



Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Pædagogiske spændingsfelter ved inddragelse af programmering i grundskolen

Læreres perspektiv på at inddrage programmering som fagligt element i deres undervisning

Elving, Pernille Risør; Ryberg, Thomas

Published in:
Læring og Medier

DOI (link to publication from Publisher):
[10.7146/lom.v11i19.103100](https://doi.org/10.7146/lom.v11i19.103100)

Creative Commons License
CC BY-NC-ND 4.0

Publication date:
2018

Document Version
Også kaldet Forlagets PDF

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Elving, P. R., & Ryberg, T. (2018). Pædagogiske spændingsfelter ved inddragelse af programmering i grundskolen: Læreres perspektiv på at inddrage programmering som fagligt element i deres undervisning. *Læring og Medier*, 11(19). <https://doi.org/10.7146/lom.v11i19.103100>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- ? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- ? You may not further distribute the material or use it for any profit-making activity or commercial gain
- ? You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.



Pædagogiske dilemmaer ved inddragelse af programmering i grundskolen

– Læreres perspektiv på at inddrage programmering som fagligt element i deres undervisning

Pernille Risør Elving

Videnskabelig Assistent

Center for Undervisningsudvikling og Digitale Medier,
Aarhus Universitet.



Thomas Ryberg

Professor

Institut for Kommunikation og Psykologi, Aalborg
Universitet.



Abstract (dansk)

I artiklen undersøges, hvordan programmering benyttes i en pædagogisk praksis i grundskolen, samt hvilket perspektiv faglærerne har på at inddrage programmering som fagligt element i deres undervisning. Artiklen bygger på en empirisk undersøgelse af tre skolars inddragelse af programmering i undervisningen på mellemtrinnet under Coding Class-forløb i Vejle Kommune, hvor fire faglærere blev interviewet omkring deres erfaringer med forløbet. Analysen af lærernes erfaringer viste en række dilemmaer på tværs af de tre cases omkring brugen af programmering i den faglige undervisning. Dilemmaerne omhandler blandt andet lærernes vægtning af at bruge programmering som et fagspecifikt eller tværfagligt element, elevernes faglige udbytte overfor deres interesser samt af lærerens rolle som enten ekspert i klasseværelset eller pædagogisk guide. Erfaringerne fra lærerne var blandt andet, at de alle oplevede højt engagement og fordybelse med programmeringsaktiviteterne blandt eleverne, men flere udtrykte behov for større kontrol med elevernes fagspecifikke udbytte og oplevede begrænsninger omkring egne tekniske kompetencer. På baggrund af analysen opstilles en række konkrete pædagogiske overvejelser, der udgør opmærksomhedspunkter omkring fremtidig planlægning og inddragelse af programmering i undervisningen samt omkring lærerens rolle i sådanne forløb.

Abstract (engelsk)

This article examines how programming is adopted as a pedagogical practice in K-12 and what perspective teachers have on implementing programming as a subject-specific element in their teaching activities. The article is based on an empirical study of three schools' use of programming during Coding Class in Vejle Kommune, where four teachers were interviewed about their experiences with the project. The analysis shows a number of challenges across the three cases including; whether programming should be used as a subject-specific or interdisciplinary element and whether the goal is to enable development of students' subject-specific learning or support their interests, and finally how teachers approach their own role in the classroom. All the teachers in the study experienced high levels of engagement and immersion in the programming activities among the students but several of the teachers lacked control over the students' learning outcomes and experienced limitations in their own technical skills. Based on the analysis, a number of concrete pedagogical considerations are presented, which can form the basis for planning and incorporating future programming activities in K-12.

Introduktion

Ifølge IT-Branchen oplever hver tredje it-virksomhed i Danmark en mangel på kvalificeret arbejdskraft, fordi efterspørgslen på it-uddannede er større end udbuddet (IT-Branchen, u.å.2). Danmarks Vækstråd ser ligeledes en stor stigning i efterspørgsel på it-specialister og argumenterer for, at fremtidens arbejdsmarked kræver en bredere kompetencesammensætning, som både indebærer kreative, tekniske og it-baserede færdigheder (Danmarks Vækstråd, 2016, s. 25). For at imødekomme efterspørgslen på medarbejdere med teknologisk viden og erfaring kræver det ifølge Danmarks Vækstråd, at der stilles nye krav til vores uddannelsessystem; det skal tilpasses den fremtid, arbejdsmarkedet står over for (Danmarks Vækstråd, 2016, s. 82). Danmarks Vækstråd anbefaler derfor, at computational thinking introduceres tidligt for børn og unge for at kunne møde denne udvikling:

Der bør sættes ind på et tidligt stadie for at vække især børn og unges interesse for „Computational thinking“, hvilket på sigt kan være med til at imødekomme virksomhedernes voksende rekrutteringsbehov og dermed bidrage til at forbedre landets produktivitet og konkurrenceevne (Danmarks Vækstråd, 2016, s. 25).

Flere initiativer søger at imødekomme udviklingen, hvilket eksempelvis ses i Folkeskolereformen fra sommeren 2014, hvor it og medier inkluderes som tværgående tema for undervisningen i grundskolen. Der findes dog ligeledes flere initiativer der arbejder på, at it og programmering inddrages mere direkte i grundskolens undervisning. Coding Class er et af de initiativer, som er blevet igangsat for at gøre it til en del af den almene dannelse i grundskolen med henblik på at skabe større interesse for it blandt børn og unge, og derved spore dem ind på informationsteknologi som et fremtidigt karrierevalg (IT-Branchen, u.å.1). Visionen bag Coding Class er “[...] at få børn og unge til at tænke på teknologi og it – og at gøre it til en del af den almene dannelse i folkeskolen” (IT-Branchen, u.å.1). Coding Class består af designorienterede programmeringsforløb på mellemtrinnet i grundskolen, hvilket giver nye muligheder for at arbejde med programmering direkte i grundskolens undervisning. Med Coding Class ønsker IT-Branchen at skabe interesse hos eleverne for at arbejde innovativt med it og opbygge elevernes it-kompetencer. Dette vil ifølge IT-branchen både kunne gavne Danmark som helhed samt elevernes egen fremtid (IT-Branchen, u.å.1).

Der er således både stærke politiske og industrielle stemmer, der har en interesse i at få programmering, teknologi og it-kompetencer sat højere på både folkeskolens og de unges agenda. Både i Danmark og internationalt er der rejst debatter omkring grundskolens formål, metode og ansvar i relation til dette. Det vil vi i denne artikel

ikke gå nærmere ind i, selvom debatten både er relevant og vigtig. Artiklen vil derimod forholde sig til de initiativer, der allerede er sat i gang og er ved at blive rullet ud i større skala. Vi anskuer det derfor i denne artikel nødvendigt og presserende at undersøge, hvordan allerede igangsatte initiativer indvirker på lærerne og deres tilrettelæggelse af undervisningen, da det er denne virkelighed lærerne står over for og skal forholde sig til i øjeblikket.

I denne artikel vil vi derfor undersøge faglærernes erfaringer med og perspektiv på at inddrage programmering som fagligt element i deres undervisning på mellemtrinnet i grundskolen. Formålet er at give et indblik i de udfordringer af faglig og professionel art, som en underviser møder, når det faglige arbejde skal inddrage teknologi og programmering. Det er således ikke artiklens formål at belyse elevernes udbytte i form af generelle kompetencer, fx computational thinking og digitale dannelse, eller en for-og-imod debat omkring programmering i grundskolen. Formålet er derimod at belyse de dilemmaer, som faglærerne i praksis møder ved inddragelse af programmering i deres undervisning.

