



COMPUTATIONAL THINKING

— HVORFOR, HVAD OG HVORDAN?

Michael E. Caspersen
Ole Sejer Iversen
Mogens Nielsen
Arthur Hjorth
Line Have Musaeus

Støttet af
VILLUM FONDEN



It-vest

samarbejdende universiteter

Computational Thinking

— hvorfor, hvad og hvordan?

Direktør Michael E. Caspersen, It-vest – samarbejdende universiteter

Professor Ole Sejer Iversen, Aarhus Universitet

Professor emeritus Mogens Nielsen, Aarhus Universitet

Postdoc Arthur Hjorth, Aarhus Universitet

Ph.d.-studerende Line Have Musaeus, Aarhus Universitet

© It-vest – samarbejdende universiteter, 2018.

Resumé

Digitalisering giver radikalt nye muligheder inden for alle fag – videnskabsfag, professioner og uddannelsesfag.

Bortset fra videregående it-specialistuddannelser har uddannelsesfokus hidtil været på it som en understøttende teknologi. Globalt er dette imidlertid under hastig forandring. Promoveret af begrebet 'Computational Thinking' (CT) er informatik internationalt i færd med at blive en del af almen uddannelse på alle niveauer; mange mener, at det i det 21. århundrede er en ligeså væsentlig grundlæggende kompetence som sprogfag og matematik.

Denne rapport beskriver visionen for informatik og CT og den computationelle revolution, der ligger til grund herfor. Baggrunden for udbredelsen af begrebet CT præsenteres og diskuteres i relation til beslægtede uddannelsesmæssige bevægelser.

Der gives eksempler på udbredelse af CT i erhverv og videnskab og på de nye, effektive og grænseoverskridende muligheder som computationelle metoder og teknikker giver.

Kernen i rapporten er dels en status for CT i uddannelse såvel internationalt som nationalt, herunder en beskrivelse af forskningsaktiviteter og de mange åbne spørgsmål, der mangler gode svar på, dels en beskrivelse af potentialet for CT. Især fremhæves den vigtige tostrengede strategi med informatik som en fundamental og selvstændig disciplin (*som fag*) og informatik/CT som en tværgående disciplin integreret i alle andre discipliner (*i fag*). Specielt er der fokus på muligheder og udfordringer i det danske uddannelsessystem.

Myndiggørelse af børn og unge ift. den computationelle revolution fordrer, at uddannelsessystemet gradvist, men til fulde, omfavner denne faglighed. Det er en meget stor opgave; til gengæld er potentialet enormt.

* * *

Tak til Villum Fonden for opfordring og støtte til udarbejdelse af dette notat og til Mark Guzdial, Leigh Ann DeLyser og Shuchi Grover, som har bidraget med anbefalinger af litteratur m.m.

Michael E. Caspersen

Ole Sejer Iversen

Mogens Nielsen

Arthur Hjorth

Line Have Musaeus

Aarhus, den 18. december 2018

Indhold

1. Vision: Hvorfor informatik og CT?	1
1.1 Den fjerde industrielle revolution?	2
1.2 Det fjerde videnskabelige domæne	3
1.3 Den fjerde sprogform	3
1.4 Den fjerde fundamentale kompetence	5
1.5 'Program or be programmed'	6
1.6 Automatisk udførelse af processer forandrer alt	7
2. CT – baggrund, karakteristik og relaterede begreber	10
2.1 Karakteristik og udbredelse af CT	10
2.2 CT og 'mathematical thinking'	13
2.3 CT og STEM	14
2.4 CT og makerspaces	15
3. Intermezzo om CT i erhverv og videnskab	17
4. CT i uddannelse – status	19
4.1 CT i international uddannelse	19
4.2 CT i dansk uddannelse	26
4.3 Teknologiforståelse	29
5. CT i uddannelse – potentialet	31
5.1 Informatics for All – en tostrengt strategi	31
5.2 Tentativ ramme for CT i universitetsuddannelser	32
5.3 Computationselle modeller af dynamiske systemer	32
6. CT i det danske uddannelsessystem – muligheder og udfordringer	34
6.1 Udvikling af curriculum og eksemplariske materialer	35
6.2 Uddannelse af undervisere	35
6.3 Forskning og fagdidaktisk indsigt	36
6.4 Udfordringer	38
7. Konklusion	40
8. Referencer	42
9. Appendikser	48
A.1 Computational Thinking – et vidt begreb	49
A.2 Constructionism og restructurations	51
A.3 Review af udvalgt forskningslitteratur om CT i uddannelse	54
A.4 Tentativ dansk taksonomi for CT i uddannelser	66

1. Vision: Hvorfor informatik og CT?

Afsnittet er en forholdsvis kort appetitvækker og vil bl.a. redegøre for visionen for informatik og CT, herunder visionen om at gøre dette til en del af almene digitale kompetencer og til en central del af målsætningen for det danske uddannelsessystem på linje med sproglære og matematik.

Verden bliver i stigende grad digitaliseret, og informatikken¹ bidrager med radikal og transformerende udvikling af professioner, videnskabelige discipliner og vores sociale liv i almindelighed.

Informatikken understøtter forskning, innovation og udvikling på tværs af alle sektorer og tilvejebringer radikalt nye og forbedrede muligheder for undervisning inden for alle fagområder og for uddannelse generelt.

De konkrete teknologier, der har set dagens lys gennem det sidste halve århundrede (f.eks. internettet og www), er kun begyndelsen; i de næste 15-20 år vil vi se dramatiske forandringer, der vil udvikle alle professioner samt skole- og videnskabsfag og give radikalt nye muligheder.

Som Klaus Schwab (stifter og leder af World Economic Forum) udtrykker det: "Hurtigere, end de fleste forestiller sig, vil professioner som advokater, advokatsekretærer, ejendomsmæglere, finansanalytikere, læger, revisorer, forsikringsmæglere, bibliotekarer og journalister blive delvist eller helt automatiserede".

Digitale teknologier udfordrer radikalt den måde, vi tænker, forstår og organiserer verden på, og de griber ind i vores samfundsstrukturer juridisk, økonomisk, demokratisk, sundhedsfagligt, videnskabeligt m.v. Personer, der ikke forstår præmisserne for digitaliseringen, vil i stigende grad være stillet som de, der for århundreder tilbage ikke kunne læse og skrive.

Det er vigtigt at omfavne de muligheder allerede fra en tidlig alder. I konsekvens heraf bør alle elever, studerende og lærere ikke blot have digitale brugskompetencer, men have viden, færdigheder og kompetencer inden for informatik og 'Computational Thinking' (CT).

Men hvorfor egentlig? Er informationsteknologien ikke bare en service, der bliver mere og mere transparent og integreret i stort set alt, hvad vi foretager os, og som vi blot skal koncentrere os om at forbruge med måde og omtanke? Hvorfor skal alle lære at kunne *udtrykke sig computationelt* med den digitale teknologi?

I det følgende giver vi (mindst) fire begrundelser; det overordnede synspunkt er, at vi betragter informatik/CT-kompetencer som færdigheder på lige fod med sproglige færdigheder og almene matematiske færdigheder: en grundbetingelse for at kunne begå sig i det 21. århundrede.

¹ 'Informatik' benyttes her som fællesbetegnelse for it-faglige videnskabelige discipliner.

1.1 Den fjerde industrielle revolution?

Der er i politiske kredse og den brede offentlighed en erkendelse af, at vi med digitaliseringen oplever en radikal teknologisk forandring, der bl.a. omtales som disruption og industri 4.0 – en industriel revolution, der startede i slutningen af det 18. århundrede og nu skulle have nået sit fjerde stadie.

Men det er ikke bare "endnu en teknologi" (som f.eks. kniven, dampmaskinen, telegrafen og antibiotika). Andre teknologier strækker menneskehedens fysiske formåen; men informatikken strækker vores mentale/kognitive formåen og åbner for radikalt nye muligheder.

Der er således ikke bare tale om endnu et skridt i den teknologiske udvikling. Som vi taler om revolutionen med Gutenbergs opfindelse af trykpressen i 1400-tallet og den industrielle revolution fra slutningen af 1700-tallet, kan man med rette tale om den digitale eller **computationelle revolution**, som vi først nu for alvor er ved at erkende.

Visionen for den computationelle revolution blev formuleret for 175 år siden af Ada Lovelace, idet hun forstod, at den analytiske maskine designet af hendes mentor, Charles Babbage, ikke bare kunne benyttes som programmerbar regnemaskine, men som generel informationsbehandlingsmaskine (Computer History, 2018)

Charles Babbage var optaget af *automatisering af kalkulation* med henblik på maskinel generering af matematiske tabeller. Ada Lovelace erkendte imidlertid et langt større potentiale i Babbage's analytiske maskine – *computation* i stedet for kalkulation – men et potentiale som det skulle tage mere end 100 år at realisere og næsten 200 år for os alle at indse.

Ada Lovelace indså, at maskinens tal kunne bruges til at indkode alt muligt andet end numeriske værdier, og udtrykte det således, eksemplificeret med automatisk komposition af musik (MIMI, 2018):

... it might act upon other things besides 'number', were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine.

Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.

In fact, the engine may be described as being the material expression of any indefinite function of any degree of generality and complexity...

Det næsten paradoksale, men dybt interessante, som Ada Lovelace erkendte, er, at informationsteknologi i sin rå form – computeren – ingenting kan, men vha. programmer kan bringes til at kunne næsten altting.

På samme måde som en Jacquardvæv intet kan uden mønsterkort, så kan computeren intet uden programmer. Men hvor en Jacquardvæv "kun" kan bringes til at væve, så kan en computer vha.

programmer (og avancerede ydre enheder) bringes til at udføre opgaver inden for håndværk, industri, forretning, videnskab, kunst osv.

Den digitale revolution muliggør *automatisering af abstraktioner* over næsten vilkårlige strukturer og processer inden for alle fag, brancher og videnskabelige domæner.

1.2 Det fjerde videnskabelige domæne

Informatik er et veletableret videnskabsområde, men flere og flere ser efterhånden informatik ikke bare som et af mange videnskabsområder, men som et fjerde videnskabeligt domæne på linje med teknik-, sundheds- og naturvidenskab, humaniora og samfundsvidenskab og som et fagfelt, der har stigende betydning for alle andre fagfelter.

Teknik-, natur- og sundhedsvidenskab er baseret på, at naturen kan forstås (måles og vejes); inden for disse fagfelter arbejder man med at forstå og manipulere med naturen. Inden for humanioras fagfelter er man optaget af at forstå og udvikle menneskets kulturprodukter og sprog. Inden for samfundsvidenskabens fagfelter er man optaget af at forstå og udvikle samfund og organisationer.

Fagfeltet informatik er baseret på, at verden – såvel den reelle som den imaginære – er 'computable', og man er optaget af at analysere, konstruere og forstå computationelle strukturer, processer, artefakter og systemer. Informatikken er et selvstændigt fagfelt, men med rige relationer til og implikationer for alle andre fagfelter (Rosenbloom, 2012; Tedre & Denning, 2017).

Denne erkendelse havde Nobelprisvinder i fysik Ken Wilson allerede i 1975, da han betegnede 'computation' som en tredje videnskabelig metode, der komplementerer teori og eksperiment (Denning, 2009, pp. 28-29):

As early as 1975, Physics Nobel Laureate Ken Wilson promoted the idea that simulation and computation were a way to do science that was not previously available. Wilson's Nobel Prize was based on breakthroughs he achieved in creating computational models whose simulations produced radical new understandings of phase changes in materials. In the early 1980s, Wilson joined with other leading scientists in many fields to advocate that the grand challenges of science could be cracked by computation and that the government could accelerate the process by supporting a network of supercomputing centers. They argued that computation had become a third leg of science, joining the traditional legs of theory and experiment. The term "computational thinking" was common in their discussions.

Simulering af fænomener og scenarier, som gennem computationel modellering gør abstrakte modeller både konkrete og eksekverbare, er under hastig udvikling som en tredje måde at drive videnskab på – og nogle gange er det den eneste holdbare, når teori er uhåndterlig, og eksperimenter er ugennemførlige. Vi vender tilbage til dette emne i afsnit 3.

1.3 Den fjerde sprogform

Man kan betragte informatikken som det fjerde trin i menneskehedens sproglige udvikling: (1) talesprog, (2) skriftsprog, (3) formelt sprog, specielt matematik og (4) computationelt sprog.

Computationelle sprog kan bl.a. benyttes til at beskrive kognitive processer, der kan udføres automatisk.

Skriftsprog er grundlaget for hele vores kultur. Science er videnskaben, og matematik er sproget, bag den teknologiske revolution. Informatik er videnskaben *og* sproget bag den computationelle revolution.

Hver ny sprogform muliggør, at ting, der før var komplekse og uden for almindelig fatteevne, kan gøres begribelige og bidrage til dannelse og kvantespring i civilisationens udvikling.

At vi alle har et skriftsprog, giver os en kraftfuld stemme i samfundet, således at vi på alle måder kan bidrage og agere med egne idéer, holdninger osv.

Tilsvarende med matematisk sprog. Simple forhold mellem tal kan beskrives i enkle ligninger, der kan læres i skolen, men som er svære eller umulige at forstå, når de er udtrykt i almindeligt skriftsprog. Her illustreret med et poetisk eksempel i form af et af Piet Heins vidunderlige græk:

It is indeed	
too odd	
for words	$\frac{1}{2} = \frac{3}{4} \cdot \frac{2}{3}$
that half's	
three quarters	
of two thirds	

Hvis det i stedet udtrykkes matematisk i form af en ligning, kan enhver, der er elementært matematisk skolet, straks verificere korrektheden af udsagnet. Men det omvendte gælder også: hvis man *ikke* er skolet, forstår man ikke en bjælde!

På samme måde giver informatikkens sprog mulighed for at simplificere beskrivelsen af vilkårlige såvel fysiske som mentale processer samt at "levendegøre" disse.

Eksempel: Matematisk modellering af bølger i form af koblede differentiaalligninger er voldsomt kompleks og utilgængelig for langt de fleste, men kan man lægge to tal sammen og dividere med to, kan man i et passende computationelt sprog lave en simpel og let tilgængelig digital bølgemodel, som børn og unge kan bruge til at undersøge avancerede begreber som amplitude, frekvens, interferens, plasticitet m.m.

På samme måde som bølgemodellen kan andre komplekse og dynamiske systemer repræsenteres ved hjælp af relativt simple og transparente computationelle modeller. Modellerne er langt nemmere at forstå end de tilsvarende matematiske modeller, og de har den ekstraordinære kvalitet, at de kan eksekveres/simuleres. Dermed inviterer de til "nærkontakt" i form af undersøgelse, justering af modellen ved simpel programmering, hypotesedannelse, eksperimenter, flere justeringer af modellen osv.

Informatikken som ny sprogform, og tilhørende udvikling af en ny form for 'literacy', behandles udførligt og visionært af Andrea diSessa i bogen *Changing Minds: Computers, Learning, and Literacy* (diSessa, 2000). I indledningen beskriver diSessa, hvorledes 'Computational Literacy' vil udvikle menneskers måde at tænke og lære på (pp. 4-5):

Computers can be the technical foundation of a new and dramatically enhanced literacy, which will act in many ways like current literacy and which will have penetration and depth of influence comparable to what we have already experienced in coming to achieve a mass, text-based literacy. [...]

If a true computational literacy comes to exist, it will be infrastructural in the same way current literacy is in current schools. Students will be learning and using it constantly through their schooling careers and beyond in diverse scientific, humanistic, and expressive pursuits. Outside of schools, a computational literacy will allow civilization to think and do things that will be new to us in the same way that the modern literate society would be almost incomprehensible to preliterate cultures.

Beherskelse af nye sprogformer gør det muligt for os at tænke og udtrykke nye, supplerende tanker – at vokse intellektuelt både som individ og samfund – og med informatikkens mulighed for automatisering af abstraktioner kan disse tanker manifestere sig i vilkårligt avancerede digitale artefakter.

1.4 Den fjerde fundamentale kompetence

I det 21. århundrede er informatik ikke blot en sag for specialister. Informatik er et nyt dannelsesområde og en ny basiskompetence, som alle bør udstyres med fra barnsben – helt på niveau med at kunne læse, skrive og regne. Intet mindre. Som Barack Obama har udtrykt det: "*In the new economy, computer science isn't an optional skill — it's a basic skill for everyone.*"

I den industrielle tidsalder blev læse-, skrive- og matematikkompetencer anset for fundamentale og nødvendige for alle for at kunne lære, agere og udmærke sig inden for alle fag i skolen og professioner i arbejdslivet. I det 21. århundrede skal disse tre fundamentale kompetencer suppleres med en fjerde, informatik.

Hvor matematik (især) er de tekniske og naturvidenskabelige fags sprog, er informatik hastigt i færd med at blive et sprog for alle fag og professioner (World Economic Forum, 2018), og som nævnt i forrige afsnit bliver informatik og computationelle metoder mere og mere udbredt som videnskabelig metode, der inden for alle videnskabelige domæner supplerer teori og eksperiment (mere om dette i afsnit 3). Informatikkens rolle og position i uddannelsessystemet bør reflektere denne samfundsmæssige betydning og relevans.

Det handler ikke kun om at uddanne flere specialister, som kan udvikle og designe de mange digitale artefakter, der i stigende grad sætter rammerne for det liv, vi lever. Det handler først og fremmest om at sikre, at vi som mennesker bliver klædt på til *kompetent informatisk adfærd*, således at vi kan omsætte vores teknologiske fremskridt til bedre og mere meningsfulde liv i et mere sikkert, retfærdigt og demokratisk samfund (Besenbacher & Caspersen, 2017; Ko 2018).

Visionen er, at alle, uafhængigt af deres særlige interesse, ekspertise og (fremtidige) profession, skal uddannes til at kunne omfavne informatikken og anvende informatisk viden og færdigheder som en integreret kompetence i alle fag og professioner.

Dette fordrer en *tostrøget strategi* for informatik på alle uddannelsesstrin: som *specialisering*, dvs. et fundamentalt og selvstændigt fag i skolen og som specialiserede videregående uddannelser; og ved *integrering* af informatikfaglighed i andre skolefag og uddannelser.

En overordnet definition af kompetent informatisk adfærd – herunder digital dannelse, digitale kompetencer og 'Computational Thinking' – er præsenteret i figur 1.1.

- (0) Evnen til kvalificeret, kreativt, kritisk og etisk forsvarligt at kunne bruge relevante computerbaserede værktøjer.
- (1) Evnen til at analysere, designe, realisere og evaluere computationelle data- og informationsprocesser.
- (2) Brug af (1) – som en udvidelse af vores mentale evner – til computationelt at erfare, analysere, begribe, repræsentere og manipulere verden og skabe ting, der har betydning for os.

Figur 1.1: Kompetent informatisk adfærd og 'Computational Thinking'

(0) er en reflekteret brugskompetence, der – med alderen i forhold til mere og mere avancerede og specialiserede værktøjer – skal udvikles gennem hele uddannelsessystemet, men også i voksenlivet og her specielt i professionsorienterede sammenhænge.

(1) og (2) er det banebrydende, som fremadrettet bør findes på *alle* uddannelsestrin både som selvstændigt fag/uddannelse og integreret i andre fag/uddannelser.

Meget forenklet kan man karakterisere (1) som 'Learn to Compute' og (2) som 'Compute to Learn and Create' (se afsnit 2.1).

Populært kan man sige, at informatik og programmering er det 21. århundredes blyant, med hvilken vi kan skabe helt nye udtryk, som er digitale og dynamiske i stedet for analoge og statiske – virtuelle verdener, om man vil. Informatikken er også det 21. århundredes mikroskop, igennem hvilket vi med digitale og dynamiske modeller kan tilgå, analysere og reflektere over verden – kort sagt: erkende verden. Ikke bare den digitale verden, men verden som sådan.

1.5 'Program or be programmed'

Allerede omkring 1960 leverede to ekstremt visionære personer uafhængigt, men næsten synkront, henholdsvis pisk og gulerod for nødvendigheden af at inkludere informatik i almen uddannelse.

C.P. Snow leverede pisken (guleroden følger i afsnit 1.6). I bogen *The two cultures and the scientific revolution* (Snow, 1959) diskuterer Snow splittelsen mellem science og humaniora og problematiserer, at denne splittelse ikke bidrager til løsning af verdens problemer.

Det er imidlertid ikke den diskussion, vi vil tage op her, men derimod et par citater fra bogen – et par citater, som er bemærkelsesværdigt visionære, og som i lyset af den seneste tids debat om Google og Facebook er mere aktuelle end nogensinde:

Those who don't understand algorithms, can't understand how decisions are made.

A handful of people, having no relation to the will of society, having no communication with the rest of society, will be taking decisions in secret which are going to affect our lives in the deepest sense.

Diskussionen kan sammenfattes i dette udsagn: Den største fare er ikke kloge robotter, men dumme mennesker. Digital/computational magtkoncentration og en uvidende befolkning er ikke et attraktivt fremtidsscenario.

I en trods alt ikke så fjern fortid var det kun et fåtal forundt at kunne læse og skrive, og selv da flere lærte at læse, var det kun et fåtal, der havde adgang til den tids skrifter.

Det betød, at samfundsmagten var centralt forankret i nogle ganske få institutioner, som generelt gjorde alt, hvad der stod i deres magt – hvilket var en hel del – for at bevare og udbygge magten ved at undertrykke og udnytte menneskeheden.

Heldigvis er vi kommet langt siden den mørke middelalder. Inden for de sidste 100-200 år har størstedelen af menneskeheden lært at læse og skrive, og ingen er vel i tvivl om, hvilken betydning det har for en human, demokratisk og ligeværdig samfundsudvikling.

Paradoksalt nok risikerer vi imidlertid, at netop informationsteknologien kan være medvirkende til at skabe en ny mørk tidsalder, nemlig hvis ikke alle børn og unge lærer præmisserne for informationsteknologien samt at erkende og udtrykke sig vha. denne.

Formuleringen er måske lige vel dramatisk og kategorisk, men pointen er, at det essentielle 'digital divide' ikke er mellem de, der kan bruge informationsteknologien, og de, der ikke kan. Det essentielle 'digitale divide' er mellem de, der skaber teknologien, og de, der er underlagt den.

I bogen *Program or be programmed* er det udtrykt meget direkte (Rushkoff, 2010):

In a digital age, whoever holds the keys to programming ends up building the reality in which the rest of us live.

Derfor skal informatik være en del af almindelsen, så alle børn og unge uddannes til at være aktive samfundsborgere, der med deres viden, færdigheder og holdninger medvirker til at forme den politiske, økonomiske og digitale virkelighed.

1.6 Automatisk udførelse af processer forandrer alt

Guleroden for at inkludere informatik i almen uddannelse blev leveret af Alan Perlis, den første modtager af Turingprisen (datalogiens Nobelpris), da han i 1961 argumenterede således:

Computer science should be part of a liberal education. Computer science is the study of process. Automated execution of process changes everything!

Bemærk i øvrigt, at 1961 var et år før første datalogiske institut i USA så dagens lys, og ti år før vi fik sådanne enheder i Danmark.

Kristen Nygaard (også modtager af Turingprisen) havde tilsvarende indset, at informationsprocesser stod helt centralt i faget (Nygaard, 1986):

Informatics is the science that has as its domain information processes and related phenomena in artefacts, society and nature.

Også Nygaard argumenterede for, at informatik skulle være et fundamentalt fag i skolen.

Peter Naur, der modtog Turingprisen i 2005, formulerede tidligt en mere vidtgående vision, idet han allerede i 1967 argumenterede for datalogiens fundamentale natur og en plads i almen uddannelse på niveau med sproglære og matematik (Naur, 1967).

