2000). Explaining systems: investigating middle school students' understanding of emergent phenomena. *Journal of Research in Science Teaching*, 37, 784–806.

Piaget, J. (1972). *Psykologi og erkendelsesteori*. Hans Reitzels Forlag

Priowijanto, E. A. (2013). *Simulasi Digital*. South East Asia Ministers of Education Organization Regional Open Learning Center.

Repenning, A. (2000). AgentSheets®: An interactive simulation environment with end-user programmable agents. *Interaction*.

Resnick, M. (1992). *Beyond the Centralized Mindset: Explorations in Massively-Parallel Microworlds*, PhD Dissertation, MIT.

Resnick, M. (1996a). Beyond the Centralised Mindset, *The Journal of the Learning Sciences*, 5 (1), pp. 1-22.

Resnick, M. (1996b). Distributed Constructionism, *Proceedings of the 1996 International Conference on Learning Sciences*, ICLS '96, pp. 280-284.

Resnick, M. (1996c). StarLogo: An Environment for Decentralized Modeling and Decentralized Thinking, *Conference Companion on Human Factors in Computing Systems*, CHI '96, pp. 11-12.

Rich, K. M., Binkowski, T. A., Strickland, C., & Franklin, D. (2018). Decomposition: A k-8 computational thinking learning trajectory. In *Proceedings of the 2018 ACM conference on international computing education research* (pp. 124-132).

Roth, W. M. (2020). The Thinking Body of Mathematics. In *Adventures of Mind and Mathematics* (pp. 199-223). Springer, Cham.

Sand, O. P., Odden, T. O. B., Lindstrøm, C., & Caballero, M. D. (2018). How computation can facilitate sensemaking about physics: A case study. In *PERC Proceedings arXiv preprint arXiv:1807.03999*. <https://arxiv.org/abs/1807.03999> (visited 27.10.2021)

Schwarz, C. V., & White, B. Y. (2005). Metamodeling knowledge: Developing students' understanding of scientific modeling. *Cognition and instruction*, 23(2), 165-205.

Secondary Educations (Gymnasiale Uddannelser), (2020). Ministry of Education. <https://www.uvm.dk/gymnasiale-uddannelser> (visited 27.10.2021)

Sentance, S. (2018). Introducing why teach computer science in school. *Computer Science Education. Perspectives on Teaching and Learning in School*, S. Sentance, E. Barendsen, and C. Schulte (Eds.). Bloomsbury Academic, London, 3-4.

Sentance, S., Selby, C., & Kallia, M. (2018). Assessment in the Computing classroom. *Computer Science Education: Perspectives on Teaching and Learning in School*, 151.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2), 351-380.

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. *Computational Thinking in the STEM disciplines*, 49-72.

Shapiro, L., & Stolz, S. A. (2019). Embodied cognition and its significance for education. *Theory and Research in Education*, 17(1), 19- 39.

Sherin, B., diSessa, A. A., & Hammer, D. M. (1993). Dynaturtle revisited: Learning physics through collaborative design of a computer model. *Interactive Learning Environments*, 3(2), 91–118.

Sheu, F. R., & Chen, N. S. (2014). Taking a signal: A review of gesture- based computing research in education. *Computers & Education*, 78, 268- 277.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying Computational Thinking. *Educational Research Review*, 22, 142-158.

Sien, V. Y. (2011). Teaching object-oriented modelling using concept maps. *Electronic Communications of the EASST*, 34.

Solomon, A., Guzdial, M., DiSalvo, B., & Shapiro, B. R. (2018). Applying a gesture taxonomy to introductory computing concepts. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 250-257).

Solomon, A., Bae, M., DiSalvo, B., & Guzdial, M. (2020). Embodied Representations in Computing Education: How Gesture, Embodied Language, and Tool Use Support Teaching Recursion. In *Proceeding of International Society of the Learning Sciences (ICLS)*

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798.