Artiklen vil indledningsvis rammesætte undersøgelsens relevans gennem et kort overblik over forskningsfeltet. Dernæst præsenteres casestudiet, herunder den metodiske udførelse heraf samt en beskrivelse af de tre konkrete cases. Dette leder til en caseanalyse, hvor lærernes erfaringer med Coding Class-forløbet belyses. Analysen er struktureret omkring en række dilemmaer, der trådte frem i arbejdet med analyse og kodningen af data, og som omhandler inddragelsen af programmering i den faglige undervisning. På baggrund af analysen opstiller artiklen nogle konkrete pædagogiske overvejelser, der udgør opmærksomhedspunkter til planlægning af undervisningsforløb, som inddrager programmeringsaktiviteter. Afslutningsvis fremføres undersøgelsens konklusion, herunder lærernes perspektiv på at inddrage programmering i deres undervisning og de udfordringer, som opstår i denne praksis.

Kort om forskningsfeltet

Flere undersøgelser inden for feltet studerer brugen af programmering i forbindelse med naturvidenskabelige fag i grundskolen ofte med fokus på udviklingen af elevernes computational thinking. Wing beskriver computational thinking som vores evne til at udføre problemløsning, kommunikere samt interagere med andre mennesker (Wing, 2006, s. 34). Computational thinking omhandler ifølge Wing individers måde at håndtere abstrakt og konceptuel tænkning og afhænger dermed ikke af, om man kan programmere eller få computeren til at udføre en opgave: *“Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction”* (Wing, 2006, s. 34). Computational thinking dækker således over kompetencer, der ikke nødvendigvis

kræver tekniske evner, men er en måde at tænke på, der gør os i stand til at agere hensigtsmæssigt med computere. Begrebet indebærer derfor generelle kompetencer, der sætter fokus på nogle digitale grundkompetencer, som kan benyttes på tværs af fagligheder. Flere undersøgelser inden for forskningsområdet undersøger, hvordan programmeringsundervisning kan understøtte sådanne generelle kompetencer. Zaharija, Mladenovic og Boljat (2013) belyser eksempelvis, hvorledes introduktionen af programmering til elever i grundskolen kan bidrage til at udvikle deres evner inden for problemløsning samt logisk tænkning gennem programmering og leg med robotter. Kalelioğlu (2015) har undersøgt udviklingen af grundskoleelevers refleksive evner og kompetencer inden for problemløsning gennem et forløb med undervisning i Code.org, som kun viste en lille forbedring af disse kompetencer, men hvor lærerne oplevede et stort engagement blandt eleverne, som ligeledes udviste en stor interesse for programmeringsaktiviteterne. I en nyligt udgivet rapport med dokumentation og evaluering af Coding Class undersøger Hansbøl og Ejsing-Duun (2017) ligeledes udvikling af elevernes digitale kompetencer samt computational thinking under Coding Class forløbene. Rapporten viste blandt andet, at programmeringsaktiviteterne understøttede elevernes kreative udfoldelse samt deres evne til at udføre computational thinking, omend dette ikke var lærernes tiltænkte mål med at inddrage programmering i deres undervisning (Hansbøl & Ejsing-Duun, 2017). Rapporten viste dog ligeledes, at det kan være krævende for uerfarne lærere at indgå i aktiviteterne omkring programmering og computational thinking på grund af deres manglende erfaring (Hansbøl & Ejsing-Duun, 2017). En undersøgelse udført af Nielsen, Pedersen og Majgaard (2015) belyser elevernes evner inden for kreativ udfoldelse og problemløsning med robotprogrammering på tværs af matematik, natur/teknik og engelsk. Her bidrog programmeringen både fagligt og personligt, skabte højt engagement blandt eleverne og viste, via skriftlige refleksioner fra lærerne, en udvikling af flere generelle kompetencer blandt eleverne. Erfaringen var dog, at det krævede en særlig indsats at indarbejde aktiviteterne meningsfuldt i fagene. Undersøgelser der belyser elevernes fagspecifikke udbytte fokuserer ofte på undervisningen inden for de naturvidenskabelige fag. Ejsing-Duun og Misfeldt (2015) undersøger eksempelvis mulighederne for at anvende robotprogrammering i matematikundervisningen, og hvordan inddragelsen heraf kan understøtte elevernes abstrakte og algoritmiske tænkning, samt deres opsøgning af matematisk viden. Undersøgelsen peger på, at måden hvorpå elever præsenteres for opgaver har stor betydning for læringspotentialet, og at det er krævende for læreren at knytte aktiviteterne direkte til fagets mål, men at robotprogrammering kan udgøre et meningsfuldt element i undervisningen. Her undersøger Foerster (2016) mere direkte, hvorvidt programmering kan bidrage til elevernes matematiske læring omkring geometri, hvilket viste store faglige fordele i forhold til kontrolgruppen, der ikke fik koblet programmering ind i faget. Lopez, González og Cano (2016) fremlægger, hvordan et undervisningsforløb med programmet Scratch kan have en

positiv indflydelse på elevernes beregningsmetoder, deres logiske tænkning samt praktiske anvendelse af tekniske kompetencer, men at udbyttet afhang meget af elevernes engagement i opgaverne.

Undersøgelserne ovenfor eksemplificerer, hvorledes programmering kan bidrage i især matematikundervisningen i grundskolen. Undersøgelser omkring elevernes generelle kompetencer, herunder computational thinking, samt omkring elevernes faglige udbytte viser forskellige resultater i forhold til elevernes udbytte.

Undersøgelserne giver et billede af, at den konkrete tilrettelæggelse af forløbene har stor betydning for, hvad der læres, og at det kan være krævende at knytte aktiviteterne til de faglige mål særligt for lærere med mindre erfaring med denne type af aktiviteter. Der er således en række didaktiske problemstillinger, som må indtænkes i designet af undervisningsforløb der gør brug af programmering. Det er sådanne problemstillinger vi i denne artikel behandler som en række pædagogiske dilemmaer, som lærere må navigere i ved tilrettelæggelse af denne type undervisning. Undersøgelsen fokuserer således på lærernes mål og erfaringer med at inddrage programmering i deres undervisningsforløb for at give indblik i, hvordan undervisningsaktiviteter med fordel kan designes, så programmering og udvikling af elevernes teknologiske færdigheder kan inddrages meningsfuldt i fagene. Med henblik på at forbedre arbejdet med at integrere it og programmering i grundskolen fremadrettet undersøges samtidig, hvilke udfordringer lærerne oplever hermed og hvilke kompetencer, det kræver at inddrage programmering i den faglige undervisning.

Baggrund: Coding Class i Vejle Kommune

Følgende afsnit beskriver først undersøgelsens metode og herefter gennemgås de overordnede rammer for hver enkelt case samt den overordnede kontekst, som casestudiet tager udgangspunkt i. Casestudiet er udført af artiklens førsteforfatter, der har været observerende på skolebesøgene snarere end interventionistisk. Således har undersøgelsen ikke involveret design af forløbene, valg af emne, eller en involvering i de didaktiske beslutninger omkring forløbene.