Processer står vitterligt centralt i faget, både som metode og genstandsfelt. Mere og mere avancerede informationsprocesser inden for en bredere og bredere kreds af fag og professioner er gjort til genstand for computationel modellering (digitalisering) i bestræbelser på at understøtte eller helt automatisere komplekse arbejdsgange og processer, herunder i stadigt stigende grad kognitive processer.

Et konkret eksempel fra skoleverdenen kunne være at automatisere den relativt komplekse proces, det er at løse andengradsligninger (Schwartzbach, 2011) – en proces/algoritme vi alle har brugt mange, mange timer af vores skoletid på at lære og øve os i at udføre (figur 1.2).

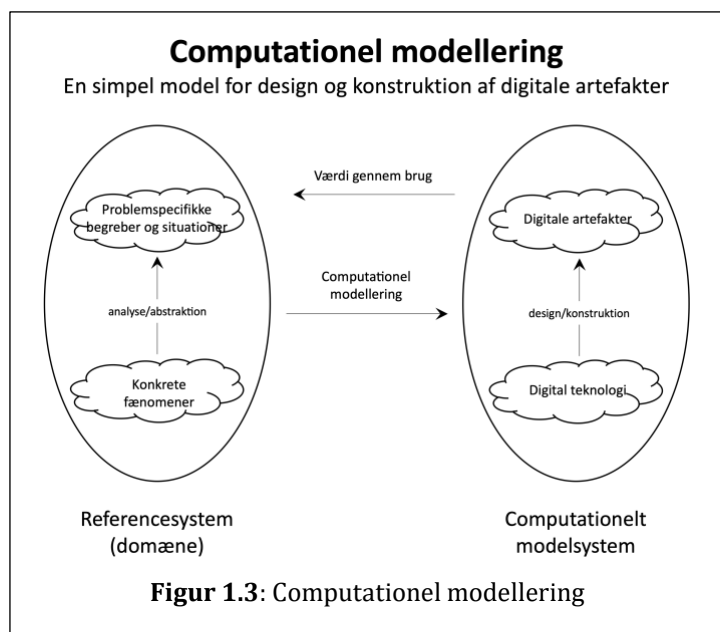
Selv dette enkle eksempel illustrerer det potentiale, der er i informatikken, og de implikationer, det må få i forhold til prioritering og behandling af indhold i andre fag og uddannelser i uddannelsessystemet.



Figur 1.2: Computational modellering af kognitive processer ('automation of abstractions'); f.eks. et program, der kan løse andengradsligninger.

Mere generelt, men meget forenklet, kan man sige, at computationel modellering består i at begribe en reel eller imaginær del af verden (referencesystemet/domænet) og repræsentere denne i et computationelt modelsystem i en form, som muliggør konstruktion af digitale artefakter, der skaber værdi i verden (figur 1.3).

Et konkret eksempel er vejret, hvor massive datamængder opsamles og benyttes i meteorologiske modeller, der kan bruges til vejrudsigter. Et andet eksempel er sundhedsplatforme, hvor data om patienter, diagnoser, behandlinger m.m. repræsenteres og gøres tilgængelige i et system, hvor data kan tilgås fra flere steder og medvirke til at give patienter bedre oplevelser af mødet med sundhedssystemet samt at effektivisere arbejdsgange for sundhedspersonalet. Et tredje og mere jordnært eksempel er et system til understøttelse af skole-hjem-kommunikation.



Det er essentielt, at vi giver børn og unge evne til at medskabe den fremtid, der opstår gennem design og konstruktion af digitale artefakter.

Det indbefatter en kritisk og nysgerrig stillingtagen til teknologien og en evne til at forstå den effekt, som digital teknologi har på børnene selv, deres nærmeste og samfundet (Iversen og Klokmose, 2017).

Og det indbefatter at skabe studerende med videbegær og nysgerrighed, der besidder personlige færdigheder som problemløsning, samarbejdsevne og kreativitet, der kan blive anvendt på tværs af fagområder og ekspertiser. På niveau med sprog og matematik skal informatik og CT være en drivkraft i denne proces (Besenbacher & Caspersen, 2017).

Myndiggørelse af børn og unge ift. den computationelle revolution fordrer, at uddannelsessystemet gradvist, men til fulde, omfavner denne faglighed. Det er en meget stor opgave; til gengæld er potentialet enormt.

2. CT – baggrund, karakteristik og relaterede begreber

Afsnittet introducerer begrebet Computational Thinking. Afsnittet vil bl.a. berøre CT's genstandsfelt og formål samt beslægtede begreber som mathematical thinking, STEM og makerspaces.

Informatik er et selvstændigt videnskabsområde, der har eksisteret i mere end 80 år – længere end den teknologi, vi forbinder dermed; og som nævnt i forrige afsnit blev visionen for informatik allerede formuleret i første halvdel af 1800-tallet. I Danmark har vi haft videregående uddannelse inden for området i 50 år, internationalt lidt længere.

På det seneste er der kommet et globalt fokus på informatik som en fundamental kompetence, der er relevant for alle. Kimen hertil var Jeannette Wings tre-siders artikel om Computational Thinking (Wing, 2006). Artiklen, og specielt begrebet, åbnede manges øjne for vigtigheden af at inkludere informatik/CT som en fundamental komponent i almen uddannelse.

I 2016 fandt termen vej til en dansk politisk publikation, idet Digitalt Vækstråd i *Rapport om kvalificeret arbejdskraft* som én af fem "Her & nu" anbefalinger skrev (Vækstråd, 2016):

„Computational Thinking“ gøres til en integreret og obligatorisk del af undervisningen på relevante uddannelser – både i folkeskolen, på ungdomsuddannelser, de videregående uddannelser samt på voksen- og efteruddannelsesforløb. Det er afgørende, at det rette digitale og analytiske videnniveau også sikres blandt undervisere i uddannelsessystemet.

Anbefalingen er siden fulgt op af Digitalt Vækstpanel (Vækstpanel, 2017), af generelle politiske initiativer (Regeringen, 2018; Teknologipagt 2018) og af særlige politiske initiativer ift. folkeskolen (UVM, 2018) og universitetsuddannelser (UFM, 2018a, 2018b).

Men hvad er baggrunden for begrebet CT, hvilken rolle spiller det i den internationale diskurs, og hvordan kan det forstås ift. relaterede begreber? Det giver dette afsnit en forklaring på. Emnet diskuteres mere udførligt i appendiks A.1.

2.1 Karakteristik og udbredelse af CT

Almengørelse af begrebet CT, som det diskuteres i tiden, blev diskret introduceret af Seymour Papert i 1980², men for alvor lanceret og udbredt af Jeannette Wing i 2006 og fremefter.

Papert brugte kun begrebet to gange (Papert, 1980; Papert, 1996). For Papert udtrykte CT en vision om programmering som en radikalt ny erkendelses- og udtryksform:

Computational Thinking is the use of programming —as an extension of our mind —to experience and understand the world, to manipulate the world, and to create things that matter to us.

² I manuskriptet til en kommende bog (Tedre & Denning, 2019) argumenterer forfatterne for, at CT's stamtavle går ca. 4.000 år tilbage! Og som vi omtalte i afsnit 1 er CT's rolle i moderne forskning også af ældre dato end både 2006 og 1980.

Da Jeannette Wing lancerede CT (Wing, 2006) var det ud fra denne vision:

The use of computational concepts, methods and tools will transform the very conduct of every discipline, profession and sector. Someone with the ability to use computation effectively will have an edge over someone without.

De to visioner er ikke i modstrid med hinanden – tværtimod komplementerer de på forbilledlig vis hinanden, som det fremgår af figur 1.1 side 6 om kompetent informatisk adfærd og CT.

Paperts vision kan kort karakteriseres som 'Compute to Learn and Create', hvor Wings vision omvendt kan karakteriseres som 'Learn to Compute', og det er netop den selvforstærkende effekt i *kombinationen* af de to perspektiver, der giver det enorme potentiale i uddannelse.

CT er således en begrebsramme for den informatikfaglighed, der er alment relevant – ikke blot i informatik, men også i andre fag og professioner (Wing, 2006, 2008, 2014; Caspersen et al., 2018).

Wing definerer CT således:

Computational Thinking is the thought processes involved in formulating a problem and expressing its solution(s) in a way that a computer can carry it out [without human interpretation].

Wings definition er udtrykt i meget overordnede termer, og selv i publikationer, hvor begrebet uddybes, er det temmelig abstrakte termer, der benyttes til at karakterisere CT (Wing, 2011):

*Thus, computational thinking overlaps with **logical thinking** and systems thinking. It includes **algorithmic thinking** and parallel thinking, which in turn engage other kinds of thought processes, such as **compositional reasoning, pattern matching, procedural thinking** and **recursive thinking**. Computational thinking is used in the design and analysis of problems and their solutions, broadly interpreted.*

*The most important and high-level thought process in computational thinking is the abstraction process. **Abstraction** is used in defining **patterns, generalizing** from specific instances and **parameterization**. It is used to let one object stand for many. It is used to capture essential properties common to a set of objects while hiding irrelevant distinctions among them.*

Wings definition og karakteristik af CT er formuleret som del af en vision for informatikkens betydning for forskning og udvikling inden for alle discipliner og professioner og dermed som en vision for en ny almen komponent i videregående uddannelse. Wings definition og karakteristik af CT har aldrig været ment som en anbefaling af nøglebegreber i et curriculum for K-12.

Imidlertid har Wings definition af CT, der som pointeret er udtrykt i meget generelle og abstrakte termer, gået sin sejrsgang verden over som en væsentlig del af argumentationen for, at man skal lære informatik i skolen. På basis heraf har en række forskellige aktører, der arbejder med at promovere og implementere informatik i K-12 (0.-12. klassetrin) taget CT til sig og målrettet politikere, beslutningstagere, lærere og ledere udarbejdet vejledninger, der forsøger at forklare disse termer og dermed CT. Sådanne aktører er f.eks. Computer Science Teacher

Association i USA (CSTA) og Computing At School i UK (CAS), men også Europakommissionen (European Commission, 2016).

Til eksempel udfoldes CT af (CAS, 2015) således:

Computational thinking is a cognitive or thought process involving logical reasoning by which problems are solved and artefacts, procedures and systems are better understood.

CAS-hæftet (CAS, 2015) har to centrale afsnit, *Concepts of computational thinking* (pp. 6-8) og *Techniques associated with computational thinking* (pp. 9-13), som hver især har følgende underafsnit:

Concepts of computational thinking: *Algorithmic Thinking; Decomposition; Generalisation (patterns); Abstraction; Evaluation*

Techniques associated with computational thinking: *Reflecting; Coding; Designing; Analysing; Applying.*

Bortset fra de to understregede ord er der ingen af begreberne, der antyder, at det er CT og dermed informatik, det handler om; det kunne ligeså vel være matematik, biologi, psykologi, musik – ja, stort set et vilkårligt andet fagområde. Det kan ses som en styrke, men også som en svaghed!

CT er bl.a. karakteriseret ved disse begreber, men det omvendte er *ikke* tilfældet: at man kan karakterisere aktiviteter vha. disse begreber, gør det ikke i sig selv til CT. Essensen af CT er at formulere problemer så tilstrækkelig klart, at de kan modelleres computationelt (jvf. figur 1.3 side 8).

CT-vejledningerne fra CSTA og CAS er blevet normsættende for en omfattende mængde undervisningsmateriale, som i disse år udvikles over hele kloden (i mange tilfælde ved at oversætte/bearbejde materiale fra nogle ganske få kilder), og som på forskellig vis tilstræber at give elever forståelse, viden og kompetencer om disse højniveaubegreber og dermed CT.

CT-vejledningerne spiller også en central rolle i diverse politiske dokumenter herhjemme og i EU (Vækstråd, 2016; European Commission, 2016; Vækstpanel, 2017).

På den ene side er det en udvikling uden sidestykke i verdenshistorien! Næppe tidligere er det sket, at en faglig disciplin, som der alment i befolkningen er en yderst begrænset viden om, så hurtigt og massivt kommer ind i uddannelsessystemet.

På den anden side er det spørgsmålet, om vi griber det rigtigt an – specielt ved at lade nogle få fortolkninger af de meget overordnede og abstrakte begreber være rammesættende for udformning af curriculum og undervisningsmaterialer for informatik-faglighed og CT i K-12.

På trods af den enorme internationale aktivitet er området stadig meget umodent og præget af afsøgende pioneraktiviteter mht. udformning af curriculum, materialer, evaluering osv. Der er ingen klare og entydige resultater, der afstikker retning for præcis, hvorledes informatik/CT skal realiseres i uddannelsessystemet. Men det ville også være for godt til at være sandt.

Man er kun lige begyndt at begribe potentialet i CT, og hvordan denne vision kan almengøres og realiseres generelt i uddannelsessystemet. Det faktum illustreres med al ønskelig tydelighed af

de mange beslægtede termer som internationalt fremtrædende forskere aktuelt bringer til torvs, og som alle på den ene eller anden måde viderefører Paperts, DiSessas og Wings visioner for Computational Thinking og Computational Literacy.

Overordnet er mange af disse termer karakteriseret ved en kombination af Wings vision med Paperts supplerende fokus på brug af programmering til forståelse og manipulering af omverdenen samt skabelse af artefakter af betydning, eksempelvis Computational Participation (Kafai & Burke, 2013), Computational Literacy (Wilensky, 2014), Computational Fluency (Resnick, 2017) og Computational Action (Abelson, 2018).

Udviklingen i disse termer og deres perspektiver kan siges at være sammenfattet i begrebet Computational Empowerment (Iversen et al., 2018):

We define computational empowerment as the process in which children, as individuals and groups, develop the skills, insights and reflexivity needed to understand digital technology and its effect on their lives and society at large, and their capacity to engage critically, curiously and constructively with the construction and deconstruction of technology.

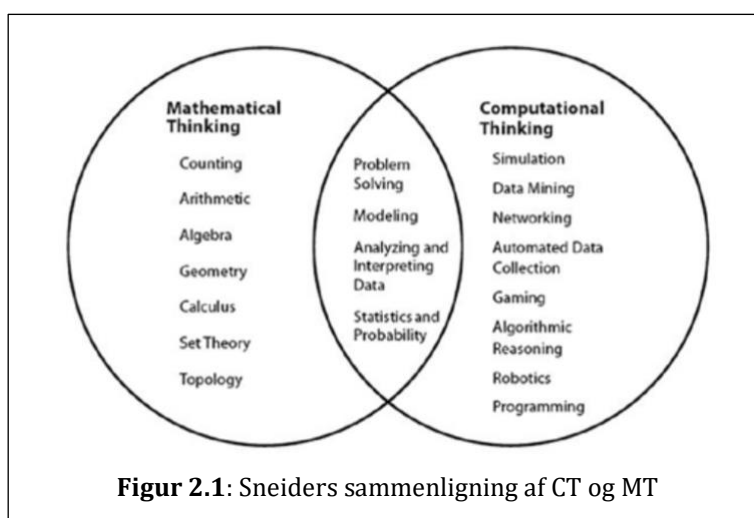
I de følgende tre afsnit gives et ganske kort overblik over CT-diskursens relation til beslægtede emner som 'mathematical thinking', STEM og makerspaces.

2.2 CT og 'mathematical thinking'

Mange af de begreber, der benyttes til overordnet karakteristik af CT, f.eks. abstraktion og problemløsning, er nært beslægtet med tilsvarende begreber fra 'Mathematical Thinking' (MT).

MT handler om brug af matematiske færdigheder og kompetencer til løsning af matematiske problemer i andre faglige sammenhænge, primært vha. ligninger og funktioner. (Harel & Snowden, 2005, p.31) karakteriserer MT som universelle matematiske kompetencer, der er relevante på tværs af discipliner og problemtyper, og som en fundamental måde at tænke om problemer på ("*[MT] governs one's way of understanding*").

Relationen mellem CT og MT kan ifølge (Sneider et al., 2014) beskrives vha. et Venn-diagram, der illustrerer ligheder og forskelle. Det skal imidlertid understreges, at det er en klassisk Wing-fortolkning af CT, der ligger til grund for denne sammenligning (figur 2.1). Figuren skal derfor ikke forstås som en universel sandhed, men snarere som et indlæg i debatten om, hvad CT er og ikke er.



Figur 2.1: Sneider's sammenligning af CT og MT

Den exceptionelle, men også kontroversielle, fysiker Stephen Wolfram skriver om relationen mellem matematik og Computational Thinking således (Wolfram, 2016):

Computational thinking is going to be a defining feature of the future—and it's an incredibly important thing to be teaching to kids today. There's always lots of discussion (and concern) about how to teach traditional mathematical thinking to kids. But looking to the future, this pales in comparison to the importance of teaching computational thinking. Yes, there's a certain amount of traditional mathematical thinking that's needed in everyday life, and in many careers. But computational thinking is going to be needed everywhere. And doing it well is going to be a key to success in almost all future careers.

As far as I'm concerned, its intellectual core is about formulating things with enough clarity, and in a systematic enough way, that one can tell a computer how to do them. Mathematical thinking is about formulating things so that one can handle them mathematically, when that's possible. Computational thinking is a much bigger and broader story, because there are just a lot more things that can be handled computationally.

I stedet for som Sneider at fokusere på relationen mellem MT og CT fokuserer Wolfram på virkefeltet for hhv. matematisk og computationel modellering. Wolframs væsentlige pointe er, at virkefeltet for computationel modellering er meget større end virkefeltet for matematisk modellering – eller sagt med andre ord: at mængden af fænomener, der kan håndteres computationelt, er langt større end mængden af fænomener, der kan håndteres matematisk.

På et filosofisk plan kan dette diskuteres, men i praksis har Wolfram en pointe – specielt fordi der kan sættes strøm til computationelle modeller – de kan udføres.

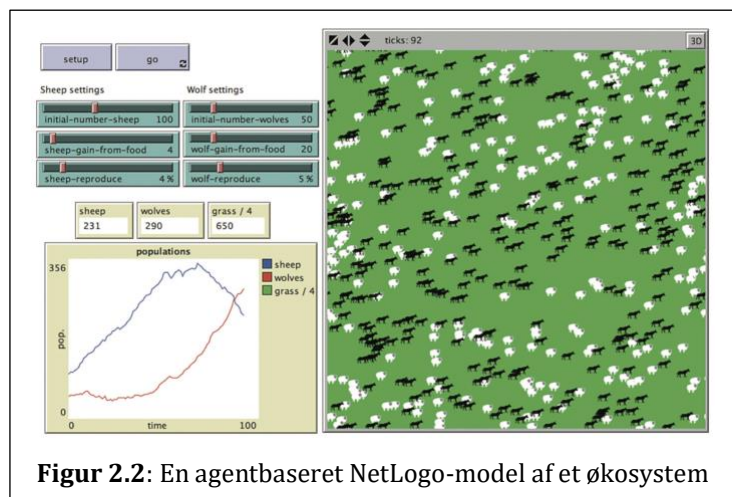
2.3 CT og STEM

Flere forskningsgrupper arbejder specifikt med at realisere CT i kontekst af sciencefag og matematik. De mest markante eksempler er Uri Wilenskys *CT-STEM Initiative* på Northwestern University i Chicago og kollegaen Alexander Repennings *Scalable Game Design Initiative* på University of Colorado i Boulder.

Wilenskys gruppe arbejder med en modelbaseret tilgang til CT baseret på modelleringsværktøjet NetLogo, der er udviklet specifikt til agentbaseret modellering af komplekse dynamiske systemer (f.eks. trafikken i en bydel, miljøet omkring en myretue eller priselasticitets indvirkning på afsætning og forbrugeradfærd/-vilkår).

I figur 2.2 er Wilenskys tilgang illustreret med et økosystem af rovdyr, byttedyr og føde (ulve, får og græs).

Agentbaseret modellering tager udgangspunkt i beskrivelse af den ofte meget simple adfærd, som de enkelte agenter i systemet har isoleret set og i interaktion med hinanden (f.eks.: hvis jeg som trafikant oplever mere kødannelse i min vognbane end i vognbanen ved siden af, skifter jeg til denne vognbane).



Figur 2.2: En agentbaseret NetLogo-model af et økosystem

Ved at "befolke" modellen med et tilpas stort antal agenter og simulere deres adfærd over tid ved at sætte strøm til modellen, opstår emergens – en kompleks systemadfærd, som kan undersøges og gøres til genstand for eksperimenter ved at justere modellens parametre og den regel-baserede (programmerede) adfærd af modellens agenter.

Med hensyn til uddannelsesniveau er fokus for Wilenskys arbejde på gymnasiet og videregående uddannelser (Wilensky, 2014; Weintrop et al., 2016; CT-STEM, 2018).

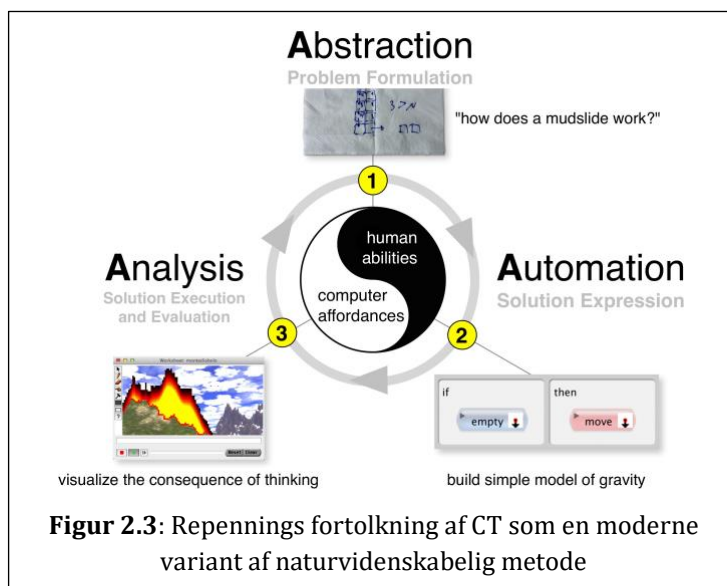
Alexander Repennings gruppe arbejder med en lignende tilgang, men med grundskoleelever som målgruppe og med afsæt i et andet modelleringsværktøj, AgentCubes, som er udviklet specielt til målgruppen (Repenning, 2012; Repenning et al. 2016).

Repenning karakteriserer sin tilgang således:

While the term CT is relatively new, the process implied by Wing can be recognized as a computationally enhanced version of the well-established scientific method. Based on Wing's definition the CT process can be segmented into three stages: 1. Problem Formulation, 2. Solution Expression and 3. Execution & Evaluation.

Dette er helt centralt og illustrerer CT som videnskabelig metode (og som nævnt i afsnit 1.2 som den tredje videnskabelige metode, der komplementerer teori og eksperiment).

Repennings fortolkning af CT som videnskabelig metode er illustreret i figur 2.3, hvor der eksemplificeres med simulering af et jordskred. Metoden involverer først en problemformulering (hvordan virker et jordskred?), efterfulgt af løsningsmodel (simpel simuleringsmodel, hvor reglerne for agenterne programmeres) og afrundet med eksperimenter og evaluering (visualisering af konsekvenserne af jordskred). Dette giver idéer til justering/forbedring af modellen, og dermed kan processen gentages.



I Danmark er der gennemført et pilotprojekt, og der er netop initieret nye projekter, hvor en agentbaseret tilgang med NetLogo udforskes i gymnasiefag – bl.a. et projekt finansieret af VILLUM FONDEN om CT i matematik og sciencefag (mere herom i afsnit 4.2).

Den agentbaserede tilgang blev kort omtalt i afsnit 1.3 (bølgemodellen) og diskuteres endvidere i afsnit 5.3 som et af de store potentialer ved at introducere informatik/CT i uddannelsessystemet.