The Central Jutland Technology Pact (2018). Region Midt. <https://www.rm.dk/regional-udvikling/fremtidssikrede-uddannelser-og-kompetenceloft/den-midtyjske-teknologipagt/> (visited 27.10.2021)

The Royal Society. 2017. After the reboot: Computing education in UK schools. Technical Report. The Royal Society, London, United Kingdom. 1--116 pages. https://royalsociety.org/topics-policy/projects/computing-education/?utm_source=royalsociety.org&utm_medium=redirect&utm_campaign=computing-education (visited 27.10.2021)

Tripto, J., Assaraf, O. B. Z., & Amit, M. (2013). Mapping what they know: Concept maps as an effective tool for assessing students' systems thinking. *American Journal of Operations Research*, Vol. 3 (1A). https://www.scirp.org/html/16-1040203_27564.htm?pagespeed=noscript (visited 27.10.2021)

Tuhkala, A., Wagner, M. L., Iversen, O. S., & Kärkkäinen, T. (2019). Technology Comprehension—Combining computing, design, and societal reflection as a national subject. *International Journal of Child-Computer Interaction*, 20, 54-63.

Tissenbaum, M., Sheldon, J., Seop, L., Lee, C. H., & Lao, N. (2017). Critical computational empowerment: Engaging youth as shapers of the digital future. In *2017 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1705-1708). IEEE

Tisue, S., & Wilensky, U. (2004). Netlogo: A simple environment for modeling complexity. In *International conference on complex systems* (Vol. 21, pp. 16-21).

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728.

Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Harvard university press.

Wagh, A., Cook-Whitt, K., & Wilensky, U. (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), 615-641.

Walkington, C., Woods, D., Nathan, M. J., Chelule, G., & Wang, M. (2019). Does restricting hand gestures impair mathematical reasoning? *Learning and Instruction*, 64, 101225.

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring Computational Thinking in middle school. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 215-220).

Wei, W., & Yue, K. B. (2016). Using concept maps to teach and assess critical thinking in IS education. In *Proceedings of AMCIS 2016*

Wei, W., & Yue, K. B. (2017). Concept mapping in computer science education. *Journal of Computing Sciences in Colleges*, 32(4), 13-20.

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.

Weintrop, D., Holbert, N., & Tissenbaum, M. (2020). Considering Alternative Endpoints: An Exploration in the Space of Computing Educations. *CONSTRUCTIONISM 2020*, 220.

Wilensky, U. (1999). *NetLogo*. Evanston, IL: Center for connected learning and computer-based modeling, Northwestern University

Wilensky, U. & Resnick, M. (1999). "Thinking in Levels: A Dynamic Systems Approach to Making Sense of the World", *Journal of Science Education and Technology*, 8 (1), pp. 3-19.

Wilensky, U., & Stroup, W. (2002). Participatory Simulations: Envisioning the networked classroom as a way to support systems learning for all. In *annual meeting of the American Research Education Association, New Orleans, LA*.

Wilensky, U. (2003). Statistical mechanics for secondary school: the GasLab modeling toolkit. *International Journal of Computers for Mathematical Learning*, 8(1), 1–41 (special issue on agent-based modeling).

Wilensky, U., & Reisman, K. (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), 171–209.

Wilensky, U. (2006). *NetLogo Rope 3D model*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/Rope3D> (visited 27.10.2021)

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. *Constructionism 2010*, Paris, France.

Wilensky, U., & Rand, W. (2015). *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. Mit Press.

Wilkerson-Jerde, M. H., Gravel, B. E., & Macrander, C. A. (2015). Exploring shifts in middle school learners' modeling activity while generating drawings, animations, and computational simulations of molecular diffusion. *Journal of Science Education and Technology*, 24(2-3), 396-415.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.

Wing, J. M. (2008). CT and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.

Wing, J. (2009). Computational thinking. *Journal of computing sciences in colleges*, 24(6), 6-7.

Wing, J. (2011). Research notebook: Computational Thinking—What and why. *The link magazine*, 6.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1-16.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, 60(6), 565-568.

Zoanetti, N., & Griffin, P. (2017). Log-file data as indicators for problem-solving processes. Lerche, T., & Kiel, E. (2018). Predicting student achievement in learning management systems by log data analysis. *Computers in Human Behavior*, 89, 367-372.