Undersøgelsesmetode

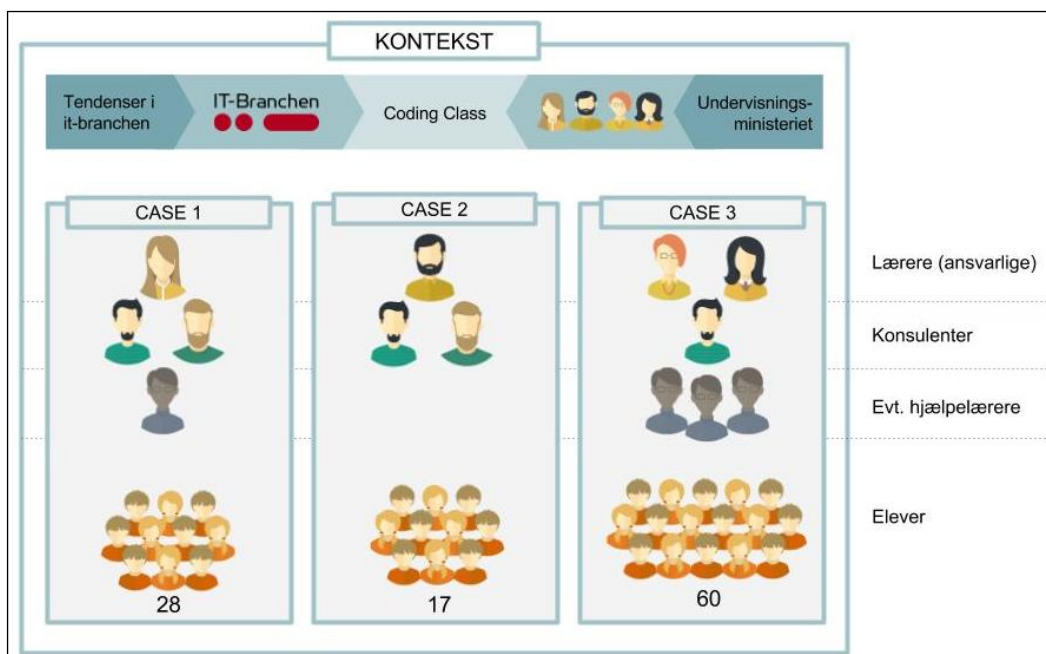
I den aktuelle undersøgelse benyttes casestudiet som strategi til at få indsigt i brugen af programmering i undervisningen under Coding Class-forløb på tre skoler i Vejle Kommune i skoleåret 2016/2017. Casestudiet er valgt som metode, da det giver mulighed for, at *“explore and investigate contemporary real-life phenomenon through detailed contextual analysis of a limited number of events or conditions, and their relationships”* (Zainal, 2007, s. 2). Ifølge Flyvbjerg giver casestudiet mulighed for at kunne komme i tættere forbindelse med virkelige situationer, hvilket giver et mere nuanceret syn på det, som undersøges (Flyvbjerg, 2010, s. 467). Det aktuelle

casestudie inddrager tre skoler, hvor rammerne for undervisningsaktiviteterne under Coding Class er forskellige, hvilket har den fordel at informationerne om proces og resultat på tværs af de forskelligartede cases kan udfordre forståelsen for det undersøgte (Flyvbjerg, 2010, s. 475). Gennem brug af observationer og interviews under casestudiet undersøges læringssituationen under Coding Class-forløbene i Vejle Kommune og lærernes erfaringer med at inddrage programmering som fagligt element i deres undervisning. Undersøgelsens feltarbejde udgøres af uformelle dialoger under skolebesøg, observationer samt interviews. Under skolebesøgene foregik både observationerne samt de uformelle dialoger med de involverede lærere, elever, eventuelle hjælpelærere samt konsulenter fra Coding Class. Dialogerne med de forskellige aktører havde til formål at skabe en bedre forståelse for situationerne og aktiviteterne under hver case samt give indblik i den samlede kontekst for casestudiet. Disse dialoger fungerede som uformelle interviews, der udgjorde en forberedelse til de formelle interviews, da *“through informal interviews the ethnographer begins to learn enough about the community to conduct more structured, systematic interviews”* (Blomberg et al., 1993, s. 134). Observationerne foregik på alle tre skoler i løbet af fire hele dage, hvor undervisningen blev observeret og noter blev skrevet ned undervejs. Observationernes fokus var at forstå læringssituationen under Coding Class-forløbene ved at observere dynamikken i klasseværelset, hvilket både indebærer samspil mellem eleverne samt konsulenterne og lærernes interaktion i klasseværelset og med eleverne. Forskerpositionen var af denne grund i en observatør-rolle under observationerne for at kunne forholde sig diskret til de aktører og de aktiviteter som udfoldede sig (Stake, 1995, s. 12). Observationerne blev senere kodet ud fra en teoretisk sampling, hvor data bearbejdes gennem begreber og teorier som allerede er udviklet (Brinkmann & Tanggaard, 2010, s. 227), hvilket blev gjort på baggrund af konstruktionismens læringsteori, som den fremlægges af Martinez & Stager (2013) (Elving, 2017). Interviewene blev foretaget med de fire ansvarlige lærere for forløbene og var opbygget som semistrukturerede interviews for både at opnå forståelse for hver case og dennes kontekst, for at give indsigt i de temaer, som havde vist sig relevante under de uformelle interviews, samt med henblik på at få adgang til de fortællinger, som lærerne selv fandt presserende (Brinkmann & Tanggaard, 2015, s. 37). Interviewene blev bearbejdet ud fra en datadreven kodning med afsæt i det empiriske datasæt og de temaer, der dukkede op i lærernes besvarelser (Brinkmann & Tanggaard, 2010, s. 47). En udfoldet argumentation for analyse og præsentation af kodningskategorier i relation til både interviews og observation kan findes i Elving (2017).

Coding Class-initiativet

Coding Class er et resultat af IT-Branchens samarbejde med blandt andet Styrelsen for IT og Læring samt Coding Pirates, hvor en række femte og sjette klasser på syv skoler i Vejle og København Kommune får it på skoleskemaet i et skoleår. IT-Branchen har i foråret 2016 indsamlet 1,5 millioner kroner, som blandt andet er gået til at frikøbe undervisere fra Coding Pirates, der skal understøtte faglærerne i inddragelsen af programmering i deres undervisning (IT-Branchen, 2017).

Casestudiet tager udgangspunkt i tre skoler fra Vejle Kommune, som har fået stillet to konsulenter til rådighed af IT-Branchen til at undervise i programmering under fire skoledage i løbet af skoleåret. Fagene og aktiviteterne i forløbet er blevet formet ud fra de enkelte skolars muligheder og lærernes idéer i dialog med konsulenterne. Omstændighederne for hver case beskrives nedenfor, men fælles for alle skolerne er, at spilprogrammering har udgjort udgangspunktet for forløbene, og det hovedsageligt har været konsulenterne, der har planlagt forløbene med et teknologisk frem for et didaktisk fokus. Det har således efterfølgende været op til de ansvarlige lærere at koble faglighederne på forløbene og sikre, at arbejdet med programmering havde en faglig relevans. Forholdet og rollefordelingen mellem konsulent og lærere kan med fordel undersøges nærmere, men artiklen vil herfra fokusere på lærerens oplevelse med at inddrage programmeringen i fagene med dette udgangspunkt. Nedenfor illustreres hver enkelt case med involverede lærere, eventuelle hjælperlærere, konsulenter og antal elever.