2.4 CT og makerspaces

Med udviklingen af nye digitale fabrikationsteknologier (såsom 3D printere og vinylskærere) og små programmerbare computere (såsom Micro:Bits, Arduinos og Makey-Makeys) er det i dag

muligt at invitere børn og unge med helt ind i designprocessen af nye teknologier. Det har sat et stort aftryk også inden for uddannelse. Således har fabrikationsteknologierne igennem de seneste 10 år skabt en opblomstring af uddannelsesorienterede 'makerspaces' eller 'digitale fabrikationslaboratorier' verden over med rod i forskningen og netværksdannelsen omkring den globale FabLearn organisation. FabLearn er en global bevægelse for forskere, undervisere og eksperter, der undersøger, hvordan digitale teknologier kan anvendes i uddannelser. Formålet med anvendelse af makerspaces i uddannelser er udtrykt af FabLearn stifter, Paulo Blikstein (Blikstein, 2018):

FabLearn Labs (formerly known as FabLab@School) are a worldwide growing network of educational digital fabrication labs that put cutting-edge technology for design and construction – such as 3D printers and laser cutters and robotics – into the hands of middle and high school students.

Such labs are a place for invention, creation, discovery and sharing, a space of inquiry where everyone learns and knowledge gets integrated into personal interests and daily life.

Anvendelse af FabLearn og digitale teknologier inden for uddannelser tager globalt set karakter efter de enkelte uddannelsessystemers karakteristika. I såvel USA som Asien knyttes FabLearn tæt sammen med det stigende fokus på STEM-uddannelser og Computer Science (Blikstein, 2013; Martin, 2015). I Tyskland kobles FabLearn med muligheden for at træne elevernes digitale medborgerskab og deres 'modelforståelse' for digital teknologi. Her ses tydeligt forbindelsen til den tyske 'bildung' tradition, hvor barnets almene dannelse også varetages i makerspaces (Schelhowe, 2013). I Belgien og Holland eksisterer også stærke udviklingsmiljøer for makerspaces i uddannelser. Her dyrkes makerspaces særligt i forhold til design og formgivning.

3. Intermezzo om CT i erhverv og videnskab

Computational Thinking praksisser udøves overalt i vores samfund. Derfor er en styrkelse af CT vigtig, uanset om målet er at styrke dansk industri, dansk forskning eller forberede danske skole- og gymnasieelever på fremtidens samfund. I dette kapitel gives et kort overblik over, hvordan CT-praksisser har gjort deres indtog i stort set alle brancher og professioner.

Vi indledte dette notat med bemærkninger om, at verden i stigende grad bliver digitaliseret, og at informatikken bidrager med radikal og transformerende udvikling af professioner, videnskabelige discipliner og vores sociale liv i almindelighed.

Mads Tofte, IT-Universitetet i Københavns rektor gennem næsten 20 år (1999-2018), har formuleret de radikale og transformerende muligheder på følgende vis:

Med it kan mennesker skabe, dele og håndtere tanker, produkter, processer og services, som skaber nye, effektive og grænseoverskridende muligheder, der er utænkelige uden den digitale teknologi.

Vi kender alle til, hvorledes teknologien vinder indpas i såvel arbejdsliv som privatliv, og i en tid hvor begrebet "disruption" fylder en del på den politiske og erhvervsmæssige dagsorden, kan de fleste nok også give eksempler på produkter eller services, der giver nye og grænseoverskridende muligheder, som ikke ville kunne lade sig gøre uden den digitale teknologi.

Den digitale udvikling er på ingen måde ny, men først de seneste år er der på højeste politiske niveau kommet en global erkendelse af, hvor grundlæggende og omfattende forandringerne er for vores samfund, og at udviklingen skal adresseres med ambitiøse tiltag blandt andet ift. uddannelse (se afsnit 1, specielt 1.4).

En kraftigt medvirkende årsag til den gryende politiske erkendelse var World Economic Forums Davos-møde i 2016. Temaet for mødet var "Mastering the Fourth Industrial Revolution". Klaus Schwab, grundlægger og bestyrelsesformand for World Economic Forum introducerede mødet således (World Economic Forum, 2016):

We must develop a comprehensive and globally shared view of how technology is affecting our lives and reshaping our economic, social, cultural, and human environments. There has never been a time of greater promise, or greater peril.

Pierre Nanterme, CEO for Accenture, understregede udviklingen og gav næring til "disruption"-begrebet på denne måde:

Digital is the main reason just over half of the companies on the Fortune 500 have disappeared since the year 2000.

I Danmark har det affødt en række politiske rapporter og initiativer kulminerende med regeringsinitiativet Strategi for Danmarks digitale vækst i januar 2018 (Vækstråd, 2016; Vækstpanel, 2017; Disruptionråd, 2017; Teknologipagt, 2018; Regeringen, 2018), og i 2019 følger regeringsinitiativet Digitalt Topmøde 2019, det første af sin slags.

Således omfavner professioner, erhverv og det politiske system i stadigt stigende grad digitalisering og dermed CT som et middel til at transformere og udvikle nye produkter,

processer og services, og hele sektorer er i færd med at konceptualisere digitaliseringspotentialer og formulere standarder for systemgrænseflader og deling af data m.m., f.eks. sundhedssektoren (Riisgaard Hansen, 2016).

Tilsvarende omfavner flere og flere videnskabsområder CT som et middel til at transformere og udvikle nye videnskabelige metoder. Også dette emne berørte vi indledningsvist (afsnit 1.2), men lad os uddybe en smule.

For praktisk talt ethvert videnskabeligt område X eksisterer der i dag et tilsvarende Computational X ; det gælder så vidt forskellige videnskabelige emner som økonomi, musik, historie, lingvistik, litteratur, teologi, samfundsvidenskab, arkæologi, jura, psykologi, journalistik, etnografi, medicin og uddannelsesvidenskab samt naturligvis naturvidenskaberne fysik, kemi, biologi, hvor der er både inden for fysik og kemi er uddelt nobelpriser specifikt for computationelle bedrifter.

Også i Danmark finder denne udvikling sted; her skal blot nævnes nogle få eksempler, der repræsenterer denne udvikling:

- Center for Information og Boblestudier ved Københavns Universitet benytter f.eks. agentbaseret modellering (se afsnit 2.3) som en væsentlig komponent i deres forskningsaktiviteter (Boblestudier, 2018).
- Sociologer på Center for Social Data Science, ligeledes ved Københavns Universitet, benytter tilsvarende computationelle metoder som en definerende komponent i deres forskning og som supplement til klassiske kvalitative og kvantitative metoder (SODAS, 2018).
- Antropologer i TANTLAB (The Techno-Anthropological Laboratory) ved Aalborg Universitet benytter tilsvarende computationelle metoder i deres forskningslaboratorium (TANTLAB, 2018).
- Ved Syddansk Universitet har man etableret det tværfakultære SDU eScience Center, som dels udvikler en computationel infrastruktur og dels støtter forskere, der vil bruge computationelle metoder i deres forskning (eScience, 2018). Ved Aarhus Universitet er man i færd med at etablere et tilsvarende center.
- Ved Aalborg Universitet har man netop åbnet CALDISS (Computational Analytics Laboratory for Digital Social Science) – et digitalt data- og metodelaboratorium for alle med tilknytning til det Samfundsvidenskabelige Fakultet ved AAU (CALDISS, 2018).
- Ved Aarhus Universitet arbejder projektet ScienceAtHome med at rykke grænserne for forskning ved gennem computationelle teknikker og computerspil at involvere borgere i at løse store forsknings spørgsmål (ScienceAtHome, 2018).

I de kommende år vil vi se computationelle metoder brede sig endnu mere inden for alle forskningsområder som en tredje videnskabelig metode, der komplementerer teori og eksperiment. Og som nævnt i afsnit 1.2 er det nogle gange den eneste holdbare metode, når teori er u håndterlig, og eksperimenter er ugennemførlige.

4. CT i uddannelse – status

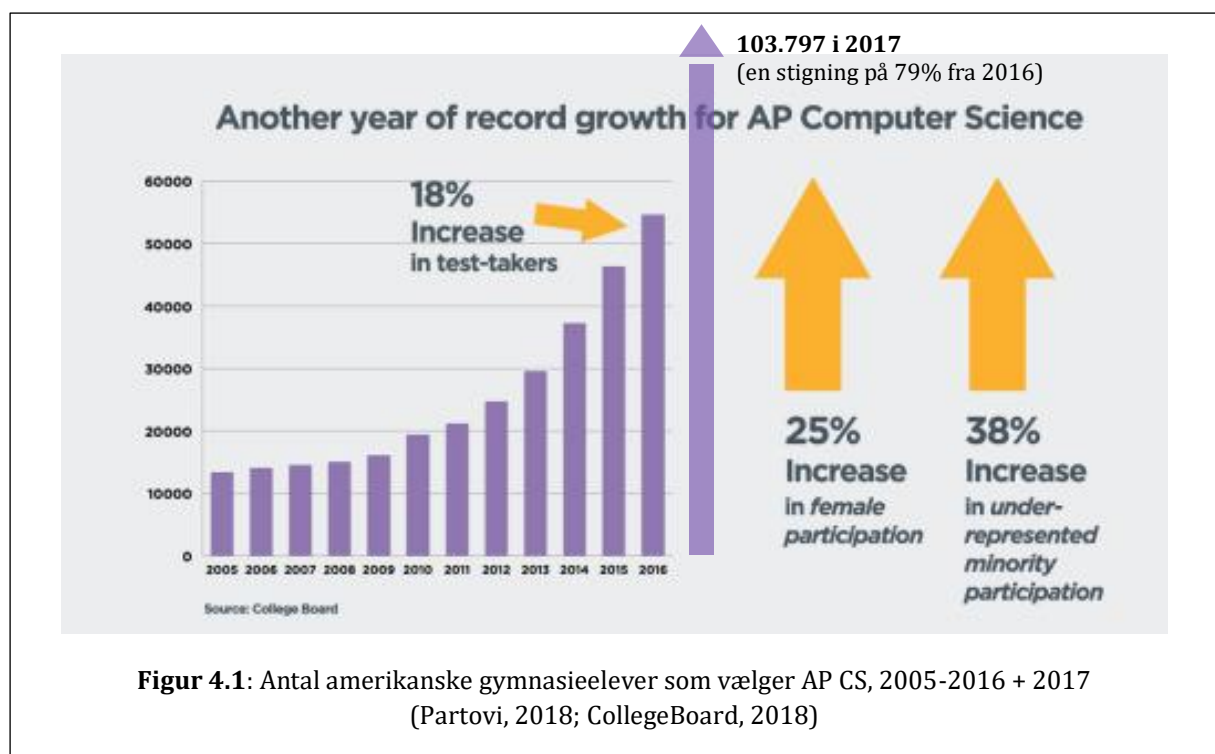
Dette afsnit beskriver, hvordan der eksperimenteres med CT i uddannelsessystemet dels ved at give en oversigt over internationale initiativer og erfaringer og dels ved at præsentere aktuelle nationale initiativer.

4.1 CT i international uddannelse

I dette afsnit vil vi fokusere på udvikling i USA og UK, da denne har været meget markant, er veldokumenteret og har inspireret mange lande til at gå samme vej, herunder Danmark.

2008 var et markant år, idet mange lande – uafhængigt af hinanden – netop det år igangsatte initiativer til udvikling af almene it-fag i skolen.

I USA initierede man et større redesign af det Computer Science-kursus som gymnasier kan udbyde, og som giver kredit, når man starter på college-uddannelse (CollegeBoard, 2018). Kurset har eksisteret i mange år, og det har længe været et introducerende programmeringskursus i tidens aktuelle begynderprog på college (Pascal, C++, Java). Populariteten var imidlertid ikke stor og endda dalende. Redesignet, kendt som CS Principles, førte til et moderne, alment kursus, som er blevet voldsomt populært – også for kvinder (se figur 4.1).



Græsrodsbevægelsen Computing At School (CAS, 2018) i UK startede i 2008. I begyndelsen var det blot fire-fem personer, men bevægelsen fik en utrolig vind i sejlene og fik i 2012 prominent støtte af Royal Society, og så skete der for alvor ting og sager. Denne udvikling vender vi også tilbage til.

I New Zealand startede der en udvikling med informatik i skolen i 2008; det internationalt mest markante resultatet af dette initiativ er en fantastisk ressource, CS Unplugged, med materialer til undervisning i informatik uden en computer (Bell, 2018). Tilsvarende materialer er udviklet i UK (CS4FN, 2018).

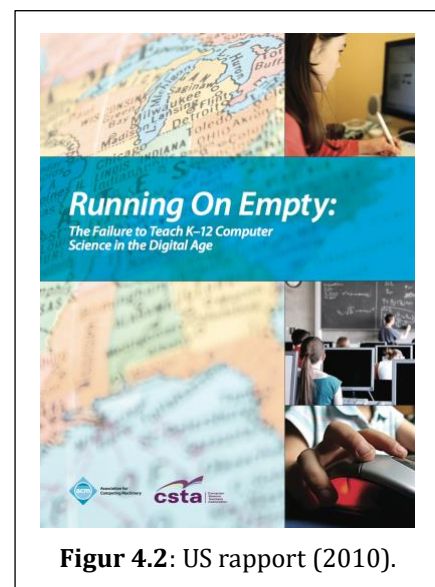
Det var også i 2008, at en kontorchef i UVM bad fagkonsulenten for it-fagene om at analysere den kritiske situation for it-fagene i gymnasiet efter reformen i 2005; der var mange it-valgfag, men med den nye studieretningsstruktur var der stort set ingen elever, der valgte fagene (ca. 1% af en årgang). Analysen førte til en rapport med anbefaling om at rydde op i porteføljen af fag og lave et nyt almindende it-fag, informatik (og et mere specialiseret programmeringsfag til de særligt interesserede). Informatik var prøvefag i perioden 2011-2015. Faget blev permanentgjort i foråret 2015, og med gymnasireformen året efter blev der skabt nye rammer for fagets placering i ungdomsuddannelserne. Vi vender tilbage til udviklingen i Danmark i afsnit 4.2

USA

I 2010 udgav ACM³ og CSTA⁴ i fællesskab rapporten *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. Rapporten indeholder et nationalt Call to Action:

No other subject will open as many doors in the 21st Century, regardless of a student's ultimate field of study or occupation, as computer science. At a time when computing is driving job growth and new scientific discovery, it is unacceptable that roughly two-thirds of the entire country has few computer science standards for secondary school education, K-8 computer science standards are deeply confused, few states count computer science as a core academic subject for graduation, and computer science teacher certification is deeply flawed. These are national failings and ones that we can ill afford in this digital age.

Parents must ask difficult questions about how computer science is being introduced to their children in K-12 education and demand that schools move beyond the current basic technology literacy curriculum. Policy makers at all levels need to review how computer science is treated within existing policy frameworks and schools, and ensure that engaging computer science courses based on fundamental principles of the discipline are part of the core curriculum. Now is the time to revitalize K-12 computer science education and ensure universal access to computer science courses by making it one of the core academic subjects students require to succeed in the 21st Century.



Figur 4.2: US rapport (2010).

Rapporten, og et systematisk lobbyarbejde, specielt i Washington D.C., fik daværende præsident Obama til at sætte emnet på den politiske dagsorden med større og større vægt gennem de sidste fire år af sin embedsperiode.

Kulminationen kom i Obamas sidste State of the Union-tale i januar 2016, hvor han annoncerede et massivt initiativ på området. Initiativet, *CS For All* (White House, 2016), blev annonceret et

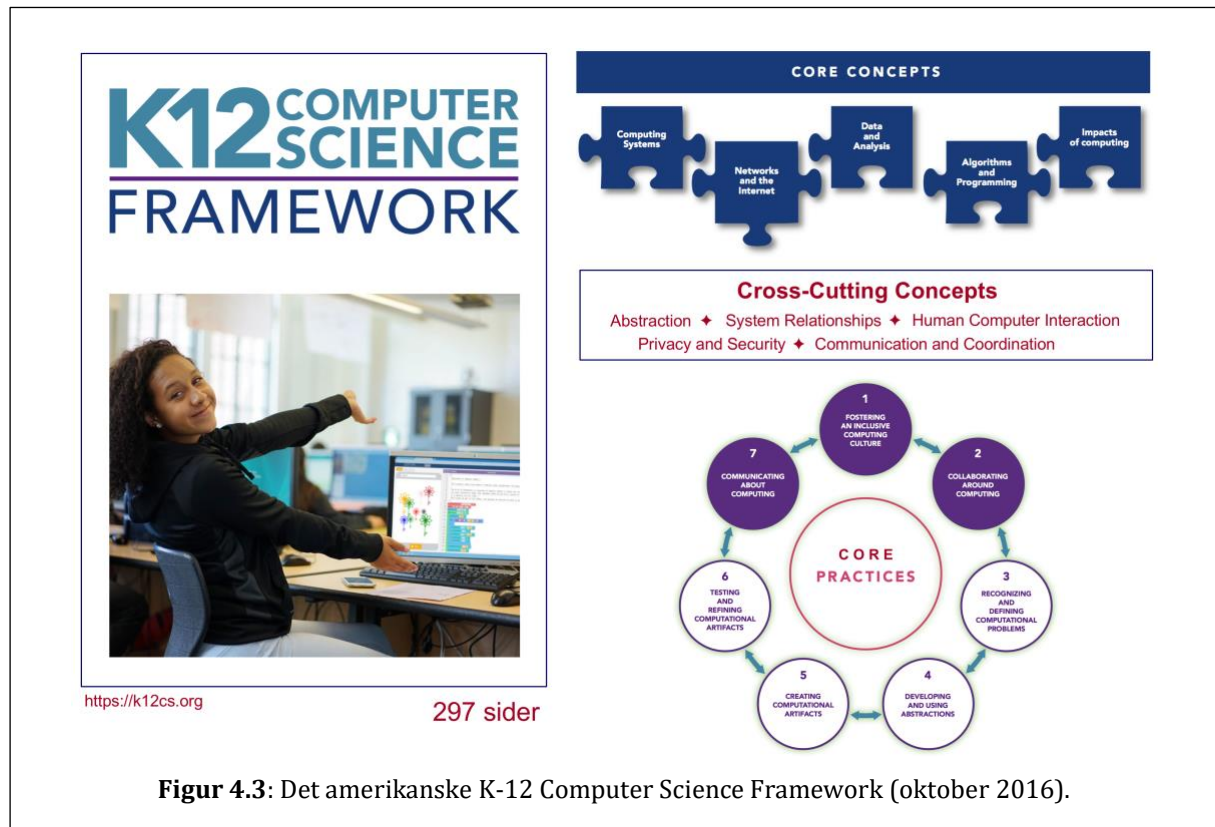
³ ACM, Association for Computing Machinery, er verdens største interesseorganisation for computing professionals, specielt inden for forskning og uddannelse. Det er bl.a. ACM, der uddeler Turingprisen, datalogiens Nobelpris.

⁴ CSTA, Computer Science Teachers Association, er en global forening for undervisere i computer science. Foreningen blev startet i 2004 af ACM.

par uger senere, den 30. januar 2016, og her signalerede Obama vigtigheden af området ved at udmelde et ambitionsniveau på 4 mia. USD.

Med den nye præsident er ambitionsniveauet blevet reduceret, men initiativet lever i bedste velgående i form af et nationalt konsortium (CSforALL, 2018).

I oktober 2016 publicerede ACM, Code.org, CSTA m.fl. *K-12 Computer Science Framework* (K12CS, 2016), et dokument på 297 sider med et substantielt rammeværk for CS curricula i skolen (figur 4.3).



Figur 4.3: Det amerikanske K-12 Computer Science Framework (oktober 2016).

I USA har de enkelte stater autonomi ift. curriculum i skolen (nogle gange er autonomien helt ude på skoledistriktsniveau); derfor har man ikke et nationalt curriculum, men i stedet et rammeværk (framework), som konkrete curricula kan laves ud fra.

Nogle store amerikanske skoledistrikter var tidligt ude med en ambitiøs politik på området. Som det første skoledistrikt annoncerede Chicago Public Schools i marts 2014, at de ville etablere en "pipeline" for CS-uddannelse i hele K-12 (Chicago CS, 2014).

I juni 2015 fulgte San Francisco efter med følgende annoncering (SF CS, 2015): *Within a few years, every single student in the San Francisco Unified School District will be studying computer science, at all levels.*

I september 2015 var det New York Citys borgmester, De Blasio, der annoncerede en 10-års deadline for, at alle skoler skulle undervise i CS for alle elever (NYC CS, 2015). Over en 10-års periode har NYC budgetteret 81 mio. USD til formålet, svarende til godt en halv mia. kr.

Investeringen på godt en halv mia. kr. er bemærkelsesværdig, idet NYC's skoledistrikt er på størrelse med Danmark.

Den nationale udvikling i USA drives af CSforALL-konsortiet, og med mere end 40 stater ombord har CSforALL god vind i sejlene, og mange stater/skole-distrikter er fulgt i sporene af Chicago, San Francisco og New York City.

Vi kan i Danmark lære meget af udviklingen i USA, specielt mht. implementeringsprocesser, og det er man i Undervisningsministeriet opmærksom på, idet ministeriet tiltrækker amerikanske nøglepersoner som keynotes til den årlige Læringsfestival, f.eks. Chris Stephenson og Leigh Ann DeLyser, der begge var dybt involveret i tilblivelsen af rapporten *Running on Empty* (figur 4.2). Chris Stephenson var grundlægger og leder af CSTA de første ti år af organisationens levetid (2004-2014) og er nu leder af Googles CS Education Strategy samt formand for ACMs Education Board; i 2016 fik hun ACM Presidential Award for sit virke. Leigh Ann DeLyser er en af de to ledere af CSforALL-konsortiet og samtidig forsknings- og uddannelsesleder for initiativet i NYC.

UK

Som nævnt indledningsvis udfører græsrodsbevægelsen CAS⁵ et kæmpe arbejde, specielt har professor Simon Peyton Jones, der er en meget anerkendt forsker, helt fra starten været en utrættelig pioner (Peyton Jones er formand for CAS).

I 2010 lykkedes det CAS at alliere sig med Royal Society, som i 2012 udgav rapporten *Shut down or restart? The way forward for computing in UK schools*. I rapportens forord skriver præsidenten for Royal Society, Sir Paul Nurse:

Computing is of enormous importance to the economy, and the role of Computer Science as a discipline itself and as an 'underpinning' subject across science and engineering is growing rapidly. [...]

The Government has recognised the need for more high quality CS teaching, and has committed to exploring the best ways to achieve this. Our report therefore provides a particularly timely source of evidence that will be needed to inform important policy decisions relating to the National Curriculum in England and to support a drive towards improving Computing education throughout the UK.

[...] I hope that the Royal Society's work will provide a solid foundation which the community can build on, to ensure that the next generation of young people in this country can be creators of technology – not just consumers of it.



Figur 4.4: Britisk rapport (2012).

⁵ CAS, Computing At School, er en britisk organisation af mere end 29.000 frivillige, der arbejder for at få computing etableret som et obligatorisk skolefag og understøtte implementering heraf.

Rapporten fra Royal Society havde en markant effekt, og allerede i september 2013 vedtog det britiske undervisningsministerium et nationalt curriculum for computing (UK CS, 2013).

Allerede i 2014 implementerede man den politiske beslutning med et "big bang" (dvs. synkront på alle niveauer og uden ordentlig uddannelse af lærerne), men det var præmaturligt, og man har siden fortrudt "hastværket".

I en nylig rapport fra Royal Society er erfaringerne fra de første to år dokumenteret (Royal Society, 2017). Erfaringerne viser bl.a., at implementeringen i UK er skrøbelig – først og fremmest på grund af manglende lærerkompetencer på alle niveauer, og kunne man gøre det om, ville man *ikke* have startet med det omtalte "big bang". Rapportens anbefalinger er gengivet i figur 4.5.

1. Der er behov for monitorering af, om og hvordan skoler underviser i computing.
2. Der skal sikres adgang til computing for alle elever.
3. Forskning i fagets didaktik skal også fokusere på kønsproblematikken.
4. Stats- og fondsstøttede tiltag skal prioritere og evaluere indflydelsen på kønsbalancen.
5. Staten bør indføre og finansiere kvalitetssikret efteruddannelse for alle lærere svarende til kompetencekrav i matematik.
6. Staten skal samarbejde med universiteterne om at udvikle og akkreditere faglig læreruddannelse og gøre det muligt for folk med forskellig baggrund at blive computing-lærere.
7. De videregående uddannelsesinstitutioner skal promovere karrierer i computing-undervisning til en bred skare af studerende.
8. Erhvervslivet og universiteterne skal støtte og opmuntre til forgrenede karrierer, hvor medarbejdere, som ønsker at undervise, kan få mulighed herfor.
9. Staten og erhvervslivet skal spille en aktiv rolle for at forbedre vedvarende efteruddannelse for computing-lærere.
10. Erhvervslivet og non-profit organisationer skal samarbejde med relevante parter for at kunne tilbyde undervisningsstøtte til skoler og lærere.
11. Der skal tilvejebringes en national strategi for (i) etablering af en langsigtet forskningsagenda for computing-didaktik, (ii) opbakning til et sådant program fra et antal interessenter, (iii) udvikling af kapacitet til at varetage forskningen og (iv) effektiv deling af viden mellem computing-didaktiske forskere, lærere og lærer-lærere.
12. Relevante forskningsråd og andre bidragydere skal arbejde på at adressere de forskningsprioriteringer, der er identificeret i rapporten.



Figur 4.5: Anbefalinger fra *After the reboot: computing education in UK schools* (pp. 8-11)

Europa

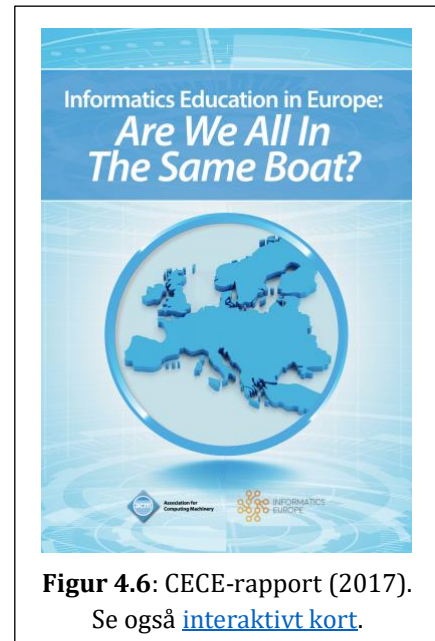
Der er en tilsvarende udvikling i gang i mange andre europæiske lande, f.eks. Italien (Nardelli et al. 2018), Sverige (Heintz & Manilla, 2018) og naturligvis Danmark (se afsnit 4.2).

En europæisk kortlægning af udbredelsen af informatik i skolen blev foretaget i 2015-2016 af Committee on European Computing Education (CECE). Kortlægningen er baseret på data fra 53 lande ("administrative enheder").

Kortlægningen dækker såvel informatik som mere almene it-kompetencer (digital literacy) samt vilkår for læreruddannelse.

Kortlægningen er tilgængelig som en rapport (CECE, 2017) og som et interaktivt kort, hvor man kan sammenligne situationen i forskellige lande ift. 26 parametre (CECE's Map, 2017).

Rapporten er en europæisk pendant til de to rapporter fra henholdsvis USA (*Running on Empty*) og UK (*Shut down, or restart?*).



Figur 4.6: CECE-rapport (2017).
Se også [interaktivt kort](#).

Hovedkonklusionerne i rapporten er, at elever, med få undtagelser, først har adgang til informatik/CT på sekundært niveau (udskoling og gymnasiet); at tilbuddene på sekundært niveau er temmelig uensartede og kun obligatoriske i en femtedel af landene (22%). I de tilfælde hvor faget er valgfrit, er søgningen deprimerende lav (dog få tal).

Generelt er det for gymnasieniveau muligt at uddanne sig til informatikunderviser efter samme standarder som for andre gymnasiefag, men få gør det (formodentlig pga. mere attraktive muligheder i erhvervslivet).

I mange lande gælder, at lave krav (om nogen) til efteruddannelse for at kunne undervise i informatik underminerer anerkendelsen af fagligheden og dermed indførelsen heraf som en ligeværdig faglig disciplin.

Sammenfattende konkluderes, at informatik er *meget langt fra* at være sidestillet med andre sciencefag, og i mange lande kan elever færdiggøre sekundær uddannelse uden nogensinde at have været eksponeret for informatikkens basale begreber og principper.

Rapporten indeholder en række anbefalinger, som er uddybet og kvalificeret i *Informatics for All: The Strategy* (Caspersen, 2018); Informatics for All-strategien er en direkte parallel til Obama-administrationens CS For All-initiativ, og på europæisk niveau samarbejdes med nogle af de amerikanske personer og organisationer, som var centrale aktører i det lobbyarbejde, der førte til CS For All. Informatics for All-strategien uddybes i afsnit 5.1.

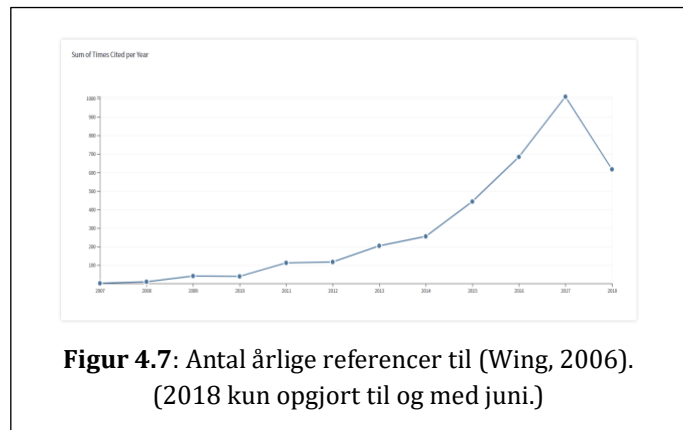
Forskning

De mange politiske tiltag modsvares af en stor mængde forskning og udvikling i almengørelse af informatik/CT som en relevant faglighed for alle.

Det er svært at give et kvalificeret estimat for antallet af forskningspublikationer om CT, men et groft slag på tasken fås ved at se, hvor mange der refererer til Wings skelsættende artikel (Wing, 2006).

P.t. har godt 4.000 publikationer refereret til denne artikel; fordelingen over år fremgår af figur 4.1.

Som det fremgår, er forskning i CT omfattende og støt stigende (2018 er kun indekseret til og med juni, men fremskriver vi konservativt de ca. 600 citationer til resten af året, ser vi igen en stigning på omkring 20% sammenlignet med året før).



Det er imidlertid et broget landskab, der bærer præg af, at man først inden for de seneste ti år for alvor er begyndt at opdyrke det. Der er en lang tradition for forskning inden for computing education i videregående uddannelse, men bortset fra nogle enkelte undtagelser er forskning i almengørelse af informatik/CT relativt ny.

Tidligere i år udkom den første internationale bog om Computer Science Education målrettet nuværende eller kommende lærere i K-12 (Sentance et al., 2018). 22 internationalt førende forskere har bidraget med kapitler om forskellige aspekter af undervisning i informatik/CT. I bogens forord skriver Simon Peyton Jones:

We must confront the deep questions: what should we teach, and how should we teach it? Other subject disciplines have had centuries to develop answers to these questions (and are still debating them) but computing has not. Yet the need is pressing, because educators across the world are hungry for an inspiring vision to frame their teaching, for pedagogies that provably work and for assessments that validly measure learning and progress. [...] I am absolutely delighted to see such a substantial contribution in such an under-served space.

Pointen, som Peyton Jones fremhæver, er, at dette jomfruelige område naturligvis er et 'under-served space'.

Internationalt er der en del årlige konferencer om 'computing education' (f.eks. SIGCSE, ITiCSE, ICER, Koli, ACE, WiPSCE, ISSEP, CSERC, SACLA, CompEd, FIE, PPIG, RESPECT, LaTICE, IDC og FabLearn). Hertil kommer en lang række tidsskrifter.

Appendiks A.3 er et kondenseret litteraturreview af godt 70 udvalgte publikationer. Hovedkonklusionen er, at der er mange interessante forsknings- og udviklingsaktiviteter, som bl.a. forsøger at

- finde frem til CT's væsen og identitet
- foreslå curricula og konkretiseringer heraf
- foreslå og afprøve konkrete undervisningsforløb og tilgange
- foreslå og afprøve didaktiske principper
- foreslå og afprøve vurderings- og evalueringskriterier og -metoder.

På trods af de mange forsknings- og udviklingsaktiviteter, som har set dagens lys de seneste år, er der endnu ikke klare resultater og evidens for bestemte tilgange, som vi uden videre kan støtte os til.

Der er behov for meget mere forskning og udvikling, der kan belyse og informere området og på sigt skabe evidens for, *hvad* der skal undervises i *hvornår* og *hvordan*.

4.2 CT i dansk uddannelse

Arbejdet med indførelse af et alment informatikfag i ungdomsuddannelserne begyndte for 10 år siden. Fra 2011 kom der et forsøgsfag, og i foråret 2015 blev faget permanentgjort; med gymnasiereformen året efter blev faget obligatorisk i HHX, men desværre kun semi-obligatorisk i STX og HTX.

Siden 2017 har det været Danske Gymnasiers politik, at informatik skal være obligatorisk. Et hav af rapporter og politiske anbefalinger sender samme signal, men ikke kun for ungdomsuddannelserne; anbefalingen er, at informatik og CT gøres obligatorisk på alle uddannelsestrin (Vækstråd, 2016; Vækstpanel, 2017).

Med *Strategi for Danmarks digitale vækst* (Regeringen, 2018) har regeringen taget første skridt i retning af at realisere anbefalingerne, idet der udvikles og over de næste tre år afprøves et it-fag (teknologiforståelse) i folkeskolens 1.-9. klasse.

Videregående uddannelser

Uddannelses- og Forskningsministeriet anbefaler i rapporten *Universitetsuddannelser til fremtiden* (UFM, 2018), at universiteterne indarbejder digitale kompetencer i alle uddannelser, herunder kompetencer som kodning og dataanalyse, samt at studerende trænes i at forholde sig til etiske og samfundsmæssige konsekvenser af den teknologiske udvikling. Rapporten understreger vigtigheden af, at de nye elementer *integreres* i uddannelserne. Da der allerede findes et betragteligt antal it-specialistuddannelser på universiteterne, favner UFM således til fulde den tostrengede strategi: *som* og *i* uddannelse.

Ungdomsuddannelserne

Også i ungdomsuddannelserne er man i færd med at realisere den tostrengede strategi; her *som* og *i* fag. Med gymnasiereformen i 2016 blev krav til digitale kompetencer skrevet ind i samtlige mere end 200 læreplaner for fag i gymnasiet (ikke altid lige vellykket, men det er en start).

Læreplansændringerne, sammen med gymnasiernes ønske om at omfavne informatik og digitale kompetencer, banede vejen for pilotprojektet *CT i gymnasiefag* (CCTD, 2018a; CCTD 2018b) med deltagelse af ni midtjyske gymnasier og Center for Computational Thinking & Design ved Aarhus Universitet.

Projektet, der var finansieret af Region Midtjylland, viste meget lovende resultater af forsøg med at indføre CT i biologi, kemi, bioteknologi og samfundsfag (Musaeus & Musaeus, 2019).

Projektet har resulteret i et opfølgingsprojekt i regi af Den Midtjyske Teknologipagt. Derudover har projektet givet genlyd i andre kredse; således har både Region Nordjylland og Region Syddanmark nu tilsvarende projekter, og Danske Science Gymnasier (DASG) har fået midler fra VILLUM FONDEN til et projekt om CT i matematik og sciencefag (se også afsnit 5.3).

I forbindelse med udviklingen i ungdomsuddannelser, specielt med informatikfaget i gymnasiet, er der udviklet en række fagdidaktiske principper, som kan støtte tilrettelæggelse af undervisning og udvikling af undervisnings- og evalueringsmateriale (Caspersen & Nowack, 2013; Caspersen, 2017b).

Grundskolen

I grundskolen har der de seneste år været forsøgsprojekter i retning af at favne digitale kompetencer, specielt fablab og CodingClass (en skolevariant af fritidstilbuddet Coding Pirates).

Makerspaces

I Skandinavien og særligt i Danmark har makerspaces og introduktionen af digitale teknologier i undervisningen været udforsket af Aarhus Universitet og en række danske kommuner. Hvor STEM-fokus står stærkt i landene omkring os, står makerspaces i Danmark frem med en kraftig toning af såvel et dannelses- som uddannelsesaspekt, som ikke blot knytter an til STEM fagene.

I en dansk kontekst definerer vi et makerspace som (Smith et al., 2016):

”Et hybrid læringslaboratorie, der kombinerer digital fabrikation, design tænkning, kollaborativ idegenerering og skabelse i løsningen af komplekse samfundsmæssige udfordringer. En indsats rettet mod ’design dannelse’ i curriculum-båret uddannelse”.

I danske makerspaces har børnene selv mulighed for at rammesætte problemer i verden, arbejde systematisk med idégenerering og produktudvikling for senere at kunne omsætte deres idéer og design til digitale artefakter.

Gennem hands-on og mind-in processer får børnene en håndgribelig og autentisk forståelse for digitale teknologier og afprøver dermed selv den computationelle tænkning i praksis. Til forskel fra det stærke STEM-fokus i andre lande har danske fablabs fokus på at kombinere STEM uddannelse med en digital dannelse. Danske skoleelever lærer at designe og kode egne digitale artefakter, og igennem den proces lærer de også at *afkode* de digitale artefakter, som de omgiver sig med. Derigennem kan eleverne opnå computationel empowerment, som betegner elevernes evne til at medskabe den fremtid, der opstår gennem design med digitale teknologier. Det indbefatter en kritisk og nysgerrig stillingtagen til teknologien og en evne til at forstå den effekt, som digitale artefakter har på børnene selv, deres nærmeste og samfundet.

I fablabs regi defineres den procesforståelse, der optrænes i fablabs som *design* (Smith, Hjorth & Iversen, 2016). Følgeligt udvikles elevernes designkompetence eller designdømmekraft som et led i undervisningen (Iversen et al., 2017). I andre sammenhænge refereres disse designprocesser som en bredere brug af begrebet ’engineering’ (Engineer the Future).

På verdensplan står den danske version af makerspaces meget stærkt. På de danske FabLearn-konferencer udveksler 300-400 eksperter, undervisere og forskere årligt idéer, tanker og undervisningsforløb. Det er den største nationale konference i verden. Danske universiteter har som de første i verden udviklet et 5 ECTS undervisningsforløb til undervisere på grund- og ungdomsuddannelserne i digital fabrikation og designtænkning. Danske forskere er repræsenteret i de mest citerede publikationer om digital fabrikation i uddannelser, ligesom danske forskere i øjeblikket står for opkvalificering af undervisere og forskere i nogle af vores europæiske nabolande.

I 2018 fik det danske FabLearn-projekt den prestigefyldte pris for 'Global Excellence Organizational Leadership' af Stanford University for evnen til at kunne skalere fablabs og bringe forskningsviden ud i skolemiljøer i Danmark og Europa.

På trods af de gode resultater, som FabLearn-projektet har opnået i Danmark, mangles der stadig en stor viden om væsentlige elementer af forholdet mellem makerspaces og undervisning. Særligt savnes der en forskningsbaseret viden om:

- hvordan designkompetence og elevens evne til at skabe med digital teknologi kan evalueres
- hvordan underviseres evne til at arbejde med komplekse designproblemer kan udvikles
- hvordan nye teknologier, hvor eleverne selv kan lære at udvikle og forstå kunstig intelligens, kan udvikles
- hvordan computermodellering kan indgå i FabLearn-undervisningen
- hvordan FabLearn kan introduceres til skoler, der ikke har særlige forudsætninger for at inddrage digitale teknologier i undervisningen, og
- hvordan man implementerer 'deep making' i undervisningen (i modsætning til 'shallow making' eller såkaldt "skoæske-kreativitet").

Coding Class

Coding Class, der er en skolevariant af fritidstilbuddet Coding Pirates, blev gennemført første gang i 2016-2017 i Københavns og Vejle Kommune og havde bl.a. fokus på CT.

Indholdsmæssigt har der været fokus på spiludvikling i Scratch (Resnick et al., 2007). De deltagende klasser har haft fire sessioner med Coding Class-instruktører, så det er et meget begrænset grundlag at evaluere og vurdere effekter på.

Evalueringen viser, ikke overraskende, at det er et helt nyt og uopdyrket område (Hansbøl et al., 2017, p. 19):

CT i skolen er en ny dagsorden, der først er ved at vinde terræn og etablere sig.

Da der ikke har været formuleret læringsmål, kan evalueringen ikke foretages ift. sådanne. I stedet foretages en vurdering af initiativet ift. fire formål (side 66-69):

- at fremme elevernes forståelse for den digitaliserede verden, der omgiver dem nu og i fremtiden (læringsudbytte)
- at fremme elevernes interesse for it (motivation)
- at igangsætte undervisning, der fremmer elevernes evne til selv at arbejde mere kreativt og skabende med it i grundskolen (pædagogik), og

- at sætte fokus på it, kodning og computationel tænkning som vidensdomæne i grundskolen (det videre perspektiv).

Vurderingen, der strækker sig over fire sider, peger både på udfordringer og potentialer, men afholder sig viseligt fra entydige konklusioner.

Projektet gennemføres også i 2018-2019 med yderligere deltagelse fra Fredericia, Furesø, Høje Taastrup, Odense og Aarhus Kommune.

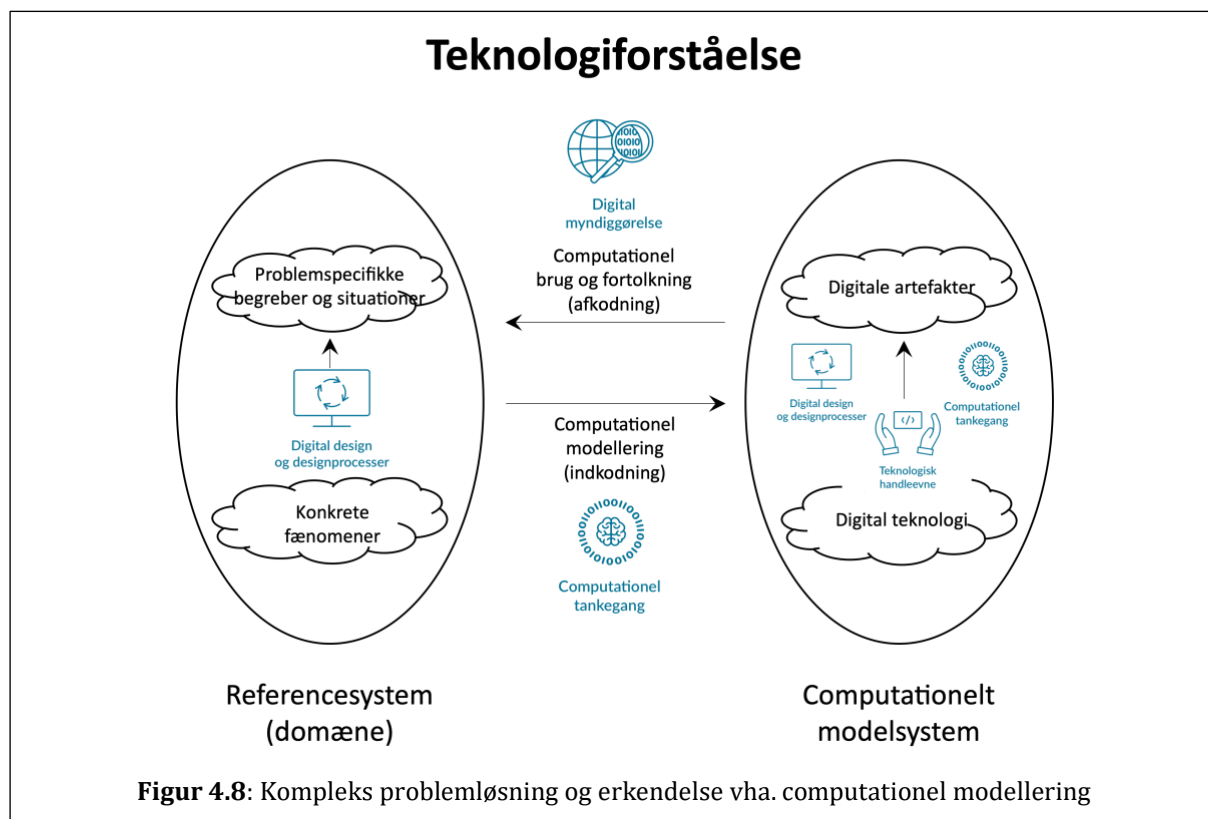
4.3 Teknologiforståelse

I 2018 er der udviklet et nyt forsøgsfag, teknologiforståelse til den danske folkeskoles 1.-9. klasse. Faget kombinerer CT med Iversens vision om Computational Empowerment (afsnit 2.4).

Formålet med faget er, at eleverne skal udvikle faglige kompetencer og opnå færdigheder og viden, således at de konstruktivt og kritisk kan deltage i udvikling af digitale artefakter og forstå deres betydning.

Elevernes mestring af faget fordrer en beherskelse af digitale designprocesser og af digitale teknologiers sprog og principper med henblik på iterativt og i samarbejde at kunne analysere, designe, konstruere, modificere og evaluere digitale artefakter til erkendelse og løsning af komplekse problemer.

Målene for og indholdet af faget teknologiforståelse kan overordnet beskrives ift. figur 4.8 (sammenlign med figur 1.3 side 8). Overordnet set afspejler figur 4.8 det at tage en del af verden og repræsentere den i computeren med henblik på at skabe et digitalt artefakt, der adresserer et (komplekst) problem eller bidrager til erkendelse. 'Learn to Compute' eller 'Compute to Learn'.



Beskrivelsen af fagligheden er struktureret i fire kompetenceområder:

- Digital myndiggørelse
- Digital design og designprocesser
- Computational tankegang
- Teknologisk handleevne.

Digital myndiggørelse omhandler kritisk, reflektiv og konstruktiv undersøgelse og forståelse af digitale artefakters muligheder og konsekvenser. Dette svarer groft sagt til pilen Computational brug og fortolkning i figur 4.8.

Digital design og designprocesser omhandler tilrettelæggelse og gennemførelse af en iterativ designproces under hensyntagen til en fremtidig brugs kontekst. Dette svarer groft sagt til aktiviteterne i referencesystemet samt design og evaluering af prototyper i det computationelle modelsystem.

Computational tankegang omhandler analyse, modellering og strukturering af data og dataproceser udtrykt i abstrakte modeller (f.eks. algoritmer, data- og interaktionsmodeller). Dette svarer til modelleringsprocesser i det computationelle modelsystem samt pilen Computational modellering i figur 4.8.

Teknologisk handleevne omhandler mestring af computersystemer, digitale værktøjer og tilhørende sprog samt programmering. I figur 4.8 svarer det groft sagt til læren om og arbejde med konkrete digitale teknologier til frembringelse af digitale artefakter.

Ved design af fagligheden er der tilstræbt "balance" i figuren både ift. at have blik for den verden, vi lever i og ønsker at leve i (referencesystemet), og de muligheder, som digitalisering repræsenterer. Der er store muligheder i det digitale, men også store spørgsmål alle skal klædes på til at kunne stille, diskutere, have informerede holdninger til og kunne være med til at besvare. Ambitionen er kort sagt at udvikle elevernes computationelle sprog (den fjerde sprogform, afsnit 1.3) og samtidig udvikle elevernes kritiske sans ift. digital teknologi og digitale artefakter.