Figur 1: Oversigt over de tre skoler i casestudiet

Case 1:

Ansvarlig: Dansklærer. Elever: 28. Konsulenter: 2. Hjælpelærer: matematiklærer.

Forløbet foregik over seks uger i danskundervisningen, hvor en sjettede klasse blev fordelt i grupper, der skulle fungere som virksomheder i forløbet. Hver gruppes virksomhed skulle skabe et spil i Scratch, holde et fiktivt budget samt reklamere for deres virksomhed og spil, og de skulle påtage sig forskellige roller som projektleder, grafiker og udvikler mm. Ud over de fire afsatte dage, hvor konsulenterne fra Coding Class assisterede i undervisningen, arbejdede klassen videre med deres projekter både i dansk og matematik. Eleverne blev introduceret til programmering før forløbets start, hvor de fik til opgave at udføre basale programmeringsopgaver i Scratch samt udføre opgaver, der introducerede dem til computational thinking. Faglæreren, der stod for undervisningen var dansklærer, havde ingen erfaring med programmering men blev assisteret af årgangens matematiklærer, der også var involveret i forløbet. Elevgruppen bestod af tre sammenlagte sjettede klasser, hvor læreren havde inddelt de 28 elever i grupper á fire, og eleverne var selv ansvarlige for at fordele rollerne i virksomheden mellem sig.

Case 2:

Ansvarlig: Matematiklærer. Elever: 17. Konsulenter: 2. Hjælpelærer: ingen.

Forløbet foregik over fire dage i en enkelt femte klasse, hvor 17 elever blev opdelt i grupper, der skulle fungere som virksomheder i forløbet, hvor hver virksomhed designede spil i Sploder, lavede en app, holdt et fiktivt budget for virksomheden og reklamerede for deres arbejde i virksomhederne. Programmeringsaktiviteterne blev brugt løbende over hele skoleåret, når det passede ind både i fagene matematik og natur/teknik. Faglæreren havde ikke tidligere arbejdet med programmering i undervisningen men havde stor interesse herfor, hvorfor han havde eksperimenteret med det løbende i undervisningen i fællesskab med eleverne.

Case 3:

Ansvarlig: 2 matematiklærere. Elever: 60. Konsulenter: 2. Hjælpelærer: 3 lærere (dansk, matematik).

Forløbet foregik i matematikundervisningen, hvor de fire undervisningsdage med Coding Class blev fordelt over nogle få uger. Her skulle 60 elever fra sjettede årgang udvikle spil i Scratch i selvvalgte grupper. De arbejdede udelukkende med programmering på de fire dage, hvor konsulenterne var til stede, men matematiklærerne havde efter det egentlige forløb med Coding Class yderligere planlagt en ekstra dag, hvor sjettede årgang skulle introducere elever fra fjerde og femte årgang i programmering. På femtedagen var elever fra ni klasser på mellemtrinnet således involveret i programmeringsaktiviteter i grupper på tværs af årgange og klasser, kun assisteret af to matematiklærere, tre hjælpelærere og de to konsulenter fra Coding Class.

Dilemmaer ved inddragelse af programmering i undervisningen

Følgende analyse belyser lærernes erfaringer med at inddrage programmering i deres undervisning, herunder deres vurdering af programmingsaktiviteternes faglige relevans, samt hvilke faktorer der påvirker læringssituationen. Gennem datakodning af lærerinterviewene fra casestudie blev flere tendenser i lærernes erfaringer med forløbene synlige, og en række dilemmaer i lærernes arbejde med programmering i undervisningen trådte frem (Elving, 2017). Dilemmaerne omhandler blandt andet lærernes vægtning af at bruge programmering som et fagspecifikt eller tværfagligt element, elevernes faglige udbytte overfor deres interesser samt af lærerens rolle som enten ekspert i klasseværelset eller pædagogisk guide. Disse dilemmaer gennemgås herunder og udmunder i en række opmærksomhedspunkter, der kan vejlede fremadrettet arbejde med programmering i undervisningen på baggrund af erfaringerne fra casestudiet.

Fagspecifikt kontra tværfagligt element

Lærernes oplevelser med at inddrage programmering som fagligt element i deres fag varierede på tværs af de tre cases. Lærerne ønskede at opfylde nogle af fagenes mål gennem inddragelsen af programmering, men det vekslede, hvorvidt og herunder hvordan og i relation til hvilke fag målene blev opfyldt. Det var især de danskfaglige mål, som blev opfyldt i både Case 1 og 2 gennem designorienterede forløb med spilprogrammering i deres opfundne virksomheder. I Case 1 var forløbet koblet på danskundervisningen, men læreren erfarede dog løbende, at det ikke var selve programmeringsaktiviteterne men derimod de omkringliggende aktiviteter uden for selve koden, der opfyldte de danskfaglige mål. Læreren fremhævede her aktiviteter, der involverede feedbackprocesser, gruppearbejde, projektarbejde og mundtlige præsentationer og påpegede, at selve programmeringen var mere relevant for matematikfaglige mål. Læreren i Case 1 havde haft større ambitioner om at knytte programmeringsaktiviteterne til danskfaget men oplevede begrænsninger herved grundet egne evner:

“Vi har fået koblet nogle danskfaglige mål ind i det, men ikke i den grad som jeg gerne ville, men det tror jeg handler om, at jeg ikke har kundskaberne i kodningen endnu, til at kunne tænke i de baner.”

I dette tilfælde blev det lærerens kompetencer, der havde indflydelse på, hvordan programmeringsaktiviteterne kunne kobles til de fagfaglige mål. De omkringliggende aktiviteter opfyldte læringsmålene, men hendes egne kompetencer forhindrede hende i at bruge programmering som tiltænkt. Hun så dog fortsat en relevans i at inddrage programmering i relation til danskfaget både som tværfagligt

forløb med matematik men ligeledes mere direkte ind i forløb med danskfaglige temaer og mål i centrum for programmeringsaktiviteterne. I Case 2 havde læreren lettere ved at koble danskfaglige mål på forløbet end på de naturvidenskabelige fag, som forløbet egentligt var knyttet til, hvorfor det her, ligesom i Case 1, blev et tværfagligt forløb mellem især dansk og matematik. De faglige mål havde dog mindre betydning for læreren i Case 2 end i de resterende cases, da han oplevede en større relevans i, at eleverne tilegnede sig generelle kompetencer, som kan bruges tværfagligt. Læreren oplevede især at elevernes problemløsning blev styrket:

“Jeg synes det er sjovt, og jeg synes det er givende for eleverne. Jeg synes de er gode til at tage imod det, og jeg synes de får meget ud af det. Deres måde at tænke på, fx hvis de har taget fejl, så retter de bare deres fejl. Det kan de også tage med i andre fag.”

Forløbet i Case 2 fungerede således som et tværfagligt forløb med fokus på tilegnelse af generelle kompetencer, der er relevante på tværs af fag. I Case 3 var forløbet udelukkende en del af matematikundervisningen, men lærerne oplevede ikke, at programmeringen bidrog til indfrielse af fagets fagspecifikke mål, men de mente dog alligevel, at eleverne lærte noget og havde det sjovt med aktiviteterne. Lærerne i Case 3 så dog en fordel ved at bruge programmering som redskab til at arbejde med matematiske temaer, da den ene lærer havde kendskab hertil gennem deltagelse på et kursus specifikt målrettet kobling mellem programmering og matematikfagets temaer og mål. De ønskede ikke fremadrettet at arbejde med programmering i matematikundervisningen på samme måde som under Coding Class-forløbet. De havde dog fået inspiration fra forløbet og havde allerede planlagt inddragelse af programmering i deres næste undervisningsforløb i matematik, som fremadrettet blot skulle kobles direkte på de faglige mål i faget.

Det var således en udfordring for lærerne at balancere brugen af programmering i undervisningen mellem et middel til at understøtte henholdsvis elevernes fagspecifikke kompetencer, som det ønskes af lærerne i Case 1 og 3, eller mere generelle kompetencer som i Case 2. Lærerne i Case 1 og 3 udtrykte et ønske om, at inddragelsen af programmering i deres undervisning skulle kunne indfri fagspecifikke mål, men at deres manglende viden og kompetencer inden for området begrænsede deres evner til at gøre netop dette. Lærerne i Case 1 og 2 påpegede dog, at indfrielsen af de fagspecifikke mål godt kan ske i aktiviteterne omkring programmeringen, hvis det indtænkes som del af undervisningsforløbet og eventuelt som et tværfagligt forløb. I Coding Class var forløbene udformet som selvstændige designorienterede projektføløb, hvor eleverne skulle arbejde mere eksplorativt og ud fra deres egne interesser samt idéer og ikke direkte designet til opfyldelse af fagspecifikke mål.

Elevernes faglige udbytte kontra elevernes interesse

På tværs af de tre cases udtalte lærerne sig alle positivt omkring elevernes koncentration, engagement og interesse for deres produkter under forløbene. Eleverne i alle tre cases var i høj grad optagede af deres arbejde med programmeringsopgaverne uafhængigt af, hvordan forløbets opbygning var struktureret. Lærerne lagde især vægt på, at eleverne havde et fælles udgangspunkt for at arbejde med programmering, hvor forskellen mellem de fagligt stærke og de fagligt svage elever derfor blev mere udlignet under forløbene. Lærerne understregede dog, at det er en udfordring, at elevernes udbytte af forløbet især afhæng af, hvorvidt eleverne udviklede en interesse for programmeringsaktiviteterne eller ej. I alle cases oplevedes stort engagement blandt eleverne, men gruppesammensætningen havde ifølge lærerne en stor påvirkning på elevernes udbytte af aktiviteterne. I Case 1 og 2 var der en tendens til, at elever, der enten var dygtige eller interesserede i at kode, havde størst berøring med programmeringen, mens de andre elever i gruppen i højere grad fokuserede på eksempelvis det visuelle udtryk i deres produkter eller skriftlige formuleringer om deres arbejde. Lærerne oplevede dette som en problemstilling, da det medførte at elevernes læringsudbytte varierede, fordi eleverne fordelte rollerne i grupperne afhængig af interesse og kompetence, hvorfor eleverne, som havde mindst interesse eller sværest ved at programmere ikke fik særlig stor erfaring med programmeringsopgaverne i grupperne. På baggrund af dette ønskede læreren i Case 1 fremadrettet at lave mindre og eventuelt kønsopdelte grupper for at sikre, at alle eleverne får samme berøringsgrad med koden, da hendes erfaring var, at det ofte var drengene, der overtog kodningen i grupperne. I Case 3 havde gruppesammensætningen ligeledes betydning, men det bundede i højere grad i, at eleverne selv var ansvarlige for gruppeinddeling. De havde derfor inddelt sig i grupper alt efter hvem, de var venner med, hvilket medførte koncentrationsbesvær i visse grupper, der havde en tendens til at snakke og forstyrre hinanden frem for at arbejde. Disse gruppers læringsudbytte var derfor, ifølge læreren, mindre end nogle af de andre gruppers.

På tværs af de tre cases så lærerne altså en tendens til, at gruppesammensætningen samt rollefordeling påvirkede elevernes læringsudbytte. I Case 1 påpegede læreren, at det ofte er sådan "*[...] i de fleste undervisningsforløb, at nogle får mere ud af det end andre, men nogle har også fået mere ud af det end vi havde forventet*". Hun ønskede dog fremadrettet at indlede med et længere introduktionsforløb til programmering og dermed give eleverne en bedre basis for at kunne programmere og samtidig skabe lige forudsætninger for at arbejde med selve koden ude i grupperne. Læreren i Case 2 oplevede ligeledes, at elevernes udbytte af forløbet varierede, men han lagde vægt på, at det ikke handlede om fagligt svage eller stærke elever men om interesse: "*Når de kommer til det her, der starter de tit samme sted. Og så er der bare nogle der får en interesse for det, og andre gør ikke*". Elevernes udbytte af

undervisningen handlede ifølge ham om, hvorvidt eleverne fik interesse for programmering og de aktuelle opgaver, og han oplevede ligeledes, at eleverne blev motiveret af en høj succesrate med programmeringen, og at dette skabte gode produkter. Læreren havde større fokus på, at eleverne fik succesoplevelser med programmering frem for et stort fagspecifikt læringsudbytte. Han valgte derfor at lade dem arbejde med nemmere programmer og platforme, hvor elevernes succesrate med programmering var høj og produkterne blev gode, så at deres interesse og engagement dermed blev fastholdt. Elevernes designorienterede arbejde i virksomhederne i Case 2 blev således vægtet højere end deres faglige udbytte, hvorfor elevernes læring især blev påvirket af, hvad de selv fandt interessant og brugbart at opsøge i relation til deres projekter. I Case 3 ønskede lærerne i højere grad fremadrettet at bruge programmering i tæt relation til normale undervisningsaktiviteter i matematik, hvor eleverne selv skal kunne vælge mellem at arbejde med fastlagte opgaver i eksempelvis Scratch eller med traditionelle redskaber som pen, papir og lommeregner. Lærerne ønskede på denne måde at benytte programmering som en måde at ramme flere elevers interesse i undervisningen:

“Det passer mig rigtig godt at gøre det på den måde, at man netop finder nogle, der ikke fanges af papir, men som fanges af det andet [programmering]. Så får man begge med, så de er interesseret i det der foregår, på den måde, som de bliver fanget af det.”

Lærerne i Case 3 udtrykte således en fordel ved at bruge programmering i matematikfaget til at veksle i undervisningsaktiviteterne samt til at fange eleverne og sikre deres faglige udbytte omkring de fastlagte opgaver i matematik. Her ønskede de at bruge programmering i lukkede opgaver, eksempelvis opgaver med vinkler, grafer samt geometriske former. De så ikke relevansen i at arbejde med programmering i relation til designorienterede, kreative og eksplorative opgaver i matematikfaget. Lærerne stod således overfor et dilemma om på den ene side at vægte elevernes interesse og dermed skabe motivation blandt dem og på den anden side at styre deres arbejde til at sikre det fagspecifikke udbytte. Ifølge lærerne blev elevernes udbytte også påvirket af, hvor stor frihed de fik i forhold til gruppesammensætning, rollefordeling heri, samt hvor meget eleverne individuelt arbejdede med programmeringen. Dette er et dilemma, der afhænger af hvorledes undervisningsforløbet udformes og herunder, om det indgår som tværfagligt forløb med fokus på generelle kompetencer eller om det kobles på et enkelt fags mål og temaer med elevernes faglige viden og kompetencer for øje. Det skal derudover noteres at helt nye og anderledes aktiviteter kan have en positiv indvirkning på elevernes interesse og motivation i undervisningen, og at disse korte perioder ikke nødvendigvis viser, i hvor høj grad eleverne ville fortsætte med at være grebet af programmeringsaktiviteterne, hvis det blev en fast del af deres undervisningsforløb.