Ligeledes er der en væsentlig balance mellem de fire kompetenceområder, som på afgørende vis beriger hinanden og er hinandens forudsætninger: Uden computationel tankegang bliver digital myndiggørelse hul og overfladisk, og uden digital myndiggørelse og digital design bliver computationel tankegang og teknologisk handleevne løsrevet fra anvendelsesperspektivet, som er afgørende for en kritisk tilgang til konstruktion af digitale artefakter – en tilgang, hvor der er fokus på værdiskabelse i referencesystemet fremfor ensidigt fokus på teknologiske muligheder.

Fagligheden bør også indgå som obligatorisk komponent i ungdomsuddannelser og videregående uddannelser, naturligvis afstemt i kompleksitet og abstraktionsniveau til det pågældende uddannelsesniveau og tonet til den faglige kontekst. Med informatikfaget i ungdomsuddannelserne er vi allerede godt på vej, men dette fag bør justeres i lys af de nye tiltag i grundskolen. Og i afsnit 5.2 skitseres en tentativ ramme for CT i videregående uddannelser i Danmark.

5. CT i uddannelse – potentialet

Dette afsnit, der er en uddybning af første afsnit, giver et overblik over CT's store potentiale inden for uddannelse. Potentialet beskrives gennem tre konkrete initiativer som på forskellige niveauer søger at forme en udvikling, hvor informatik/CT udvikles som et fundamentalt og selvstændigt område, og hvor CT udvikles som en integreret disciplin i alle andre fagområder.

Det vil føre alt for vidt med en detaljeret redegørelse af potentialet for CT i uddannelse, men for at give en fornemmelse heraf præsenterer vi tre konkrete initiativer, som på forskellige niveauer søger at forme udviklingen og gøre det muligt at realisere potentialet.

Det første, *Informatics for All*, er et politisk lobbyinitiativ på EU-niveau, det andet er et konkret dansk arbejde med at udvikle en taksonomi for CT i universitetsuddannelser, og det tredje er en beskrivelse af det særlige potentiale, der er for ny form for erkendelse gennem arbejde med computationelle modeller af komplekse dynamiske systemer.

5.1 Informatics for All – en tostrenget strategi

Informatics for All (Caspersen et al., 2018) er et initiativ udtænkt i fællesskab af ACM Europe og Informatics Europe. Formålet er at skabe behørig anerkendelse af informatik som en fundamental disciplin for uddannelse i det 21. århundrede.

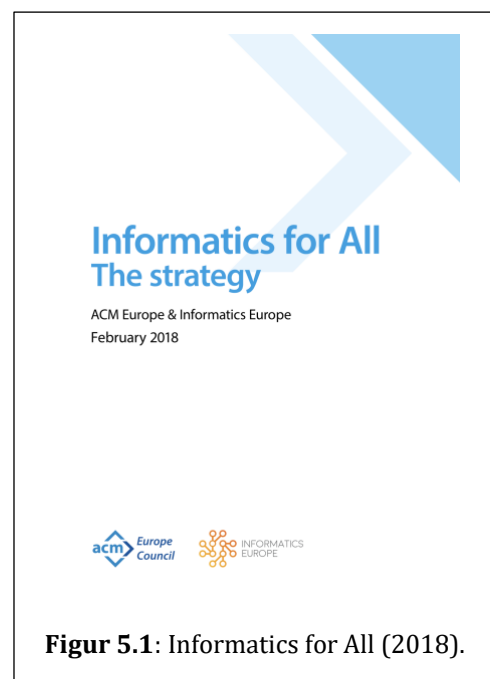
Man kan karakterisere initiativet som en europæisk pendant til Obama-administrationens *CS For All*-initiativ (White House, 2016; CSforALL, 2018).

Informatics for All blev lanceret i Bruxelles i marts 2018, og der er løbende drøftelser med repræsentanter for EU-kommissionens direktorat for uddannelse og kultur. Der planlægges med nye arrangementer i Bruxelles i 2019.

Strateginotatet (Caspersen et al., 2018) fremhæver behovet for en tostrenget strategi for informatik på alle uddannelsesniveauer: informatik som en fundamental og selvstændig disciplin og informatik/CT som en tværgående disciplin integreret i alle andre (som fag og i fag).

For at nå dette mål er der for begge strenge behov for at udforske, hvad og hvordan man skal undervise i dette (curriculum, metoder og værktøjer), og hvordan man kan uddanne lærere.

Rapporten præsenterer i alt otte anbefalinger inden for tre områder: curriculum, lærere og forskning. Anbefalingerne repræsenterer en stor og global udfordring, hvor der er behov for såvel nationalt som internationalt samarbejde. På europæisk plan er det intentionen at præge såvel nationalstaternes som EU's politikker og anbefalinger.



Figur 5.1: Informatics for All (2018).

Endvidere arbejdes der på at få informatik gjort til en del af OECDs Science PISA-test som et fjerde scienceområde: 'Physical systems', 'Living systems', 'Earth and space systems' og som det fjerde 'Digital systems'.

5.2 Tentativ ramme for CT i universitetsuddannelser

I regi af It-vest arbejder Syddansk Universitet, Aalborg Universitet og Aarhus Universitet på at formulere en dansk taksonomi for CT i universitetsuddannelser.

Taksonomien er bevidst holdt på et forholdsvis generelt niveau, idet tanken er, at taksonomien skal være en ramme på tværs af uddannelser, men en ramme som også skal specialiseres til den konkrete kontekst.

Der arbejdes "bottom-up" med dette, i første omgang med udgangspunkt i hhv. humanistiske og samfundsvidenskabelige cand.it.-uddannelser, men det er tanken efterfølgende at udbrede perspektivet til at dække universitetsuddannelser generelt.

Foreløbig har arbejdet resulteret i formulering af (tentative versioner af)

1. Forudsætningskrav til studerende (og tilbud, så de kan opfyldes/efterleves)
2. CT-kompetencebeskrivelse for humanistiske it-uddannelser
3. Principper for organisatorisk integrering af CT-faglighed
4. Tentativ taksonomi/ramme for CT i humanistiske it-uddannelser.

Appendiks A.4 indeholder en mere detaljeret beskrivelse af hvert af de fire elementer af rammen.

5.3 Computationelle modeller af dynamiske systemer

Vi vil her konkretisere og uddybe det potentiale eller den vision, som kort blev antydnet i afsnit 1.3 Den fjerde sprogform. Visionen, der er undfanget og udviklet i miljøet omkring Seymour Papert på MIT, er nærmere beskrevet i appendiks A.2

Finansieret af Villum Fonden og Region Midtjylland, og i regi af Center for Computational Thinking & Design og Danske Science Gymnasier, er der netop igangsat projekter der udforsker og udbreder CT i gymnasiefag generelt og specifikt i matematik og sciencefag.

Der er mange mulige fortolkninger af "CT i gymnasiefag", men i projekterne arbejdes der især med potentialet i ny erkendelse og nye erkendelsesformer gennem udvikling, justering og raffinering af computationelle modeller af komplekse dynamiske systemer (Musaeus & Musaeus, 2019).

Der er et særligt potentiale ved anvendelse af digitale modeller som supplerende element i undervisningen. Kort fortalt handler det om, at informatikkens sprog gør det muligt at automatisere (og dermed eliminere) meget af den kompleksitet, der er i matematisk sprog. For at modellere dynamiske systemer matematisk, skal man have et temmelig rigt matematisk sprog, der bl.a. omfatter differentiaalligninger. Det er de færreste, der mestrer matematikken på det niveau, og dermed bliver modellerne utilgængelige for de fleste.

Et essentielt og helt afgørende aspekt af informatikken er muligheden for "automatisering af abstraktioner". Altså ikke bare automatisering af konkrete processer, men også af abstrakte, mentale processer. Den finurlige formulering "automatisering af abstraktioner" er dyb; den dækker over, at man kan udforme digitale værktøjer, som tillader at "ligeegyldige detaljer" fejles under gulvtæppet og håndteres af maskineriet, således at man på øverste niveau kan koncentrere sig om at manipulere de essentielle fænomener og begreber, som man i en given sammenhæng er interesseret i.

Vi illustrerer med en uddybning af bølgeeksemplet fra afsnit 1.3. Hvis man vil modellere bølger, fordrer det en relativ kompleks matematik. Årsagen er, at den matematiske model dels har et globalt systemperspektiv, og dels at modellen eksplicit skal rumme dynamik, dvs. tilstandsændringer over tid (sidstnævnte modelleres vha. førromtalte differentiaalligninger). Ved hjælp af en digital model i et passende værktøj kan man imidlertid modellere bølger ud fra et lokalt perspektiv, hvor man for at modellere en enkelt partikels adfærd groft sagt ikke skal kunne andet end at lægge to tal sammen og dividere med to (Wilensky, 1997; Wilensky, 2006a). Håndteringen af mange partikler og tilstandsændringer over tid klares af maskineriet takket være en hensigtsmæssig "automatisering af abstraktioner" (Wilensky, 1999). Resultatet er, som det blev nævnt i afsnit 1.3, en simpel og transparent model, som er nem at forstå og dermed inviterer til "nærkontakt" i form af undersøgelse, justering af modellen ved simpel programmering, hypotesedannelse, eksperimenter, flere justeringer af modellen osv.

Generelt kan digitale modeller

1. supplere traditionelle modeller med dynamik
2. åbne for radikal ny erkendelse
3. stimulere en mere nysgerrig og undersøgelsesbaseret tilgang til læring og
4. simplificere modeller af komplekse systemer, således at disse bliver transparente og dermed tilgængelige for mange, mange flere.

En konsekvens er, at man kan undersøge, forstå og beskrive interessante og komplekse dynamiske systemer i en meget tidligere alder, fordi indgangstærsklen bliver meget lavere, vel at mærke samtidig med, at der bliver (væsentligt) højere til loftet.

Den modeldrevne tilgang, specifikt ved hjælp af såkaldt agent-baseret modellering, er i 2017 blevet afprøvet i et pilotprojekt med ni midtjyske gymnasier; tidlige idéer hertil kan findes i (Nowack & Caspersen, 2014). Tilgangen er baseret på forsknings- og udviklingsarbejde, der er foregået over de seneste 25 år på bl.a. Massachusetts Institute of Technology i Boston og Northwestern University i Chicago.

Faderen til visionen er Seymour Papert, og det er to af hans fremmeste studerende, der har realiseret visionen, henholdsvis Mitch Resnick (Resnick, 1992, 1996a, 1996b, 1996c) og Uri Wilensky (Wilensky & Resnick 1999; Wilensky, 2006b; Wilensky & Rand, 2015; Weintrop et al. 2016). Alex Repennings arbejde, der er rettet mod grundskolen, skal også nævnes i denne sammenhæng (Repenning, 2012).

Som nævnt er visionen nærmere beskrevet i appendiks A.2.

6. CT i det danske uddannelsessystem – muligheder og udfordringer

I dette afsnit diskuteres muligheder og udfordringer ved at implementere CT i det danske uddannelsessystem. CT bør tænkes både som uafhængigt fag og som en del af fagligheden i de eksisterende fag. Det kræver begrebsmæssigt arbejde at omtænke de eksisterende fag med CT, og det kræver det et stort arbejde at designe og udvikle undervisningsmateriale og træne lærere i at bruge CT i deres undervisning, herunder at udvikle en CT-didaktik. Sidst, men ikke mindst, vil skalering kræve langsigtet koordinering.

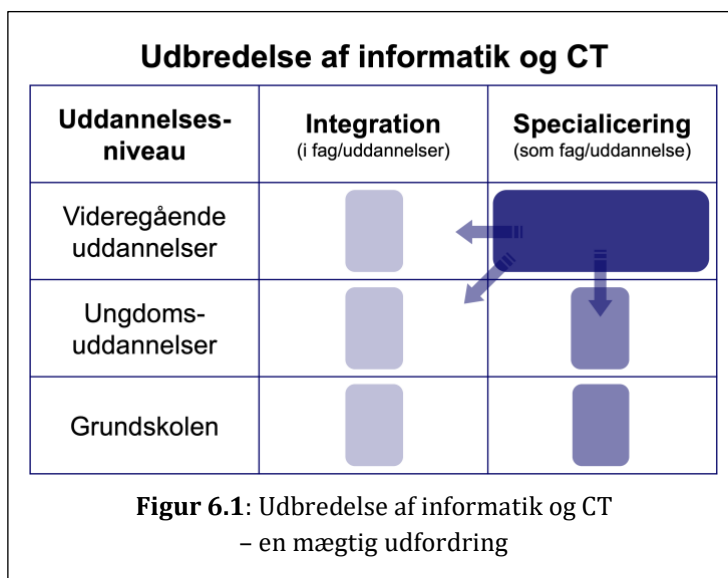
Afsnittet vil bl.a. berøre CT's rolle både som fag og i fag på alle uddannelsesstrin. Nødvendigheden af kompetence- og kapacitetsopbygning højt i fødekæden vil blive fremhævet. Desuden peges på udvikling og koordinering af nationalt samarbejde omkring informatik og CT, f.eks. i form af et nationalt videnscenter.

Også i Danmark breder der sig en erkendelse af vigtigheden af at indføre informatik/CT⁶ som et fag på alle uddannelsesniveauer: primært, sekundært og tertiært. På alle niveauer er det relevant med en tostrengt strategi med både et selvstændigt fag (uddannelser) og integrering i andre fag (uddannelser).

På de gymnasiale ungdomsuddannelser er udviklingen i gang med indførelse af faget informatik og krav om digitale kompetencer i alle fag. I folkeskolen er man, under betegnelsen teknologiforståelse, aktuelt i færd med at etablere forsøg med fagligheden både som og i fag.

Tilsvarende er der aktuelt et politisk initiativ fra Uddannelses- og Forskningsministeriet ift. at få digitale kompetencer integreret i alle videregående uddannelser og dermed supplere de mange specialistuddannelser, således at den tostrengede strategi også kan realiseres på videregående uddannelsesniveau.

Som omtalt i afsnit 2 repræsenterer udbredelsen af informatik en mægtig udfordring, og det er ikke kun en national men en global udfordring.



Udfordringen har mange aspekter, men de tre væsentligste er (Caspersen et al., 2018):

- Udvikling af curriculum og eksemplariske materialer
- Uddannelse af undervisere
- Forskning og fagdidaktisk indsigt.

⁶ I tråd med resten af rapporten benyttes informatik/CT som betegnelse for henholdsvis et selvstændigt fag på den ene side og den faglighed, der, jvf. den tostrengede strategi, skal integreres i alle andre fag/uddannelser på den anden side.

Hen over disse tre aspekter ligger den overordnede udfordring, det er, at gentænke informatik som et alment og fundamentalt fag for alle. Det er ikke trivielt at transformere et fag fra "kun" at være et forskningsområde og videregående specialistuddannelser for kommende professionelle it-folk til at være meningsfuldt og relevant for alle elever og studerende på alle uddannelsesstrin.

6.1 Udvikling af curriculum og eksemplariske materialer

Den store udfordring i at udvikle curriculum for en almen faglighed er at gentænke specialistfagligheden, så den bliver relevant for alle. En ting er at uddanne kommende professionelle inden for området, en anden er almen dannelse og uddannelse inden for området. Et væsentligt element af den udfordring er at udvikle en meningsfuld progression op gennem uddannelsessystemet, herunder effektive evaluerings- og prøveformer.

Et curriculum beskrives i en bestemt genre med målformuleringer på forskellige niveauer, men det er langt fra tilstrækkeligt ift. at sikre, at intentionerne i curriculum efterleves.

Specielt for et helt nyt fag, som stort set ingen i samfundet har viden om eller erfaring med, er det af afgørende betydning, at intentionerne udfoldes og konkretiseres gennem systematisk udvikling af eksemplariske materialer. Dvs. materialer, der efterlever og konkretiserer intentionen i det curriculum, der er manifesteret i aktuelle styredokumenter (læreplaner, læseplaner og studieordninger), specielt ved at integrere diverse læringsmål i helstøbte forløb og elevaktiviteter.

Sammen med eksemplariske materialer er grundig uddannelse/efteruddannelse af undervisere helt afgørende for at skabe rammer for, at der kan udvikles og etableres en fælles fagkultur, hvor lærere trives, erfaringsudveksler og kan levere god, engagerende undervisning, der lever op til intentionerne i curriculum.

Det siger sig selv, at det ikke går uden. I gymnasiet, hvor informatik blev indført som forsøgsfag uden efteruddannelse af lærerne og uden udvikling af eksemplariske materialer, resulterede det i en *meget* varieret praksis, der i mange tilfælde ikke var afstemt med intentionerne i curriculum (ITU, 2014).

En tilsvarende erfaring er der med det forsøgsvalgfag i teknologiforståelse, der blev indført i grundskolen i 2017 (Tuhkala et al., 2018).

Fravær af eksemplarisk materiale åbner endvidere mulighed for, at andet mere tilfældigt materiale bliver normsættende for en praksis, der kan være meget anderledes end den intenderede.

6.2 Uddannelse af undervisere

En empirisk undersøgelse i 2016 blandt danske undervisere i grundskolen viste en række udfordringer i forhold til at implementere en CT faglighed i en dansk uddannelseskontekst (Smith et al., 2016), hvor underviserne udtrykte en:

- generel usikkerhed på deres kompetencer til at undervise i og med digital teknologi
- usikkerhed i forhold til deres egen evne til at forstå kompleks problemløsning og

- usikkerhed i forhold til deres klasseledelse og autoritet, når undervisningen indeholdt elementer af projektorienteret undervisning, hvor underviseren ikke på forhånd vidste, hvor elevernes proces ville føre dem hen.

Der er grund til at antage, at undervisere på ungdomsuddannelser og på universiteter vil opleve samme udfordringer. Det er dog ikke forskningsmæssigt påvist.

Af hensyn til skalering er der behov for en storstilet indsats højt i fødekæden blandt andet gennem særlige ph.d.-programmer for (kommende) undervisere på professionshøjskoler; det slår to fluer med et smæk ved at kombinere skabelse og udbredelse af viden.

Fagets skabende og kreative natur gør, at lærerne ikke bare skal vide; de skal kunne. Og specielt skal de kunne vejlede deres elever/studerende i de strukturerede arbejdsformer, som er grundlaget for systematisk analyse, design, udvikling og konstruktion af digitale artefakter.

For undervisere i folkeskolen, i gymnasiet, på professionshøjskolerne og på universiteterne er der behov for målrettede, strukturerede og fleksible efteruddannelses tilbud. Da der samtidig er en enorm knaphed mht. personer med faglig dybde inden for området, vil det give mening med et nationalt samarbejde.

Et (distribueret) nationalt videnscenter i et tværsektorielt samarbejde synes at være den rigtige løsning.

6.3 Forskning og fagdidaktisk indsigt

Den øgede digitalisering i uddannelser sammenholdt med det store politiske fokus på digital teknologi i folkeskolen har accelereret processen, hvormed CT introduceres i uddannelser.

I grundskolen har kommuner efter bedste evne udviklet egne læseplaner for en teknologiforståelsesfaglighed. Projekter fra undervisningsministeriet (CRAFT), fra IT-brancheforeningen (Coding Class), fra kommunesamarbejder (fablab@school.dk), fra Engineer the Future og fra foreningsarbejde (Coding Pirates) giver alle nogle gode bud på, hvordan CT-kompetencer kan introduceres i en formel eller uformel skolekontekst.

Disse projekter giver en god base af empiriske erfaringer af større eller mindre omfang, men kan ikke i sig selv give svar på tre helt afgørende forhold ifm. indførelse af CT i uddannelsessystemet:

- Almengørelse
- Skalering
- Evaluering

Som nævnt i indledningen til afsnit 6 repræsenterer disse forhold en mægtig udfordring, som det vil kræve en systematisk og forskningsbaseret indsats over en længere årrække at finde solide og langtidsholdbare svar på.

De tre nævnte forhold er ikke udtømmende. Appendiks A.3 indeholder en mere fyldig diskussion af nogle af de væsentlige forskningsmæssige udfordringer ift. at implementere CT i uddannelsessystemet, herunder naturligvis de tre nævnte forhold.

Almengørelse af en dansk faglighed

CT er globalt set, og i særdeleshed i Danmark, en relativt ny faglighed uden for snævre universitetsuddannelser såsom datalogi, ingeniørvidenskab og informationsvidenskab. CT er derfor ikke i tilstrækkelig grad almengjort med henblik på at passe ind i et dansk uddannelsessystem.

Med almengørelse mener vi her en proces, i hvilken fagligheden er tilpasset et forståelsesunivers uden for en fagprofessionel diskurs. Det betyder bl.a., at der ikke systematisk er skabt viden om, hvordan progression gennem skolesystemet kunne opbygges. Der er naturligvis noget at hente i almenpædagogikken, men feltet har sin egen fagdidaktik, som skal udvikles.

Et eksempel på et aspekt af fagdidaktik kunne være fagets behandling af komplekse problemer. Et komplekst problem (wicked problem) er af en sådan karakter, at det oftest er vanskeligt eller ligefrem umuligt at løse på grund af ufuldstændige, modstridende og ændrede krav, der ofte er vanskelige at genkende [Rittle & Webber, 1973]. I en progressionsopbygning af uddannelse i kompleks problemløsning vil nogle måske forfalde til at introducere problemløsning i indskoling og i mellemklassetrinnet gennem anvendelse af simple problemer med stigende sværhedsgrad. Simple problemer kan godt være vanskelige at løse, men er ikke komplekse, fordi kompleksitet i denne sammenhæng går på problemets strukturelle opbygning. Altså går progressionen for kompleks problemløsning ikke igennem en indledende øvelse i simple problemer, men måske nærmere gennem didaktisering af værktøjer, metoder og teknikker til indlæring af processer, der inkluderer komplekse problemer. Her mangler Danmark (og verden) en forskningsbaseret fagdidaktisk viden, som kan føre til en egentlig almengørelse af faget uden for en fagprofessionel verden.

Der er et tilsvarende behov inden for de øvrige dele af fagligheden både ift. grundskolen (teknologiforståelse) og ungdomsuddannelserne (informatik).

Skalering

De eksisterende og særdeles spændende projekter foregår som oftest på udvalgte skoler med udvalgte lærere og med et særligt tilskud fra velmenende fonde, kommunekasser eller skoler.

Vi har i Danmark meget få (hvis nogen) forskningsstudier af CT-fagligheden på hverdagens præmisser, hvor lærere under den givne normering, i en obligatorisk faglighed og uden særlige omstændigheder skal bringe CT-fagligheden ind i en uddannelse. Her mangler vi viden om, hvordan fagligheden skalerer til hele uddannelsesverdenen og specifikt hvilke støttende indsatser, der vil virke, for at denne skalering indebærer også, at de rigtige digitale teknologier, som er tilpasset en progressionsforståelse (se ovenfor), er til stede og kan vedligeholdes i de respektive uddannelsesmiljøer. Her mangler vi også en tilstrækkelig viden om de digitale teknologiers didaktiske kvaliteter og om den indsats, der skal til for at vedligeholde uddannelsernes digitale teknologier.

Evaluering

Der ligger ikke i de hidtidige og relativt spæde forsøg med CT-faglighed i det danske uddannelsessystem en forskningsunderbygget forståelse af evalueringskriterier for elevernes præstationer. I den internationale forskning er der en række bestræbelser på at identificere bl.a. modeller for evaluering – se for eksempel overblikartiklen *The State of the Field in*

Computational Thinking Assessment, [Tissenbaum et al., 2018] – men generelt er der ingen konsensus på området, og specielt mangler sådanne modeller det myndiggørelsesaspekt, som er unikt i en dansk uddannelseskontekst.

Vi mangler i Danmark en forskningsindsats, som kan tilvejebringe relevante evalueringskriterier for hele fagligheden. Det arbejde skal selvsagt bygge på en dyb fagforståelse kombineret med en forståelse af det danske uddannelsessystem.