Alvidende underviser kontra pædagogisk guide

Det varierede på tværs af de tre cases, hvor stor berøring lærerne havde med selve programmeringen under Coding Class. I Case 1 og 3 overlod lærerne programmeringen til konsulenterne fra IT-Branchen, der hverken var læreruddannede eller havde pædagogisk erfaring, men som derimod havde de tekniske kompetencer, som lærerne selv manglede. Læreren i Case 1 ønskede som udgangspunkt at lære at kode selv, men der viste sig ikke at være tid til dette, hvorfor læreren følte sig afhængig af konsulenternes tilstedeværelse i undervisningen. Hun havde forventet og håbet på en større personlig kompetenceudvikling og mente, at hun havde behov for videreuddannelse for at undervise i programmering:

“Jeg ville godt kunne gå ind og hente en kollegas forløb om Astrid Lindgren og bruge det og omsætte det og få det passet ind til min undervisning og til mine elever. Men hvis der lå et kodningsforløb, og jeg igen ikke har den basisviden, så ville jeg stadigvæk komme til kort. Der ville jeg blive nødt til at hente en ressourceperson ind, som kunne hjælpe mig, medmindre jeg kunne komme på kursus i det selv.”

Hendes egne kompetencer blev således en barriere for at tage styringen over selve programmeringsarbejdet i undervisningen under forløbet. I Case 3 havde lærerne heller ingen erfaring med programmering og havde ingen berøring med koden undervejs i forløbet og overlod derfor denne del til konsulenterne og eleverne selv. Lærerne valgte at lade konsulenterne planlægge undervisningsforløbet *“for vi ved jo ikke hvad der var af muligheder. Så vi tænkte, at det ved de noget om”*. De betragtede Coding Class-forløbet som inspiration til deres matematikundervisning, så de kunne blive klogere på, hvordan programmering kunne benyttes i fremtidige undervisningsforløb i faget. Både i Case 1 og 3 var lærernes rolle derfor mere pædagogiske frem for ekspert, og de var derfor afhængige af tilstedeværelsen af konsulenterne og brugen af ældre elever i undervisningen til at håndtere den tekniske kodning, som de ikke følte sig kompetente til at påtage sig. I Case 2 havde læreren derimod større praksiserfaring med programmering og hjalp sammen med konsulenterne eleverne med deres kode. Læreren i Case 2 udtrykte stor begejstring for teknikken og valgte derfor at tilegne sig praktisk erfaring med koden og de forskellige programmer ved at indgå aktivt i programmeringsaktiviteterne. Læreren i Case 2 var den eneste af lærerne i hele casestudiet, som havde en personlig interesse for programmering og som indgik i en fælles læringsproces og problemløsning med eleverne. Selv om læreren var passioneret omkring programmering, udtrykte han dog stadig en udfordring i, at han ofte manglede kompetencerne til at hjælpe eleverne. Under observationerne virkede det dog ikke til at genere eleverne, at læreren indgik i en fælles problemløsning og læreproces med dem. Indtrykket var det samme for de yngre elever i Case 3, der ikke virkede

generede af at de ældre elever ikke kunne hjælpe med alt, men ligeledes indgik i en problemløsning sammen med dem, når komplekse problemer opstod. Brugen af andre elever som undervisere i Case 3 bevirkede derimod, at de yngre elever i høj grad søgte hjælp ved hinanden, da der ikke var nogen forventning om, at de ældre elever kunne hjælpe dem med alt. Det understøttede altså her elevernes interne samarbejde og problemløsning, at de skulle prøve sig frem i stedet for at regne med, at underviserne havde svaret på deres spørgsmål. Dette viser, at underviserne ikke behøvede at være alvidende eksperter for, at eleverne kunne opnå en progression i deres læreproces og deres arbejde med programmeringsopgaverne. Det var en udfordring for lærerne i Case 1 og 3, da de har været vant til at indgå som eksperter i undervisningen. Lærerne følte sig altså ikke kompetente til at arbejde med programmering i deres undervisning, selvom eleverne i praksis ikke virkede berørte af, at lærerne eller de ældre elever indgik i en fælles læreproces med dem. Læreren i Case 2 var klar over, at eleverne hurtigt blev bedre end ham til at kode, og han derfor ofte ikke altid kunne hjælpe dem, men han beskrev dette som en forudsætning for at bruge programmering i sin undervisning:

“Når man arbejder med programmering, bliver du også nødt til at arbejde innovativt på en eller anden måde. Så du præsenterer dem [eleverne] for noget, hvilket ofte er en åben opgave, hvor du ikke ved hvordan den skal løses.”

Han lagde vægt på, at det er krævende for en lærer at arbejde med programmering, da der ofte ikke er noget endegyldigt facit, og at elevernes arbejde altid udmunder i noget uforudsigeligt og individuelt. Han gav ydermere udtryk for, at dette kan være et stressende element for visse lærere men foreslog, at læreren kan vælge at udforme mere lukkede opgaver, hvis vedkommende foretrækker at kende facit. Den personlige interesse og passion for programmering kan have haft indflydelse på, hvorvidt læreren i Case 2 ignorerede begrænsningen i egne evner og derfor indgik i en fælles læreproces sammen med eleverne i det eksperimenterende programmeringsarbejde. Der kan således være en udfordring i lærernes tilgang til programmeringsarbejdet og deres forventninger til egne evner og rolle i klasseværelset.

Pædagogiske overvejelser ved inddragelse af programmering i undervisningen

På baggrund af analysen af lærernes erfaringer med at inddrage programmering i deres undervisning og hvilke udfordringer, dette medførte i praksis, udformes en række opmærksomhedspunkter til planlægning af fremadrettede forløb. Disse opmærksomhedspunkter er vejledende og kan støtte skoleledere og lærere i at træffe bevidste til- og fravalg, når de fremadrettet ønsker at arbejde med

programmering i undervisningen i grundskolen. Opmærksomhedspunkterne indeholder blandt andet en vægtning af elevernes frihed til at opsøge det, de selv finder interessant, og lærerens behov for at styrke elevernes fagspecifikke udbytte. Samtidig bør det overvejes, om programmering skal indgå som tværfagligt forløb, hvor generelle kompetencer understøttes, eller kobles tæt sammen med de fagspecifikke krav og relatere sig til det enkelte fags fastlagte temaer og mål. Det er ligeledes relevant for elevernes udbytte, at lærerne overvejer gruppesammensætningen, herunder antallet af elever, en eventuel kønsopdeling og graden af indledende og introducerende individuelt arbejde, før det kreative projektarbejde i grupper påbegyndes. Lærerne skal samtidig være bevidste om deres egen rolle, der befinder sig i spændingsfeltet mellem ekspert, som kan hjælpe eleverne med den tekniske kodning, og en pædagogisk rolle, som kan guide elevernes læreproces.