6.4 Udfordringer

Der er selvsagt rigeligt med udfordringer i at realisere visionen om at informatik/CT som en almen kompetence i uddannelser på alle niveauer efter den tostrengede strategi: informatik som en fundamental og selvstændig disciplin og informatik/CT som en tværgående disciplin integreret i alle andre (som fag og i fag). I den forbindelse er der meget at lære fra de erfaringer man har gjort sig i udlandet, specielt i USA og UK.

Hvad kan vi lære af erfaringerne fra USA og UK?

Overskriftens spørgsmål kan besvares meget kort: alt! Samtlige de 12 anbefalinger i figur 4.5 på side 23 er lige så relevante i Danmark, som de er i UK. Vores fordel er, at vi kan være på forkant med situationen ved at trække på briternes erfaringer.

Computing At School (CAS, 2018) er en frivillig organisation – et fællesskab, som har og har haft afgørende betydning for, at tingene et langt stykke hen ad vejen, trods alt er lykkedes. Under mottoet '*there is no "them" – only us*' har CAS formået at skabe et nationalt fællesskab omkring opgaven, hvor faglige og informatik-didaktiske eksperter oplærer 'master teachers' og stimulerer 'Networks of Excellence' med centre fordelt fintmasket rundt omkring i landet (det siges, at der intet sted er mere end 20 miles til det nærmeste center).

Amerikanerne har med konsortiet CSforALL et tilsvarende initiativ, der understøtter implementeringen i USA (CSforALL, 2018).

I Danmark har vi brug for et tilsvarende initiativ, der kan samle alle gode kræfter i sagens tjeneste. Teknologipagten bør være *en* af løftestængerne hertil (Teknologipagt, 2018). Svarende til andre steder i verden har universiteterne, som jo er vidensinstitutionerne på området, et særligt ansvar for at tage teten og indgå i samarbejde med alle relevante aktører ift. udvikling og udbredelse af den almene fag-faglige ekspertise og den hermed intimt forbundne fagdidaktik.

* * *

Indførelse af informatik/CT i uddannelsessystemet er en unik opgave ulig noget, vi tidligere har prøvet, og det kalder på massiv investering i forskningsbaseret kompetence- og kapacitetsopbygning ift. almengørelse af fagligheden, videreudvikling af informatikkens fagdidaktik, udvikling af curriculum og eksemplariske undervisningsmaterialer og sidst, men absolut ikke mindst, uddannelse og efteruddannelse af lærere og lærer-lærere.

I den proces må vi tænke ud af boksen og etablere utraditionelle, tværinstitutionelle samarbejder og uddannelsesprogrammer og højkvalitets digitale materialer, som kan sikre en effektiv og stærkt forankret faglighed (såvel fag-faglighed som undervisningsfaglighed) for lærer-lærere og de lærere – både folkeskole- og gymnasielærere – der i sidste ende skal løfte

opgaven. Måske kunne man endda genoverveje idéen om en fælles grundskole- og gymnasielæreruddannelse mellem professionshøjskoler og universiteter, som det blev foreslået på scienceområdet af VIA og AU i 2009 (SciencePlus, 2009).

Det er afgørende, at lærerne er klædt godt på og er trygge ved opgaven, samtidig med at de ikke er bange for at fejle, lære og videndele i stærke faglige netværk, der lokalt, regionalt og nationalt kan støtte, udvikle og udforske implementeringen.

Det er vores vurdering, at der er behov for at rammesætte de nødvendige initiativer i et nationalt videnscenter som forener alle gode kræfter om den fælles opgave.

7. Konklusion

Vi har præsenteret informatik og CT som nyt og fundamentalt vidensområde i uddannelser. Gennem en diskussion af centrale begreber og aspekter har vi tilstræbt at belyse, udfolde og fremhæve potentialet i at omfavne informatik og CT – både som selvstændigt fag og ved integrering i andre fag – ligesom der er peget på essentielle udfordringer i forbindelse med implementering af faget i uddannelsessystemet.

Mere præcist har vi i de foregående seks kapitler præsenteret følgende aspekter:

- informatik og CT som en fundamental kompetence på niveau med sprogfag og matematik og som en ny videnskabelig metode, der komplementerer teori og eksperiment
- baggrund for og karakteristik af CT og sammenhængen til relaterede begreber som 'mathematical thinking', STEM-uddannelse samt FabLab, makerspaces og engineering
- eksempler på udbredelse af CT i erhverv og videnskab og på de nye, effektive og grænseoverskridende muligheder, som computationelle metoder og teknikker giver
- status for CT i uddannelse såvel internationalt som nationalt, bl.a. med fokus på erfaringer fra USA og UK samt en kortlægning af situationen i Europa (anno 2016), og ikke mindst med fokus på forskningsaktiviteter og de mange åbne spørgsmål, der mangler gode svar på
- potentialet for CT, specielt med fokus på den vigtige tostrengede strategi med informatik som en fundamental og selvstændig disciplin og informatik/CT som en tværgående disciplin integreret i alle andre discipliner (*som fag og i fag*)
- muligheder og udfordringer for CT i det danske uddannelsessystem, hvor der forestår et kæmpe arbejde med at udvikle et progressivt curriculum og eksemplariske materialer, uddanne undervisere og afklare de mange forskningsspørgsmål, som udestår.

Fremstillingen er suppleret af fire appendikser. De to første uddyber begrebet Computational Thinking, som det er blevet promoveret af Wing og Papert (hhv. appendiks A.1 og A.2). Appendiks A.3 er et review af godt 70 udvalgte artikler om CT i uddannelse, og appendiks A.4 er et meget foreløbigt bud på en taksonomi for CT i videregående uddannelse i Danmark.

Vi sammenfatter præsentationen i termer af komponenterne i en SWOT-analyse, idet vi afslutningsvis helt overordnet fokuserer på styrker, svagheder, muligheder og trusler/udfordringer for CT i det danske uddannelsessystem (se figur 7.1).

* * *

Det sker ikke ofte, at der kommer et radikalt nyt fag på alle niveauer i uddannelsessystemet. Det er således en enestående mulighed, det er en spændende og mægtig samfundsopgave, og det er et stort ansvar for alle involverede. Vi må i fællesskab angribe opgaven med ansvarlighed, ydmyghed og kompetence – og først og fremmest med tanke på de elever og studerende, som skal nyde godt af bestræbelserne.

Styrker

- Vi er godt på vej med informatik i gymnasiet og teknologiforståelse i grundskolen.
- Fagidentiteten defineres af teknologiafhængige principper, tænke måder, udtryksformer, arbejdsformer og implikationer; dermed repræsenterer informatik og teknologiforståelse en stabil faglighed, der har værdi langt ud over tidens teknologi.
- Vi har et helhedsorienteret fokus med blik for både en konstruktiv-kreativ og kritisk-analytisk tilgang – i begge tilfælde med fokus på systematiske og eksplicite processer.
- Der er en underskov af initiativer både i og uden for skolen i form af fritidsaktiviteter, konkurrencer m.m.
- Der er kraftig fokus/opbakning fra erhvervslivet, politiske interesseorganisationer og strategiske politiske tiltag.
- Danmark et lille og homogent land med en stærk dannelses- og uddannelseskultur i hele uddannelsessystemet; dermed er der principielt fine rammer for skalering.

Svagheder

- Informatik/CT er kun obligatorisk i HHX og EUX, hverken i STX, HTX eller HF.
- Teknologiforståelse i folkeskolen er kun et forsøgsfag.
- Informatik/CT indgår ikke i læreruddannelsen på professionshøjskolerne.
- Informatik/CT indgår ikke i sidefag på universiteterne (gymnasielærere).

Muligheder

- Udlandet er på mange områder nogle år foran os, og dermed er der mange vigtige erfaringer at trække på, således at vi kan undgå at betale de samme lærepenge.
- Vi har gode erfaringer fra skalérbare projekter både i folkeskolen og gymnasiet.
- Der er politisk momentum for sagen.
- Vi har forskningscentre/-miljøer, hvor nødvendig forskning og udvikling kan forankres.
- Vi har med en kombination af offentlige og private midler mulighed for at prioritere en ambitiøs og målrettet indsats.
- Vi har mulighed for at etablere tværsektorielt samarbejde, f.eks. i form af et nationalt videnscenter.
- Med særlige ph.d.-programmer for (kommende) professionshøjskolelærere kan vi kombinere videnskabelse og -spredning.

Trusler/udfordringer

- Fødekæden af lærere, der kan undervise i informatik/CT/teknologiforståelse, eksisterer ikke.
- Selv på højeste niveau i fødekæden, hvor ekspertisen er til stede, er ressourcen yderst knap.
- Det er nødvendigt med et skarpt sprog for informatik/CT i en dansk/skandinavisk kontekst.
- Der bør etableres en stærk og sammenhængende progression for uddannelse i informatik/CT i hele uddannelsessystemet.

Figur 7.1: Styrker, svagheder, muligheder og trusler/udfordringer for informatik og CT i det danske uddannelsessystem

8. Referencer

- Bell, T. (2018). [CS Unplugged: Computer Science Without a Computer](https://csunplugged.org/en/). University of Canterbury, New Zealand. Hentet 4. oktober 2018 fra <https://csunplugged.org/en/>.
- Besenbacher, F. & Caspersen, M.E. (2017). [Faren er dumme mennesker, ikke kloge robotter](#), Kronik i Politiken, 4. december 2017.
- Blikstein, P. (2013). [Digital Fabrication and 'Making' in Education: The Democratization of Invention](#). In Walter-Herrmann J. & Büching, C. (Eds.), *FabLab Of Machines, Makers and Inventors*, pp. 203-222.
- Blikstein, P. (2018). [Transformative Learning Technologies Lab](#), Stanford Graduate School of Education.
- Boblestudier (2018). [Center for Information og Boblestudier](#), Københavns Universitet. Hentet 30. november 2018 fra <https://bubblestudies.ku.dk>.
- CALDISS (2018). [Computational Analytics Laboratory for Digital Social Science](#), Aalborg Universitet. Hentet 30. november 2018 fra <https://www.caldiss.aau.dk>.
- CAS (2015). [CAS computational thinking – A guide for teachers](#), Computing At School. Hentet 25. september 2018 fra <https://community.computingschool.org.uk/resources/2324/single>.
- CAS (2018). [Computing At School](#). Hentet 30. september 2018 fra <https://www.computingschool.org.uk>.
- Caspersen, M.E. & Nowack, P. (2013). [Computational Thinking and Practice – A Generic Approach to Computing in Danish High Schools](#), *Proceedings of the 15th Australasian Computing Education Conference, ACE 2013*, Adelaide, South Australia, Australia, pp. 137-143.
- Caspersen, M.E. (2017a). Computational Thinking, kapitel 4.15 i *Gymnasiepædagogik – En grundbog*, Hans Reitzels Forlag, 9 sider.
- Caspersen, M.E. (2017b). Teaching Programming, chapter 9 in *Computer Science Education: Perspectives on Teaching and Learning in School* (Sentance et al. 2018).
- Caspersen, M.E., Gal-Ezer, J., McGettrick, A. & Nardelli, E. (2018). [Informatics for All: The Strategy](#), The Informatics for All Committee by ACM Europe and Informatics Europe. En europæisk strategi præsenteret i Bruxelles 15. marts 2018.
- CCTD (2018a). [Computational Thinking in High School subjects](#), Center for Computational Thinking & Design, Aarhus University.
- CCTD (2018b). [CT i Gymnasiefag – Afsluttende rapport til Region Midtjylland /Regional Udvikling](#), Center for Computational Thinking & Design, Aarhus Universitet, april 2018.
- CECE (2017). [Informatics Education in Europe: Are We All In The Same Boat?](#), Report by the Committee on European Computing Education, Informatics Europe and ACM Europe. Hentet 12. januar 2018 fra <http://www.informatics-europe.org/component/phocadownload/category/10-reports.html?download=60:cece-report>.
- CECE's Map (2017). [CECE's Map of Informatics in European Schools](#), ACM Europe and Informatics Europe.
- Chicago CS (2014). [CPS Announces First Schools to Implement District's Comprehensive K-12 Curriculum](#), Chicago Public Schools.

- CollegeBoard (2018). [AP Computer Science Expansion](https://reports.collegeboard.org/ap-program-results/ap-computer-science-expansion). Hentet 12. oktober 2018 fra <https://reports.collegeboard.org/ap-program-results/ap-computer-science-expansion>.
- Computer History (2018). [Ada Lovelace](http://www.computerhistory.org/babbage/adalovelace/), *The Babbage Engine*, Computer History Museum. Hentet 14. januar 2018 fra <http://www.computerhistory.org/babbage/adalovelace/>.
- CSforALL (2018). [Computer Science for All Consortium](https://www.csforall.org). Hentet 30. september 2018 fra <https://www.csforall.org>.
- CS4FN (2018). [Computer Science for Fun](http://www.cs4fn.org). Hentet 29. september 2018 fra <http://www.cs4fn.org>.
- CT-STEM (2018). [Computational Thinking in Science and Math](https://ct-stem.northwestern.edu). Hentet 18. oktober 2018 fra <https://ct-stem.northwestern.edu>.
- Denning, P. (2009). The Profession of IT: Beyond Computational Thinking, *Communications of the ACM*, Vol. 52 (6), pp. 28-30.
- diSessa, A.A. (2000). *Changing Minds: Computers, Learning, and Literacy*, MIT Press, 293 sider.
- Disruptionråd (2017). [Disruptionrådet – Partnerskab for Danmarks Fremtid](https://bm.dk/arbejdsomraader/aktuelle-fokusomraader/disruptionraadet/), Beskæftigelsesministeriet, 2017. Hentet 30. november 2018 fra <https://bm.dk/arbejdsomraader/aktuelle-fokusomraader/disruptionraadet/>.
- eScience (2018). [SDU eScience Center](https://escience.sdu.dk), SDU. Hentet 30. november 2018 fra <https://escience.sdu.dk>.
- European Commission (2016). [Developing Computational Thinking in Compulsory Education – Implications for policy and practice](https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/developing-computational-thinking-compulsory-education-implications-policy-and-practice), European Commission, Joint Research Centre. Hentet 12. januar 2018 fra <https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/developing-computational-thinking-compulsory-education-implications-policy-and-practice>.
- Gardner, M. (1998). [The New New Math](http://www.nybooks.com/articles/1998/09/24/the-new-new-math/), *The New York Review of Books*, Vol. 45 (14), September 24, 1998. Hentet 28. januar 2018 fra <http://www.nybooks.com/articles/1998/09/24/the-new-new-math/>.
- Guzdial, M. (2018). [Computational Mapping: An important set of skills in Computational Thinking we can define and test](https://computinged.wordpress.com/2018/10/15/computational-mapping-a-part-of-wings-computational-thinking-i-can-get-behind-doescomputationalthinkingexist/), *Computing Education Research Blog*. Hentet 18. oktober 2018 fra <https://computinged.wordpress.com/2018/10/15/computational-mapping-a-part-of-wings-computational-thinking-i-can-get-behind-doescomputationalthinkingexist/>.
- Hansbøl, M. & Ejsing-Duun, S. (2017). [Rapport: Coding Class – Dokumentation og evaluering](https://www.ssi.dk/~/media/SSI/2017/10/Rapport_Coding_Class_-_Dokumentation_og_evaluering.pdf).
- Harel, G. & Snowden, (2005). [Advanced Mathematical-Thinking at Any Age: Its Nature and Its Development](https://www.elsevier.com/locate/S0095956205000000), *Mathematical Thinking and Learning*, Vol. 7 (1), pp. 27-50.
- Heintz, F. & Manilla, L. (2018). Computational Thinking for All – An Experience Report on Scaling up Teaching Computational hinking to All Students in a Major City in Sweden, *Proceedings of the 49th Technical Symposium on Computer Science Education, SIGCSE 2018*, Baltimore, USA.
- ITU (2014). [Evaluering af informationsteknologi C/B som forsøgsfag på STX, HF, HTX og HHX](https://tinyurl.com/y6wof7qd), IT-Universitetet i København, 2014. Hentet 30. november 2018 fra <https://tinyurl.com/y6wof7qd>.
- Iversen, O.S. & Klokmose, C.N. (2018). *Fra Computational Thinking til Computational Empowerment: En Nordisk tilgang til digital (ud)dannelse*, Aarhus Universitet, internt notat.

- Iversen, O.S., Dindler, C. & Smith, R.C. (2018a). *From Computational Thinking to Computational Empowerment: A 21st Century Participatory Design Agenda*, under review.
- Iversen, O.S., Dindler, C. & Smith, R.C. (2018b). *Digital teknologi og design i undervisningen*, Dafolo (forventes publiceret primo 2019).
- Kafai, Y.B. & Burke, W.Q. (2013). "The Social Turn in K-12 Programming: Moving from Computational Thinking to Computational Participation". In *Proceeding of the 44th ACM Technical Symposium on Computer science education (SIGCSE '13)*, pp.603-608.
- Kline, M. (1973). [Why Johnny Can't Add: The Failure of the New Math](#), St. Martin's Press.
- K12CS (2016). [K-12 Computer Science Framework](#). Hentet 12. oktober 2018 fra <https://k12cs.org/wp-content/uploads/2016/09/K-12-Computer-Science-Framework.pdf>.
- Ko, A. (2018). [Why people shouldn't learn to code](#), *Bits and Behavior — Musings from the Code & Cognition Lab at the University of Washington Information School*. January 5.
- Martin, L. (2015). [The Promise of the Maker Movement for Education](#), *Journal of Pre-College Engineering Education Research*, Vol. 5 (1), Article 4.
- MIMI (2018). [Analytical Engine](#), Museum of Imaginary Musical Instruments. Hentet 11. januar 2018 fra <http://imaginaryinstruments.org/lovelace-analytical-engine/>.
- Musaeus, L.H. & Musaeus, P. (2019). Computational Thinking in the Danish High School, to appear in *Proceedings of the 50th Technical Symposium on Computer Science Education, SIGCSE 2019*, Minneapolis, USA.
- Nardelli, E., Forlizzi, L., Lodi, M., Lonati, V., Mirolo, C., Monga, M., Montresor, A. & Morpurgo, A. (2017). [Proposal for a national Informatics curriculum in the Italian school](#), National Interuniversity Consortium for Informatics.
- Naur, P. (1967). [Datalogi – læren om data](#), Den anden af fem Rosenkjærforelæsninger i Danmarks Radio udgivet under titlen *Datamaskinerne og samfundet* af Munksgaard.
- Nowack, P. & Caspersen, M.E. (2014). [Model-Based Thinking and Practice — A Top-Down Approach to Computational Thinking](#), *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, Koli Calling 2014, Koli, Finland.
- NYC CS (2015). [De Blasio to Announce 10-Year Deadline to Offer Computer Science to All Students](#), New York Times, 15. september 2015.
- Nygaard, K. (1986). [Program Development as a Social Activity](#), in *Proceedings of the IFIP 10th World Computer Congress, INFORMATION PROCESSING 86*, Dublin Elsevier Science Publishers, pp. 189-198.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*, Basic Books, New York.
- Papert, S. (1996). [An exploration in the space of mathematics educations](#), *International Journal of Computers for Mathematical Learning*, 1 (1), pp. 138-142.
- Partovi, H. (2018). [Anybody Can Learn: Another year of record growth for AP Computer Science](#), code.org.
- Regeringen (2018). [Strategi for Danmarks Digitale Vækst](#), Regeringen.
- Repenning, A. (2012). Programming Goes Back to School—Broadening participation by integrating game design into middle school curricula, *Communications of the ACM*, 55 (5), pp. 38-40.

- Repenning, A., Basawapatna, A. & Escherle, N. (2016). [Computational Thinking Tools](#), Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 218-222.
- 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.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. & Kafai, Y. (2007). [Scratch: Programming for All](#), *Communications of the ACM*, 52 (11), pp. 60-67.
- Resnick, M. (2016). [A Message to All Makers](#), Afterword in [Start Making!](#), by D. Martin & A. Panjwani, edited by N. Rusk, pp. 181-183, Maker Media.
- Riisgaard Hansen, T. (2016). [The Role of AI in Healthcare – an in-depth guide](#). Hentet 21. oktober 2018 fra <https://www.linkedin.com/pulse/role-ai-healthcare-in-depth-guide-thomas-riisgaard-hansen/>.
- Rittle, H. & Webber, M. (1973). Dilemmas in a General Theory of Planning, *Policy Sciences*, Vol. 4, Elsevier Scientific Publishing Company, Inc., Amsterdam, pp. 155-169. Reprinted in Cross, N. (ed.), *Developments in Design Methodology*, John Wiley & Sons, 1984, pp. 135-144.
- Rosenbloom, P.S. (2012). [On Computing: The Fourth Scientific Domain](#), MIT Press, 336 sider.
- Royal Society (2017). [After the reboot: Computing education in UK schools](#). Hentet 12. januar 2018 fra <https://royalsociety.org/topics-policy/projects/computing-education/>.
- Rushkoff, D. (2010). *Program or Be Programmed: Ten Commands for a Digital Age*, O/R Books, New York.
- 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*, Vol. 18 (2), pp. 351-380.
- Sentance, S., Barendsen, E. & Schulte, C. Eds. (2018). Computer Science Education: Perspectives on Teaching and Learning in School, Bloomsbury.
- Schelhowe, H. (2013). Digital Realities, Physical Action and Deep Learning – FabLabs as Educational Environments. In Walter-Herrmann J. & Büching, C. (Eds.), *FabLab Of Machines, Makers and Inventors*, pp. 93-104.
- Schwartzbach, M. I. (2011). [Ligningsløseren](#). Tilgået 17. oktober 2018 fra <http://iftek.dk/ligningsloser>.
- ScienceAtHome (2018). [ScienceAtHome](#), Institut for Fysik, Aarhus Universitet. Hentet 30. november 2018 fra <https://www.scienceathome.org>.
- SciencePlus (2009). [Nye scienceuddannelser: Naturvidenskab – undervisning, kommunikation og organisation](#), Det Naturvidenskabelige Fakultet, Aarhus Universitet.
- SF CS (2015). [Board Approves Plans to Expand Computer Science to All Grades](#), San Francisco Unified School District, 10. juni 2015.
- Smith, R.C, Iversen, O.S & Hjorth, M. (2015). Design Thinking for Digital Fabrication in Education, *International Journal of Child-Computer Interaction*, 5 (C), pp. 20-28.

- Smith, R.C., Iversen, O.S & Veerasawmy, R. (2016). [Impediments for Digital Fabrication in Education: a study of teachers' role in digital fabrication](#), *International Journal of Digital Literacy and Digital Competence*, Bind. 7, Nr. 1, pp. 33-49.
- Snow, C.P. (1959). *The Two Cultures and The Scientific Revolution*, Cambridge University Press.
- SODAS (2018). [Copenhagen Centre for Social Data Science \(SODAS\)](#), Københavns Universitet. Hentet 30. november 2018 fra <https://sodas.ku.dk/dansk/>.
- TANTLAB (2018). [The Techno-Anthropological Laboratory](#), Aalborg Universitet. Hentet 30. november 2018 fra <https://www.tantlab.aau.dk/>.
- Tedre, M. & Denning, P. (2017). [Shifting Identities in Computing: From a Useful Tool to a New Method and Theory of Science](#), in *Informatics in the Future*, Springer-Verlag, DOI 10.1007/978-3-319-55735-9_1, 16 sider.
- Tedre, M. & Denning, P. (2019). *Computational Thinking*, MIT Press (to appear).
- Teknologiforståelse (2017). [Teknologiforståelse valgfag – Fælles Mål og Læseplan](#), EMU Danmarks læingsportal, Undervisningsministeriet. Hentet 12. januar 2018 fra <https://www.emu.dk/modul/teknologiforstaaelse-valgfag-forsog---faelles-mal-og-laeseplan>.
- Teknologipagt (2018). [Teknologipagt skal løfte danskernes digitale og teknologiske kompetencer](#). Erhvervsministeriet. Hentet 31. januar 2018 fra <https://em.dk/nyheder/2018/01-25-teknologipagt>.
- Tissenbaum, M. et al. (2018). [The State of the Field of Computational Thinking Assessment](#), *Proceedings of the 13th International Conference of Learning Sciences (ICLS 2018)*, pp. 1304-1311.
- Tuhkala, A, Wagner, M.-L., Nielsen, N., Iversen, O.S. & Kärkkäinen, T. (2018). [Technology Comprehension: Scaling Making into a National Discipline](#), *Proceedings of the Conference on Creativity and Making in Education (FabLearn Europe '18)*, pp. 72-80.
- UFM (2018a). [Universitetsuddannelser til fremtiden](#), Uddannelses- og Forskningsministeriet.
- UFM (2018b). [Call for Action: Teknologisk upgrade på de videregående uddannelser](#), Uddannelses- og Forskningsministeriet.
- UK CS (2013). [National curriculum in England: computing programmes of study](#), Department of Education, 11. september 2013.
- UVM (2018). [Kommissorium for den rådgivende ekspertskrivegruppe for forsøgsprogram for styrkelse af teknologiforståelse i folkeskolens obligatoriske undervisning](#), Undervisningsministeriet. Hentet 26. januar 2018 fra <https://www.uvm.dk/aktuelt/nyheder/uvm/2018/jan/180126-undervisningsministeren-vil-goere-teknologiforstaaelse-obligatorisk-i-folkeskolen>.
- Vækstpanel (2017). [Danmark som digital frontløber](#), Digitalt Vækstpanel.
- Vækstråd (2016). [Rapport om kvalificeret arbejdskraft](#), Danmarks Vækstråd.
- 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), pp. 127-147. Published online 8 October 2015.
- White House (2016). [Computer Science For All](#), The White House. Hentet 12. oktober 2018 fra <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>.
- Wilensky, U. (1997). [NetLogo Wave Machine model](#). Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Hentet 28. januar 2018 fra <http://ccl.northwestern.edu/netlogo/models/WaveMachine>.

- Wilensky, U. (1999). [NetLogo](#), Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Hentet 28. januar 2018 fra <http://ccl.northwestern.edu/netlogo/>.
- Wilensky, U. (2006a). [NetLogo Rope 3D model](#), Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Hentet 28. januar 2018 fra <http://ccl.northwestern.edu/netlogo/models/Rope3D>.
- Wilensky, U. (2006b). 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., Brady, C.E & Horn, M.S. (2014). [Fostering Computational Literacy in Science Classrooms](#), *Communications of the ACM*, Vol. 57 (8), pp. 24-28.
- Wilensky, U. & Rand, W. (2015). *An introduction to agent-based modeling: Modeling natural, social, and engineered complex systems with NetLogo*. Cambridge, MA: MIT Press.
- 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.
- Wing, J.M. (2006). [Computational Thinking](#), *Communications of the ACM*, Vol. 49 (3), pp. 33-35.
- Wing, J.M. (2008). [Computational Thinking and Thinking about Computing](#), *Philosophical Transactions of the Royal Society A*, 366, pp. 3717-3725.
- Wing, J.M. (2011). [Computational Thinking—What and Why?](#), *The Link*, News from the School of Computer Science at Carnegie Mellon University.
- Wing, J.M. (2014). [Computational Thinking Benefits Society](#), *Social Issues in Computing*, 40th Anniversary Blog.
- Wing, J.M. (2016a). "[Computational Thinking, Ten Years Later](#)", *Communications of the ACM*, Blog, March 23.
- Wing, J.M. (2016b). [Computational Thinking](#), Vienna Gödel Lecture, TU Wien, Austria. Hentet 12. januar 2018 fra <https://www.youtube.com/watch?v=YVEUOHw3Qb8>.
- Introduction to Computational Thinking in Research and Education, 12:40 - 16:15
- One discipline, many computational methods (bioscience), 26:05 - 34:55
- One computational method, many disciplines (machine learning), 35:15 - 44:00
- Computational Thinking in the Sciences and Beyond, 44:00 - 55:10
- Wolfram, S. (2016). [How to Teach Computational Thinking](#), Stephen Wolfram Blog. Hentet 28. oktober 2018 fra <https://blog.stephenwolfram.com/2016/09/how-to-teach-computational-thinking/>
- World Economic Forum (2018). [The Future of Jobs Report 2018](#). Hentet 30. november 2018 fra <https://www.weforum.org/reports/the-future-of-jobs-report-2018>.
- World Economic Forum (2016). [The Fourth Industrial Revolution](#). Hentet 28. januar 2018 fra <http://reports.weforum.org/future-of-jobs-2016/>.

9. Appendikser

A.1 Computational Thinking – et vidt begreb	49
A.1.1 Mathematical thinking – et tankeeksperiment	49
A.1.2 Implementering af informatik og CT	58
A.1.3 Referencer for appendiks A.1	58
A.2 Constructionism og restructurations	51
A.2.1 Constructionism	51
A.2.2 Computational Literacy	52
A.2.3 Restructurations	52
A.2.4 Referencer for appendiks A.2	53
A.3 Review af udvalgt forskningslitteratur om CT i uddannelse	54
A.3.1 Programmering og kodning som CT	54
A.3.2 Robotics som CT	54
A.3.3 Computermodellering som CT	55
A.3.4 Håndarbejde og STEAM som CT	55
A.3.5 Computerspildesign som CT	56
A.3.6 Evaluering af CT og læring	56
A.3.6.1 Viden om CT eller programmering	56
A.3.6.2 Emotionel holdning og attitude til CT	57
A.3.6.3 Viden om faget	57
A.3.6.4 Performance tasks	58
A.3.7 Tid til at stille store spørgsmål	58
A.3.7.1 Læringsteoretiske CT-principper	59
A.3.7.2 Evaluering	59
A.3.7.3 Underviseres CT-kundskaber	60
A.3.7.4 Nye værktøjer	60
A.3.7.5 Effekter af CT i undervisningen	60
A.3.7.6 Kønsforskelle og uddannelsesfremmede elever	61
A.3.8 Referencer for appendiks A.3	62
A.4 Tentativ dansk taksonomi for CT i uddannelser	66
A.4.1 Forudsætningskrav til studerende (og tilbud så de kan opfyldes/ efterleves)	66
A.4.2 CT-kompetencebeskrivelse for humanistiske it-uddannelser	66
A.4.3 Principper for organisatorisk integrering af CT-faglighed	67
A.4.4 Tentativ taksonomi for CT i humanistiske it-uddannelser	67
A.4.4.1 Problem framing	67
A.4.4.2 Data og informationsprocesser	67
A.4.4.3 Modellering og simulering	68
A.4.4.4 Computational undersøgelse og problemløsning	68
A.4.4.5 Systemtænkning	68

A.1 Computational Thinking – et vidt begreb

Som nævnt i afsnit 2.1 (i næstsidste afsnit på side 11) har Wings definition og karakteristik af CT aldrig været ment som en anbefaling af nøglebegreber i et curriculum for K-12. Spørgsmålet er derfor, om Wings definition af CT, der havde et helt andet sigte, ikke bare er blevet brugt, men er blevet misbrugt i forhold til K-12.

Nøglebegreberne, der indgår i toneangivende K-12-definitioner af CT (se afsnit 2.1 side 12 og (CAS, 2015)), og som indgår som bærende elementer i centrale publikationer til elever, lærere og ledere, er abstrakte og generelle – de er ikke konkrete og specifikke for informatik. Det betyder, at der er rigt rum til fortolkning – og misfortolkning.

For et nyt skolefag kan det være en styrke, at der i karakteristikken af faget er rummelighed til at afprøve forskellige fortolkninger og konkretiseringer af fagets overordnede begreber. Men når lærere, der skal undervise i faget, ikke har et fagligt overskud, bliver der et endnu stærkere behov end normalt for at præcisere, hvad der skal undervises i og hvordan. Og hvis de, der skal bidrage med denne præcisering, de, der skal udvikle undervisningsmaterialer til eleverne, de, der skal vejlede og (efter-)uddanne lærerne, og de, der skal udvikle undervisningsmaterialer til lærernes (efter-)uddannelse, ikke har et fagligt overskud og evner at genfortolke fagligheden ift. almen uddannelse, risikerer vi at ramme langt forbi målet.

Selv med højt kvalificerede curriculumdesignere, lærer-lærere og lærere er det usandsynligt, at vi vil ramme bare tilnærmelsesvis tæt på skivens centrum i første forsøg.

Til sammenligning kan man tænke på 'New Math', der efter sputnik-forskrækkelsen i 1957 fejede ind over uddannelsessystemet i hele den vestlige verden. Tiltaget og de pædagogiske principper blev hurtigt miskrediteret, specielt efter nobelprisvinder Richard Feynmans og Morris Klines kritik (Gardner, 1998; Kline 1973), men de lå til grund for matematikundervisningen i den danske folkeskole helt op i 1980'erne (Niss, 2018).

Risikoen for at ramme (langt) ved siden af skivens centrum forstærkes af, at der er mange kommercielle udbydere af undervisningsmateriale og -værktøjer, der konkurrerer hårdt for at være "først med det sidste", og som først og fremmest har den interesse at skabe et overskud på aktiviteterne.

Kort sagt risikerer vi for hurtigt at få slået de forkerte søm i, og det er ingen tjent med. Vi skal finde en adræt implementeringsstrategi, der på samme tid har fremdrift og er "sikker".

Disse betragtninger skal ikke forstås som forbehold over for, endsige kritik af, de mange kompetente personer og organisationer, som globalt arbejder hårdt og dedikeret for at indføre informatik og CT i almen uddannelse. Formålet er udelukkende at påpege opgavens natur og omfang og det faktum, at der naturligvis ikke findes nemme løsninger på den vigtige og mægtige opgave, vi står overfor.

A.1.1 Mathematical Thinking – et tankeeksperiment

Lad os et øjeblik forestille os en situation, hvor matematik kun eksisterer som højt specialiseret forskningsområde og uddannelse på universiteter (med forskningsområder som algebraisk topologi, differentialgeometri, funktionalanalyse og kategoriteori), og hvor befolkningen i

almindelighed blot har basale aritmetiske kompetencer svarende til "købmandsregning" (primært addition og subtraktion).

Og lad os i den situation forestille os, at nogle får den vision, at 'Mathematical Thinking' er nyttig for andre forskningsområder og bør indføres generelt i uddannelsessystemet.

Lad os endvidere antage, at visionen om 'Mathematical Thinking' er udtrykt i abstrakte og generelle, men dog matematikspecifikke termer som for eksempel *abstraktion, algebra, ligninger, funktioner, logik, beviser, matematisk analyse (grænseværdier samt differential- og integralregning) og sandsynlighedsregning*.

Fra den vision fødes, og til der er udviklet et meningsfuldt curriculum, gode undervisningsmaterialer med en meningsfuld progression op gennem uddannelsessystemet, effektive evalueringsformer, dygtige lærere, dygtige lærer-lærere og en matematisk fagdidaktik, er der mildt sagt et godt stykke vej!

Der vil i en sådan situation åbenlyst være en endog meget lang vej til, at matematik er almengjort og meningsfuldt implementeret i skolen.

A.1.2 Implementering af informatik og CT

Pointen med denne analogi er naturligvis ikke at afskrække, tværtimod. Pointen er at tydeliggøre den karakter og det omfang, som opgaven har.

Groft sagt er det den udfordring, vi som internationalt samfund står overfor med hensyn til at indføre informatik og CT i uddannelsessystemet, og analogien kan forhåbentlig hjælpe til, at vi kan tilgå opgaven informeret, kvalificeret og på den mest effektive måde med et langsigtet perspektiv, samtidig med at vi "kommer i gang" og gør det, der er politisk og praktisk muligt her og nu.

Der skal være tydelig fremdrift, hvor delmål indfries, men samtidig rum til justering af kursen. Præcis som i moderne softwareudvikling.

Der er ingen nemme løsninger, men der er behov for iterativt og inkrementelt at udvikle, eksperimentere og evaluere på såvel indhold som form for at kunne identificere og fastlægge en passende måde at implementere informatik på i skolen – både ved specialisering som fag og integrering i andre fag.

A.1.3 Referencer for appendiks A.1

- CAS (2015). *CAS computational thinking – A guide for teachers*, Computing At School. Hentet 25. september 2018 fra <https://community.computingatschool.org.uk/resources/2324/single>.
- Gardner, M. (1998). *The New New Math*, *The New York Review of Books*, September 24, 1998 Issue. Hentet 28. januar 2018 fra <http://www.nybooks.com/articles/1998/09/24/the-new-new-math/>.
- Kline, M. (1973). *Why Johnny Can't Add: The Failure of the New Math*, *St. Martin's Press*.
- Niss, M. (2018). *Matematik (undervisning)*, *Den Store Danske*, Gyldendal. Hentet 14. januar 2018 fra <http://denstoredanske.dk/index.php?sideId=122851>.

A.2 Constructionism og restructurations

Massachusetts Institute of Technology var fra 1960'erne og frem til de tidlige 90'ere det centrale sted for udvikling af tidlige idéer omkring Computational Thinking i undervisning og programmering for børn. Der var især to forskere, som satte deres præg på udviklingen af disse idéer: Seymour Papert og Andrea diSessa. En central idé for begge arbejder med børn og computere var, at veldesignet software kan understøtte lærings- og tankeprocesser hos børn og unge.

A.2.1 Constructionism

Seymour Papert opfandt den design- og læringsteori som hedder *Constructionism*. Papert arbejdede sammen med Piaget efter sin ph.d., og constructionism bygger da også på Piagets læringsteori, kaldet konstruktivism. Men til forskel fra Piaget, som primært var interesseret i læringsprocesser, var Papert interesseret i *læringsdesign*. Constructionism ser *konstruktion af personligt meningsfulde produkter, som kan vises og deles med andre*, som en vigtig type af læringsaktiviteter. Et centralt tema i constructionism er derfor, hvordan vi som læringsforskere, -udviklere og -designere bedst udvikler værktøjer, som børn kan konstruere disse produkter med.

Et eksempel på et sådant design, og måske det arbejde som Papert især blev kendt for, var hans design af en skildpadder til at undervise børn i trigonometri. Skildpadden var en udvidelse af programmeringssproget Logo, og børn kunne programmere den til at tegne former på skærmen. Trigonometri bliver som regel konceptualiseret i et koordinatsystem, hvor vi ser vinkler og former *udefra*. Papert mente, at dette *udefra*-perspektiv gør det svært for børn at forstå trigonometri, fordi det tvinger dem til at tænke om ting, som de kender (former osv.), men fra et perspektiv som er fremmed for dem. Som begrebmæssig løsning opfandt han *body syntonicity*.

Body syntonicity sætter elevernes eget perspektiv i centrum og lader dem dermed tænke om trigonometri *indefra*. Skildpadden og Paperts udvidelse af Logo-sproget giver elever mulighed for at tegne figurer eller grafer ved at beskrive den måde, skildpadden skal bevæge sig på (se Figur 1).



Figur 1: Eksempler på Logo

Skildpadden og den måde, som Logo-sproget understøtter læringsprocessen, hvormed børn lærer at forstå, hvordan former konstrueres, var også grundlaget for et andet af constructionismens vigtigste begreber – "*objects-to-think-with*". Dette begreb giver en god forståelse for, at formålet med designede produkter er at være en ekstern repræsentation af en problemstilling, som hjælper én med at tænke det specifikke problem igennem på en nemmere måde. Dette begreb er udgangspunktet for meget af det designarbejde, der arbejdes på i dag.

Et af de centrale spørgsmål i design af CT-curriculum og CT-værktøjer er derfor: hvad skal disse værktøjer kunne? Hvordan kan de understøtte læringsprocessen? Og hvad er – fra elevernes perspektiv – idéen med at bruge disse værktøjer til at løse og adressere problemer?

A.2.2 Computational Literacy

Andrea diSessa er en anden hovedperson i den historiske udvikling af Computational Thinking. Ligesom Papert var han også på MIT i 70'erne og 80'erne, men er nu professor på UC Berkeley. Et af hans hovedbegreber var *computational literacy* (diSessa, 2001). Dette begreb dækker over menneskers evne til at kunne løse problemer, som de ellers ikke ville kunne løse, i "samarbejde" med teknologi. diSessas eget arbejde fokuserede på grundskoleelevers brug af hans programmeringssprog *Boxer* (diSessa & Abelson, 1986) i fysikundervisning. Lige som Paperts skildpaddede hjalp elever med at tænke trigonometri igennem ved at lade dem tage skildpaddens perspektiv, så var Boxer designet, så elever kunne programmere f.eks. faldende objekter og beskrive, hvordan tyngdekraften ændrer disses hastighed fra objektets perspektiv.

Body-syntonicity og computational literacy var begge to eksempler på den samme grundlæggende idé: at vi kan designe computationelle værktøjer, tænkeværktøjer, som gør os i stand til at tænke forskellige problemstillinger bedre igennem.

A.2.3 Restructurations

Papert og hans tidligere ph.d.-studerende Uri Wilensky, som nu er professor på Northwestern University, har givet deres bud på den seneste iteration af denne idé. De kalder det *restructurations* (Wilensky & Papert, 2010) og giver *arabertal* som et eksempel. I tiden med romertal, altså før indførelsen af arabertal, krævede det årevis af træning at dividere et tal med et andet. Dette gjorde, at denne simple, men helt grundlæggende regnefærdighed var forbeholdt en meget lille og ofte elitær gruppe af mennesker. Resten af befolkningen var udelukket fra at kunne deltage i alle aktiviteter, som kræver division. Med indførelsen af arabertal, som repræsenterer tal på en helt anden måde, der gør det nemmere at dividere, blev division demokratiseret, og noget som alle kunne deltage i med en lille smule træning.

Det lille Piet Hein gruk i afsnit 1.3 illustrerer samme pointe.

Wilensky og Papert opstiller fem egenskaber ("properties") for *restructurations*:

1. Power properties: de kan mindst repræsentere de samme problemstillinger som tidligere repræsentationer, og gerne flere.
2. Cognitive properties: de kan gøre det nemmere at tænke en problemstilling igennem eller at løse et problem.
3. Affective properties: de kan gøre det sjovere eller rarere at løse problemer.
4. Social properties: de kan, af forskellige grunde, passe bedre sammen med de værktøjer, institutioner osv., som et problem skal løses med eller i.
5. Diversity properties: de kan tiltale forskellige menneskers måder at tænke på og dermed gøre det mere interessant at løse problemer med denne repræsentation for mennesker, som ikke ville finde det interessant med de gamle repræsentationer.

Denne idé om, at vi kan designe repræsentationer af data, problemstillinger, spørgsmål, er stadig et centralt og vigtigt aspekt i forbindelse med udvikling og forskning i Computational Thinking.

Det centrale spørgsmål er: hvordan designer vi CT-værktøjer og -arbejdsprocesser på en måde, som indeholder dele af de fem egenskaber beskrevet ovenfor, og som dermed kan være med til at gøre det nemmere og sjovere for en større gruppe af mennesker at løse vigtige problemer i samfundet?

A.2.4 Referencer for appendiks A.2

diSessa, A.A. (2001). *Changing minds: Computers, learning, and literacy*. Mit Press.

diSessa, A.A., & Abelson, H. (1986). Boxer: A reconstructible computational medium. *Communications of the ACM*, 29(9), 859–868.

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

A.3 Review af udvalgt forskningslitteratur om CT i uddannelse

Den diffuse definition af CT til trods sker der meget praktisk arbejde med forsøg på at designe CT-aktiviteter i grund- og gymnasieskoler. For at give læseren en idé om, hvad disse går ud på, gennemgår vi her kort nogle af de forskellige typer af CT-undervisningsværktøjer. Vi afslutter afsnittet med at skitsere nogle af de udfordringer og underbelyste emner, som forskningen peger på.

A.3.1 Programmering og kodning som CT

Programmering har fyldt meget i det praktiske arbejde med at bringe Computational Thinking til skoler siden begrebets oprindelse i Seymour Paperts arbejde med at lære unge at programmere i Logo (Papert, 1980; 1994).

Programmering er dog svært at lære for begyndere, og det er især et problem i et klasseværelse, hvor en underviser skal hjælpe mere end 20 børn ad gangen. Et centralt fokus har derfor været, hvordan man kan gøre det nemmere at komme i gang, og her har 'Visuelle' eller 'Blocks-Based Programming' (BBP) spillet en stor rolle. Scratch (Resnick et al., 2009) og dets mere sofistikerede storebror Snap! (Harvey, 2012; Harvey et al., 2014; Harvey & Mönig, 2010) er de mest benyttede og bliver brugt til at undervise i programmering fra grundskole og helt op til introducerende universitetskurser.

Men andre BBP'er har også været brugt i uddannelse, f.eks. Alice eller StarLogo TNG. Fordelen ved BBP er, at eleverne ikke selv skal skrive kode og derfor kan komme hurtigere i gang. Men der kan også være ulemper ved dette. Weintrop og Wilensky (2015; 2015) designede og implementerede et 9-ugers programmeringskursus, hvor eleverne kunne skifte mellem at bruge BBP og tekstbaserede sprog for at forstå, hvad disse ulemper kan være. Cetin (2016) forskede i om elever lærte bedst ved at programmere i det blokbaserede Scratch (Resnick et al., 2009) eller i det meget mere komplicerede, og tekstbaserede, sprog, C. Begge forskningsprojekter fandt, at BBP'er kan hjælpe elever i gang, men de langsigtede læringseffekter er stadig usikre.

A.3.2 Robotics som CT

En anden tilgang til at designe CT-aktiviteter har fokuseret på design og programmering af robotter.

I Danmark kender mange LEGO Mindstorm robotter (opkaldt efter Seymour Paperts bog fra 1980), og Mindstorms bliver brugt i klasseværelser som platform, f.eks. følgende:

- Atmatziodou et al. (2016) Mindstorms til at undervise i generelle Robotics- og programmeringsbegreber til gymnasieelever.
- Berland og Wilensky (2006) designede et undervisningsforløb, hvor eleverne skulle bygge autonome robotter, som konkurrerede med hinanden i robotfodbold.
- Bers viste med sit TangibleK-design (M. Bers, 2010), at det er muligt at undervise børnehavebørn i robotics, hvis programmeringssproget er designet specifikt til deres udviklingsniveau (Bers, Flannery, Kazakoff, & Sullivan, 2014).

- Holbert (2016) designede et læringsforløb til elever fra 5.-8. klasse, hvor de skulle designe robotter til yngre elever.

Tilsammen viser disse studier, at der er mange fordele ved at bygge robotter som en del af CT-læring: dels trækker det på generelle ingeniør-kundskaber, dels giver det eleverne nogle helt praktiske mål at arbejde henimod, og sidst men ikke mindst er den fysiske del af robotter motiverende for unge elever (Grover, 2011).

A.3.3 Computermodellering som CT

Computermodellering har også en lang tradition i CT-undervisning og har gennem de sidste 30 år vist sig at være et robust læringsværktøj på tværs af klasseniveauer og fag.

Et af de første modelleringssprog, som blev designet til børn og unge, var StarLogo (1996). StarLogo blev især brugt i 90'erne (Colella, Klopfer, & Resnick, 2001), men blev i starten af 00'erne erstattet af NetLogo (Wilensky, 1999) som det mest brugte modelleringssprog i uddannelse.

Computermodellering har ofte været set som et succesrigt læringsværktøj til fagkundskaber og har derfor været integreret i de fag, som emnet for modellen handler om.

Computermodellering har været brugt til at undervise i så forskellige fag som fysik (Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013; Sengupta & Wilensky, 2009), kemi (Levy & Wilensky, 2009a, 2009b), biologi (Centola, McKenzie, & Wilensky, 2000; Wagh & Wilensky, 2013) og samfundsfag (Hjorth & Wilensky, 2014; Hjorth, Brady & Wilensky, 2018).