Opmærksomhedspunkter til inddragelse af programmering i undervisningen:

- **Skal programmering indgå som element i ét enkelt fag eller anvendes i et tværfagligt forløb, og i så fald med hvilke fag involveret?**

Et tværfagligt forløb kan give anledning til at bruge programmering som del af et større projekt, eksempelvis gennem forløb med opbygning af virksomheder, hvor eleverne blandt andet programmerer spil, fører regnskab og laver merchandise og PR. Ønskes en tæt kobling til de faglige mål, kan det være en fordel at begrænse forløbet til et fag.

- **Skal forløbet fange elevernes interesse og udvikling af generelle kompetencer eller kobles direkte til opfyldelse af fagspecifikke mål?**

Hvis programmeringsaktiviteterne skal kobles direkte til de fagspecifikke krav, kan det overvejes at opstille lukkede opgave, hvilket dog kan mindske elevernes engagement samt eksperimenterende og kreative udfoldelse.

- **Skal eleverne have lige stor erfaring med kodningen, eller må de frit fordele arbejdet i grupperne?**

Hvis alle elever skal i berøring med koden, kan det være en fordel at, eleverne indledningsvis arbejder individuelt med programmering gennem eksempelvis tutorials, førend de arbejder kreativt i grupper. Alternativt kan læreren opstille krav til rolle- og opgavefordelingen for at sikre, at alle kommer i berøring med koden. Overvej også, om eleverne selv skal være ansvarlige for gruppeinddelingen, da gruppesammensætningen og antallet af elever kan have indflydelse på, hvor tæt berøring eleverne får med programmeringen.

- **Skal du som lærer indgå som ekspert i programmeringsopgaverne eller guidende underviser for elevernes egen læreproces?**

For at indgå som ekspert i undervisningen kræver det praktisk erfaring med programmering og en villighed til at prøve sig frem sammen med eleverne. For at indgå som pædagogisk guide kræver det derimod en accept af, at man som lærer ikke kan hjælpe eleverne med alt. Her kan adgang til det rette materiale, fx tutorials på nettet, udgøre den nødvendige hjælp.

Disse punkter bør tilpasses den konkrete læringssituation og lærernes egne præferencer. Punkternes formål er at give lærere nogle konkrete overvejelser, der kan bidrage til planlægningen af konkrete undervisningsforløb med brug af programmering.

Overvejelser omkring lærerens rolle i klasseværelset

At arbejde med programmering i undervisningen kan ikke udelukkende begrænses til fire opmærksomhedspunkter til planlægning af det konkrete undervisningsforløb, men bør ligeledes involvere overvejelser omkring, hvorledes lærerens rolle skal tænkes og håndteres fremadrettet. Af den grund gør vi os herunder nogle overordnede overvejelser omkring lærerens rolle og eventuel videreuddannelse af lærerne på baggrund af teoretiske perspektiver samt erfaringerne fra casestudiet.

I casestudiet var det kun læreren i Case 2, der indgik i en fælles læreproces med eleverne og havde det sjovt med at programmere, hvor lærerne i Case 1 og 3 forholdt sig mere passive grundet manglende tekniske kompetencer. I den konstruktionistiske læringsteori, som ofte bruges som inspiration til programmeringsforløb, skal læreren dog ikke nødvendigvis have alle svarene på forhånd men derimod agere guide for elevernes læreproces og at indgå i fælles problemløsning med dem (Martinez & Stager, 2013, s. 71). Målet med undervisningen i konstruktionismens perspektiv er nemlig, at de lærende er i centrum, hvor læreren skal give eleverne mulighed for at lære på egen hånd og dermed begrænse mængden af instruktion. Lærerens rolle er således ikke at løse elevernes problemer, men at understøtte eleverne i selv at rette deres arbejde og prøve sig frem, indtil problemet løses (Martinez & Stager, 2013, s. 77). Det er derfor ikke nødvendigt for en lærer at være den alvidende underviser men derimod at være en pædagogisk støtte i elevernes egen problemløsning og læreproces. Lærerne i casestudiet havde dog problemer med netop at indgå i en fælles læreproces med eleverne i Case 1 og 3, da lærerne ikke gik ind i programmeringen, selvom de alle ønskede at forbedre deres tekniske kompetencer gennem videreuddannelse. Denne villighed til at forbedre sig oplevede Papert ligeledes i sit arbejde med at indarbejde it og programmering i grundskolen siden 1970'erne. Han erfarede, at lærerne ønskede at opnå større viden og kompetencer for at inddrage programmering i deres undervisning, men visse forventninger til lærerens rolle udgjorde en forhindring (Papert, 1993). Lærerne fik ifølge Papert ikke mulighed for at videreudvikle sig og indgå i en fælles læreproces med eleverne grundet ydre

forventningerne om lærerens rolle i klasseværelset: *“many aspects of School block teachers from the fulfillment of functioning in a class as co-learners”* (Papert, 1993, s. 67). I casestudiet ses det også, hvordan lærernes egne forventninger om egen rolle kan være en blokering for, at de kan indgå som lærende i egen undervisning, hvorfor lærerrollen både fastholdes af eksterne personer og konventioner på eksempelvis skolen, i klasserummet, i fortællinger omkring læreren samt af læreren selv. Problemstillingen løses derfor ikke nødvendigvis gennem videreuddannelse, da Paperts erfaringer hermed er, at lærere er tilbageholdende:

“their awareness of being teachers was preventing them from giving themselves over fully to experiencing what they were doing as intellectually exciting and joyful in its own right, for what it could bring them as private individuals.” (Papert, 1993, s. 72).

Dette perspektiv ses ligeledes i casestudiet, hvor lærerne på tværs af de tre cases var meget bevidst omkring udfordringerne i forhold til deres egne kompetencer. De var dog alle opsatte på at få kompetenceudvikling og relevant videreuddannelse for i fremtiden at kunne anvende programmering i deres undervisning. Udgangspunktet for lærerne i deres ønske om videreuddannelse var, at de ligesom i deres normale fag skal kunne indgå i klasseværelset med en ekspertrolle, hvor de altid er i stand til at hjælpe eleverne. De ønskede derfor at få indsigt i konkrete muligheder for at inddrage programmering specifikt i forhold til den fagfaglige undervisning samt at få overblik over de konkrete muligheder, eksempelvis programmeringssprog og -programmer. Hvis programmering skal indgå i eksisterende fag, er det således vigtigt at uddannelse af lærerne er anvendelsesorienteret, hvor fagene indtænkes og lærerne får praktisk erfaring med programmering. Det er dog ligeledes vigtigt, at lærerne opfordres til at prøve sig frem og indgå i en fælles læreproces med eleverne uden et krav om, at de skal være den alvidende underviser. Dette kræver dog en ændring af en kultur og fortælling omkring lærerens rolle, hvilket kan være omfattende at realisere.

Opsamling og konklusion

I artiklen undersøges, hvordan programmering benyttes i en pædagogisk praksis i grundskolen, samt hvilket perspektiv faglærerne har på at inddrage programmering som fagligt element i deres undervisning. Analyse af lærernes erfaringer med programmeringsaktiviteterne viste overordnede tendenser på tværs af de tre cases, hvor en række dilemmaer i lærernes arbejde trådte frem:

Lærerne oplevede højt engagement omkring og fordybelse i programmeringsarbejdet blandt eleverne, men flere af lærerne manglede større kontrol over elevernes fagspecifikke udbytte, hvorfor lærernes faglige målstyring i

undervisningen skal balanceres over for at give eleverne frihed til, at deres læring styres af egne interesser. Lærerne fandt det relevant at inddrage programmering i deres undervisning på grund af elevernes udbytte under forløbet, men det varierede, hvorvidt udbyttet var tilknyttet fagspecifikke mål eller mere generelle kompetencer. Lærerne oplevede, at elevernes udbytte især påvirkedes af elevernes egne interesser, gruppesammensætninger samt rolle- og opgavefordeling i grupperne, hvorfor lærernes styringsgrad heraf ligeledes bliver relevant i forhold til design af undervisningsforløb med brug af programmering. Lærerne i casestudiet oplevede derudover alle begrænsninger i egne kompetencer, men det varierede, hvorledes de tilgik programmeringsarbejdet og opfattede deres egen rolle som lærer i klasseværelset. Disse dilemmaer i lærernes erfaringer fra casestudiet blev behandlet og formuleret som konkrete pædagogiske overvejelser til planlægning af fremtidige undervisningsforløb med programmering, og involverer følgende konkrete overvejelser:

- Skal programmering indgå som element i ét enkelt fag eller anvendes i et tværfagligt forløb, og i så fald med hvilke fag involveret?
- Skal forløbet fange elevernes interesse og udvikling af generelle kompetencer eller kobles direkte til opfyldelse af fagspecifikke mål?
- Skal eleverne have lige stor erfaring med kodningen, eller må de frit fordele arbejdet i grupperne?
- Skal du som lærer indgå som ekspert i programmeringsopgaverne eller guidende underviser for elevernes egen læreproces?

Undersøgelsen bidrager således ikke med generaliseringer om, hvordan programmering bør indgå i grundskolen, men kan bidrage med nogle pædagogiske overvejelser til læreres fremadrettede arbejde med at inddrage programmering i deres undervisningsforløb. For at lærerne i undersøgelsen kunne indarbejde programmering som et fagligt meningsfuldt element i deres undervisning, ønskede de kompetenceudvikling og videreuddannelse samt at programmeringsaktiviteterne knyttedes til fagenes temaer og mål. Undersøgelsen peger dog ligeledes på, at lærerens rolle i klasseværelset kan udfordres og herved åbne op for, at læreren kan indgå i en fælles læreproces med eleverne. Særligt i forbindelse med sammenhængen mellem programmering og de faglige mål kan undersøgelsen også antyde, at det kan være at lave en kraftig opdeling mellem fx programmeringskonsulenter og lærere, hvor konsulenter designer forløbene og lærerne så efterfølgende "på sætter" målene. Derimod skal man være opmærksom på, at der skal være nogle stærke og velovervejede koblinger mellem de dimensioner, der med fordel kan tænkes ind i designet af forløbene.

Et forslag til videre arbejde på baggrund af undersøgelsens resultater kan omhandle, hvordan programmering kan inddrages i undervisningsforløb i henholdsvis dansk

og matematik samt i kobling mellem disse fag. I undersøgelsen gjorde to ud af de tre cases meningsfuld brug af programmering i et tværfaglig forløb mellem netop disse fag, hvorfor der ses et potentiale i at arbejde videre med koblingen mellem naturvidenskabelige og humanistiske fag gennem programmeringsaktiviteter. Dette har ikke været et stort fokus i forskningsfeltet, men undersøgelsen indikerer et potentiale hermed, hvilket kan studeres nærmere og konkrete undervisningsforløb kan eventuelt designes herudfra.

Referencer

- Blomberg, J., Giacomi, J., Mosher, J., & Swenton-Wall, P. (1993). Ethnographic field methods and their relation to design. In D. Schuler, & A. Namioka (Eds.), *Participatory design: Perspectives on systems design* (s. 123-154). Hillsdale: Lawrence Erlbaum Associates, Inc.
- Brinkmann, S. & Tanggaard, L. (2010). Towards an epistemology of the hand. *Studies in Philosophy and Education*, 29(3), 243-257.
- Danmarks Vækstråd. (2016). *Rapport om kvalificeret arbejdskraft*.
- Ejsing-Duun, S., & Misfeldt, M. (2015). Programmering af robotenheder i grundskolen. *Læring Og Medier (LOM)*, 8(24).
- Elving, R. E (2017). PROGRAMMERING I GRUNDSKOLEN: Et casestudie om erfaringer fra pilotprojektet Coding Class og læringspotentialet for programmering i undervisningen. Kandidatspeciale, Aalborg Universitet.
- Flyvbjerg, B. (2010). Fem misforståelser om casestudiet. In L. P. Tanggaard, & S. Brinkmann (Eds.), *Kvalitative metoder: En grundbog*. (463-488). København: Hans Reitzels Forlag.
- Foerster, K. (2016). Integrating programming into the mathematics curriculum: Combining scratch and geometry in grades 6 and 7. *Sigite*, 91-96.
- Hansbøl, M. Ejsing-Duun, S. (2017). *Hovedrapport: Coding Class. Dokumentation og evaluering. IT-Branchen*. Lokaliseret d. 7 jan. 2018 på: [https://itb.dk/sites/default/files/Rapport %20 Coding%20Class%20-%20Dokumentation%20og%20evaluering%20Endelig.pdf](https://itb.dk/sites/default/files/Rapport%20Coding%20Class%20-%20Dokumentation%20og%20evaluering%20Endelig.pdf)
- IT-Branchen. (u.å.1). *Coding Class*. Lokaliseret 27. marts 2017 på: www.itb.dk/articles/fremtidens-kompetencer-coding-class.
- IT-Branchen. (u.å.2). *Sikring af fremtidens kompetencer*. Lokaliseret 27. marts 2017 på: www.itb.dk/fremtidens-kompetencer.
- IT-Branchen. (2017). *Støt op om Coding Class*. Lokaliseret 5. april 2017 på: www.itb.dk/news/fremtidens-kompetencer/stot-op-om-coding-class.

- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.ord. *Computers in Human Behavior*, 52, 200-210.
- López, J., González, M., & Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools. *Computers & Education*, 97, 129-141.
- Martinez, S. L., & Stager, G. (2013). *Invent to learn: Making, thinking and engineering in the classroom*. Torrance, Calif: Constructing Modern Knowledge Press.
- Nielsen, J., Pedersen, R., & Majgaard, G. (2015). 8. klasse som kreative producenter af fremtidens velfærdsteknologi - konstruktionisme, problemløsning og dialog. *Læring & Medier (LOM)*, 14.
- Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer* (2. edition ed.). United States: BasicBooks.
- Stake, R. E. (2005). Qualitative case studies. In K. B. Denzin, & Y. S. Lincoln (Eds.), *The SAGE handbook of qualitative research* (s. 443-466) Sage Publications.
- Stake, R. E. (1995). *The art of case study research*. United States: Sage Publications.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Zaharija, G., Mladenovic, S., & Boljat, I. (2013). Introducing basic programming concepts to elementary school children. *Procedia - Social and Behavioral Sciences*, 106, 1576-1584.
- Zainal, Z. (2007). Case study as a research method. *Jurnal Kemanusiaan*, 9.