I Danmark har NetLogo været brugt til at undervise i fag som biologi, kemi, samfundsfag og bioteknolog (Musaeus & Musaeus, 2019). De nyeste udviklinger af computermodellering og CT har i høj grad fokuseret på at udvikle blokbaserede sprog til modellering, f.eks. NetLogo Web (f.eks. Weintrop et al., 2016) og StarLogo TNG (Klopfer, Scheintaub, Huang, & Wendel, 2009).

A.3.4 Håndarbejde og STEAM som CT

STEAM er STEM + 'Arts', og handler om brug af CT og programmering til at lave kunst eller at bruge CT i håndværk. Fordi kunst og håndværk typisk ikke er en del af pensum på gymnasieniveau, finder STEAM-aktiviteter ofte sted i SFO'er osv., men i de sidste par år har der også været forsøg på at integrere dem i skolearbejde (Stevens et al., 2016). STEAM benytter ofte teknologi som f.eks. 3D printere eller laserskærere til at gøre det sjovt og spændende for unge at lære at bruge disse maskiner ifm. kunstneriske udtryk.



STEAM trækker i høj grad på det faktum, at computere er blevet meget billige i de senere år, og især micro-controllerne LillyPad og FLORA er blevet brugt til at undervise piger i CT ved at gøre det muligt at sy elektronik, f.eks. mikrofoner, tryksensorer, LED-lys osv., ind i tøj (Buechley, et al, 2013).

Kafai og Fields har brugt disse såkaldte e-tekstiler til at undervise piger i debugging (Fields, Searle, & Kafai, 2016; Kafai et al., 2014) – et centralt begreb i CT – og arbejder på at designe et formelt undervisningsforløb, som kan finde sted i klasseværelser og i skoler (Kafai et al., 2014) på gymnasieniveau.

A.3.5 Computerspildesign som CT

Computerspildesign som læringsmiddel går helt tilbage til de sene 80'ere og tidlige 90'ere, hvor Kafai forskede i, hvordan unge kunne lære programmering ved at designe deres egne spil (Kafai, 2006, 2012). På daværende tidspunkt var det dog de færreste, som havde adgang til computere. I takt med at computere og smartphones er blevet mere udbredte, er computerspil også blevet mainstream, og dette har CT-designere draget nytte af.

Gee (2003; 2005) var en af de første, som brugte kommercielle spil og spildesignsaktiviteter som læringsmiddel. Men på daværende tidspunkt var der kun meget få spildesignprogrammer til ikke-programmører, f.eks. det hollandske Game Maker. Men op igennem 00'erne og det indeværende årti er antallet af platforme, som gør det nemt at designe spil, eksploderet med f.eks. Scratch (Resnick et al., 2009), Processing (Shiffman, 2008), Processing.js (Resig, Fry, & Reas, 2008) og Unity (Haas, 2014). Derfor har spildesign som CT-læringsredskab gode vilkår for at blive en udbredt og accepteret læringsaktivitet.

A.3.6 Evaluering af CT og læring

Der er endnu ikke etablerede batterier af tests osv., som kan bruges til at evaluere læring af og med CT. Af den grund benytter forskellige forskningsprojekter forskellige typer af evidens og data til at måle læring. Vi gennemgår her kort de forskellige tilgange, som forskere har brugt.

A.3.6.1 Viden om CT eller programmering

Jeannette Wings første definition af CT handlede i høj grad om at forankre tænkning i Computer Science-begreber, og viden om disse grundlæggende begreber har været fokus for en del forskning i læringsevaluering.

F.eks. udviklede Román-González et. al. (Román-González, Pérez-González, & Jiménez-Fernández, 2017) et sæt af CT-spørgsmål, som måler gymnasieelevers forståelse af begreber som objekter, algoritmer, iteration, rekursion, events osv.

Evaluering af begrebslig viden er dog ikke forbeholdt de ældre elever. F.eks. har Brennan og Resnick også viden om disse begreber som en del af deres evaluering af børn helt ned til 5-årsalderen (Brennan & Resnick, 2012).

Andre gange bliver denne type af evaluering blandet med andre typer af viden. F.eks. er "algoritmisk viden" bare en ud af fem typer af viden, som Korkmaz et al. (Korkmaz, Çakir, & Özden, 2017) måler på, imens to af de fire andre er "kritisk tænkning" og "kreativitet".

På trods af et fokus på datalogi og programmering er det stadig ikke endeligt etableret præcist, hvilke programmerings- og datalogiske begreber man bør måle på, og dette diskuteres stadig i litteraturen (Grover & Pea, 2013; Grover & Pea, 2018). Der er dog bred enighed om, at det er vigtigt, at elever har en grundlæggende viden om disse centrale begreber.

A.3.6.2 Emotionel holdning og attitude til CT

En vigtig del af CT i undervisning er at få unge til at interessere sig for CT og få dem til at føle, at CT er relevant for dem og for deres fremtidige uddannelse eller job. Dette spørgsmål har da også fået mere og mere opmærksomhed i forbindelse med CT-interventionsstudier, men de præcise målinger har varieret afhængig af interventionens natur, og hvem den var rettet imod.

I projektet *ComputationalThinking4Girls* viste Brady et al. (2017), at deres intervention ændrede de deltagende kvindelige gymnasieelevers holdning til programmering og dataanalyse med computere, og at pigerne mente, at det var mere sandsynligt, at de ville vælge en CT-relateret uddannelse.

Cetins intervention var derimod rettet mod lærere, og her viste forskerne, at lærerne bagefter følte, at CT ville kunne være mere relevant i deres undervisning.

Yadav et al. (2014) og Kim et al. (2014) målte hhv. bachelorstuderendes og grundskoleelevers holdning til computere generelt og viste, at deltagerne syntes, at computere var mere spændende efter interventionen.

Weintrop og Wilenskys (2015, 2017) arbejde har især undersøgt forholdet mellem at bruge tekstbaserede eller blokbaserede programmeringssprog i klasseværelser, og i deres forskning har deres holdningsspørgsmål fokuseret på, om børn og unge foretrækker blok- frem for tekstbaseret programmering. Her fandt forskerne, at eleverne især i starten havde et lavere frustrationsniveau med blokbaseret programmering.

Det er stadig en relativt ny idé at måle emotionel holdning i CT-undervisning, men vi mener, at det er et vigtigt mål for alle CT-interventioner i skoler og uddannelse.

A.3.6.3 Viden om faget

Når CT er integreret i et andet fag, kan man evaluere læring med CT ved at evaluere læring i selve faget.

Som beskrevet i A.3.3 Computermodellering som CT, er der mange studier, hvori computermodellering optræder som CT-læringsværktøj til fagkundskaber. Evaluering i disse studier benytter som regel kvalitative interviews eller kvalitative analyser af skriftlige elevbesvarelser.

F.eks. undersøgte Wilkerson-Jerde og Wilensky (Wilkerson-Jerde & Wilensky, 2010) og senere Wagh og Wilensky (2014; 2013) gennem interviews, om elever bedre kunne forklare evolution efter at have programmeret computermodeller om emnet.

Lignende evalueringer er udført i forskellige emner såsom miljøpolitik (Gilligan, Brady, Camp, Nay, & Sengupta, 2015; Hjorth, Brady, & Wilensky, 2018) og fordeling af velstand (Guo & Wilensky 2018).

Når CT integreres i fag, flyttes læringsfokus naturligvis også hen mod fagets genstandsfelt. Der mangler dog stadig meget viden om, hvordan man bedst designer CT-værktøjer, der kan bruges som læringsværktøjer og som en integreret del af fagundervisning (Grover & Pea, 2013).

A.3.6.4 Performance tasks

En sidste type evaluering, som vi har set, er såkaldte performance tasks. Performance tasks handler ikke så meget om, hvad elever og studerende *ved*, men hvad de *kan*.

I modsætning til de fleste tests, som finder sted på papir eller gennem interviews, udføres performance tasks som regel i en autentisk kontekst og med de værktøjer (computere, programmeringsplatforme eller anden teknologi), som også blev benyttet i læringen.

F.eks. arbejder Kafai og Fields på at udvikle et sæt af performance tasks til deres e-tekstil-curriculum. Som tidligere nævnt indeholder e-tekstiler både computerkode i Arduino-sproget og et fysisk circuit, som er syet ind i tøj eller andre tekstiler, og hver performance task indeholder en eller flere bugs, som enten er i koden, i det fysiske circuit eller begge steder. Disse opgaver er designet, så eleverne først skal diagnosticere problemet og så derefter løse det.

Et lignende sæt problemer finder vi i Werner et al. (2012), som i deres spilprogrammeringskursus har lavet programmer med små fejl, som eleverne skal finde og rette.

Performance tasks er interessante, fordi de tester elevernes kompetencer i praksis, men vi mangler stadig viden om, hvordan man bedst designer dem, så de mere præcist måler, hvad eleverne kan.

A.3.7 Tid til at stille store spørgsmål

Som beskrevet ovenfor er der mange interessante aktiviteter, som forsøger at finde frem til CT's væsen og identitet. Men fordi dette felt endnu er meget ungt, er vi stadig langt fra at have en solid basis af forskning på 'CT i uddannelse' området, og der bør derfor stadig forskes i og stilles spørgsmål til områder, som endnu ikke er blevet tilstrækkeligt udforsket.

Disse områder omhandler primært:

1. Læringsteoretiske CT-principper
2. Evaluering
3. Underviseres CT-kundskaber
4. Nye værktøjer
5. Effekter af CT i undervisningen
6. Kønsforskelle og uddannelsesfremmede elever

I det følgende diskuteres nogle af de store spørgsmål i relation til hvert af disse områder.

A.3.7.1 Læringsteoretiske CT-principper

Et af de helt fundamentale områder, som behøver meget mere opmærksomhed end hidtil, er spørgsmålet om, hvad CT er, og hvordan dette bedst læres (Guzdial, 2018 In: Blikstein, 2018; Grover & Pea, 2013).

Wing påpegede i 2011, at der er behov for at anvende lærings- og kognitionsteori i design, udvikling og forskning af CT i undervisning.

Undersøgelser, som kan afdække sammenhængen mellem læring af elementer af CT-viden og faktorer såsom studerendes alder og klassetrin, er næsten fraværende. Flere forskere foreslår at bygge videre på succesfulde undersøgelser fra 1980'erne og 90'erne af kognitive aspekter af børns og nybegynderes tilegnelse af CT for at besvare spørgsmål som 'what, when and how' om CT i undervisning (Tedre & Denning, 2017).

Empiriske studier, der undersøger, hvilke CT-begynderproblemer elever har i starten af tilegnelsen af CT, og hvordan disse kan overkommes, kan vise sig at blive vigtige brikker i puslespillet om CT i undervisning, både i forbindelse med CT implementeret i eksisterende fag og som eget fag (Grover & Pea, 2013).

De store spørgsmål omhandler hvad, hvornår og hvordan. Hvilke didaktiske principper kan fungere som praktisk anvendelige guidelines til design af gode og spændende CT-læringsaktiviteter?

A.3.7.2 Evaluering

CT er endnu ikke veldefineret, og der er ikke konsensus om, hvilke begreber i CT der er de primære. Det gør det selvsagt svært at designe evalueringer, der måler CT-indhold, begreber og sammenhænge (concept inventories). Selv for andre emner, der er langt mere veldefinerede, er det en kompleks opgave at udvikle sådanne 'concept inventories'.

Der bør dog være fokus på udviklingen af sådanne tests for at muliggøre en reel sammenligning af forskellige pædagogikker til træning af CT-begreber og sådannes effektivitet (Blikstein, 2018; Shute et al., 2017; Grover & Pea, 2013).

Inden for fysik er der udviklet bredt accepterede concept inventories inden for afgrænsede og veldefinerede emner som f.eks. kraft og magnetisme, men det er også en faglighed, som der er endog meget stor konsensus om, både mht. hvad man underviser i, og hvornår man gør det.

Nogle forskere påpeger, at det bør undersøges, hvorvidt det er muligt at lave 'real-time' tests af studerende, som kan kvalificere instruktionerne i undervisningen (Shute et al., 2017).

Design og validering af evalueringer er en af de første trædesten i brolægningen af vejen mod at få CT ind i undervisningen i mange discipliner.

Det store spørgsmål er, hvordan designer vi evalueringer, som måler på de mange og alsidige aspekter af CT?

A.3.7.3 Underviseres CT-kundskaber

Undersøgelser tyder på, at undervisere generelt er ubekendte med CT's natur og væsen, og derfor også har svært ved at finde sammenhængen mellem CT og deres egen undervisning (Yadav et al., 2017). Der er således et behov for at afdække, hvordan undervisere bliver bedre klædt på til at planlægge, gennemføre og evaluere deres CT-undervisning.

Spørgsmål som, hvad har underviserne brug for, hvilken CT-didaktik passer med underviseres tilgang til undervisning, og hvilke CT-beslægtede aktiviteter anvender underviserne allerede, er vigtige i denne sammenhæng (Grover & Pea, 2013).

Også opmærksomhed på opbygningen af ressourcer til undervisningsbrug nævnes som et vigtigt fokuspunkt (Grover & Pea, 2013).

Sidst, men ikke mindst er der implementeringsspørgsmål i forhold til institutionel støtte helt fra lærerværelset og til undervisningsministeriet.

Det store spørgsmål er altså, hvordan skaleres CT op, med hjælp fra undervisere, på nationalt niveau og på tværs af uddannelsesniveauer?

A.3.7.4 Nye værktøjer

Som Grover & Pea allerede i 2013 påpegede, er der brug for udvikling af nye værktøjer til brug i CT-undervisningen. På nær nogle få veletablerede værktøjer (såsom NetLogo) er de fleste eksisterende værktøjer ikke oprindeligt designet til undervisning. Dette betyder, at undervisere i f.eks. humanistiske og kreative fag kan have svært ved at finde brugbare og tilgængelige værktøjer til deres undervisning. På trods af dette behov mangler der desværre ofte frugtbare forbindelser mellem forskere, udviklere og designere, som kan sikre udviklingen af sådanne værktøjer (Blikstein, 2018).

Hvordan skaber vi et miljø, hvor forskere, undervisere, designere og udviklere af CT-teknologier sammen kan bidrage til udvikling af og forskning i CT-værktøjer og -aktiviteter?

A.3.7.5 Effekter af CT i undervisningen

Undersøgelser, der forsøger at afdække, hvorvidt elever bruger CT-kompetencer fra deres undervisning i andre dele af deres liv eller i deres videre uddannelse og karriere, er endnu et område, der bør belyses. Der er få lovende eksempler (Grover et al., 2015), men langtidseffekterne af CT-undervisning er meget underbelyst.

Så hvorledes kan vi måle eller sandsynliggøre, om elever anvender CT-kompetencer uden for den specifikke undervisningskontekst, og i så fald hvordan og med hvilken effekt?

A.3.7.6 Kønsforskelle og uddannelsesfremmede elever

Endnu et område, der fortjener betydelig opmærksomhed, er kønsforskelle i udviklingen af CT-kompetencer (Grover & Pea, 2013). Kvinder er ofte underrepræsenteret på STEM-relaterede uddannelser, og CT bliver foreslået som motivation for at få flere piger og kvinder ind på disse studier (Shute et al., 2017).

På trods af at piger tidligt i skolegangen klarer opgaver i naturvidenskab og matematik bedre end drenge og får bedre karakterer, hvilket til dels gør sig gældende helt op på gymnasie- og collegeniveau (Valla & Williams, 2012; Rees et al. 2016), så viser undersøgelser også, at pigers selvsikkerhed i matematik og i de naturvidenskabelige fag daler i forhold til drengenes allerede fra omkring 5.-6. klasse (Seneviratne, 2017).

Svar på spørgsmål såsom, hvad er studerendes opfattelser af og holdninger til CT, og hvordan relaterer disse sig til de studerendes identitet, er vigtige i denne sammenhæng. CT-undervisning bliver også foreslået som en mulig motivation for uddannelsesfremmede studerende for derved at kunne fastholde og opkvalificere disse studerende.

Interventionsstudier med CT-aktiviteter, baseret på konstruktivistiske læringsprincipper, har vist sig at have en positiv effekt (dog ikke signifikant) på svage studerende og på frafaldsprocenten (Lye & Koh, 2014).

Ellers er det sparsomt med undersøgelser, både internationalt og nationalt, omkring motivering og engagering af elever fra uddannelsesfremmede hjem. Undersøgelser har vist, at elever af forældre fra lavindkomstgrupper lægger større vægt på værdien af at lære at kode end elever af forældre fra højindkomstgrupper (della Cava, 2015).

Således har nylige tiltag i USA, såsom Code.org, CodeNow og GirlsWhoCode, formulerede hensigter som "expanding access to computer science" og "increasing participation by women and underrepresented minorities." (The digital promise, 2017). Disse tiltag fokuserer på at gøre læring om CT tilgængelig for alle elever i uddannelsessystemet.

På den måde ønskes at udforske det potentiale, som CT forventes at have, og som Margolis og Kafai i 2016 ("The digital promise, p. 12) beskrev i form af en øget deltagelse fra ellers underrepræsenterede grupper i datalogi og naturvidenskab. De skrev bl.a.: "Computer science can help interrupt the cycle of inequality that has determined who has access to this type of high-status knowledge in our schools."

Her er det store spørgsmål, hvordan kan vi skabe en kultur omkring CT, som kan gøre CT relevant og engagerende for flere grupper af elever?

* * *

Disse store spørgsmål udspænder feltet af både praktiske implementerings- og designspørgsmål, teoretiske forskningsspørgsmål og de helt store demokratiske implikationer af CT i uddannelse og i samfundet.

A.3.8 Referencer for appendiks A.3

- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661–670.
- Berland, M., & Wilensky, U. (2005). Complex play systems--Results from a classroom implementation of VBOT. In W. Stroup and U. Wilensky (Chairs) & CD Lee (Discussant), Patterns in group learning with next-generation network technology. *The annual meeting of the American Educational Research Association, Montreal, Canada*
- Berland, M., & Wilensky, U. (2006). Constructionist collaborative engineering: Results from an implementation of PVBOT. In *annual meeting of the American Educational Research Association, San Francisco, CA*.
- Bers, M. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, 12 (2), 1–20.
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145–157.
- Blikstein, P. (2018). Pre-College Computer Science Education: A Survey of the Field. *Mountain View, CA: Google LLC*. Retrieved from <https://goo.gl/gmS1Vm>.
- Brady, C., Orton, K., Weintrop, D., Anton, G., Rodriguez, S., & Wilensky, U. (2017). All roads lead to computing: Making, participatory simulations, and social computing as pathways to computer science. *IEEE Transactions on Education*, 60(1), 59–66.
- 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).
- Buechley, L., Peppler, K., Eisenberg, M., & Yasmin, K. (2013). Textile Messages: Dispatches from the World of E-Textiles and Education. *New Literacies and Digital Epistemologies*. Volume 62. ERIC.
- Centola, D., McKenzie, E., & Wilensky, U. (2000). Survival of the Groupiest: Facilitating Students' Understanding of Multi-level Evolution through Multi-Agent Modeling - The EACH Project. In *Proceedings of The Fourth International Conference on Complex Systems*. NH: New England Complex Systems Institute.
- Cetin, I. (2016). Preservice Teachers' Introduction to Computing: Exploring Utilization of Scratch. *Journal of Educational Computing Research*, 54(7), 997–1021.
- Colella, V. S., Klopfer, E., & Resnick, M. (2001). Adventures in Modeling: Exploring Complex, Dynamic Systems with StarLogo. ERIC.
- della Cava, M. (2015). Should students learn coding? Students, schools disagree, poll finds. *USA Today*. August 20th, 2015.
- diSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy*. Mit Press.
- diSessa, A. A., & Abelson, H. (1986). Boxer: A reconstructible computational medium. *Communications of the ACM*, 29(9), 859–868.
- Fields, D. A., Searle, K. A., & Kafai, Y. B. (2016). Deconstruction Kits for Learning: Students' Collaborative Debugging of Electronic Textile Designs. In *Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education* (pp. 82–85). ACM.
- Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1), 20.

- Gee, J. P. (2005). Learning by design: Good video games as learning machines. *E-Learning and Digital Media*, 2(1), 5–16.
- Gilligan, J. M., Brady, C., Camp, J. V., Nay, J. J., & Sengupta, P. (2015). Participatory simulations of urban flooding for learning and decision support. In *Proceedings of the 2015 Winter Simulation Conference* (pp. 3174–3175). IEEE Press.
- Grover, S. (2011). Robotics and engineering for middle and high school students to develop computational thinking. In *annual meeting of the American educational research association, New Orleans, LA*.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), pp 38–43.
- Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199–237.
- Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School*, 19.
- Guo, Y., & Wilensky, U. (2018a). Mind the gap: Teaching high school students about wealth inequality through agent-based participatory simulations. In *Proceedings of the Constructionism 2018 conference*. Vilnius, Lithuania.
- Haas, J. K. (2014). A history of the Unity game engine.
- Harvey, B. (2012). The beauty and joy of computing: Computer science for everyone. In *Proceedings of Constructionism*, 33–39.
- Harvey, B., Garcia, D. D., Barnes, T., Titterton, N., Miller, O., Armendariz, D., ... others. (2014). Snap!(build your own blocks). In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 749–749). ACM.
- Harvey, B., & Mönig, J. (2010). Bringing “no ceiling” to scratch: Can one language serve kids and computer scientists. *Proc. Constructionism*, 1–10.
- Hjorth, A., & Wilensky, U. (2014). Redesigning Your City-A Constructionist Environment for Urban Planning Education. *Informatics in Education*, 13(2), 197.
- Hjorth, Brady, C., & Wilensky, U. (2018). Sharing is Caring in the Commons – Students’ Conceptions about Sharing and Sustainability in Social-Ecological Systems. *Presented at the Constructionism 2018, Vilnius, LT*.
- Holbert, N. (2016). Bots for Tots: Building Inclusive Makerspaces by Leveraging Ways of Knowing. In *Proceedings of the The 15th International Conference on Interaction Design and Children* (pp. 79–88). ACM.
- Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture*, 1(1), 36–40.
- Kafai, Y. B. (2012). *Minds in play: Computer game design as a context for children’s learning*. Routledge.
- Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1.
- Kafai, Y. & Margolis, J. (2014). Why the ‘coding for all’ movement is more than a boutique reform. *Washington Post*. October 7th, 2014.
- Kim, Y. C., Lee, W. G., & others. (2014). Computational modeling and simulation for learning an automation concept in programming course. *International Journal of Computer Theory and Engineering*, 6(4), 341.

- Klopfer, E., Scheintaub, H., Huang, W., & Wendel, D. (2009). StarLogo TNG. In *Artificial Life Models in Software* (pp. 151–182). Springer.
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, 72, 558–569.
- Levy, S., & Wilensky, U. (2009). Students' learning with the Connected Chemistry (CC1) curriculum: navigating the complexities of the particulate world. *Journal of Science Education and Technology*, 18(3), 243–254.
- Levy, Sharona, & Wilensky, U. (2009). Crossing levels and representations: The Connected Chemistry (CC1) curriculum. *Journal of Science Education and Technology*, 18(3), 224–242.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
- Musaeus, L. H., Musaeus, P. (2019). Computational thinking in the Danish High School: Learning Coding, Modeling, and Content Knowledge with NetLogo. In: *SIGCSE '19: SIGCSE '19: The 50th ACM Technical Symposium on Computer Science Education, February, 2019, Minneapolis, Minnesota, USA*.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, New York: Basic Books, Inc.
- Papert, S. (1994). *The Children's Machine: Rethinking School In The Age Of The Computer*. New York, Basic Books, Inc.
- Reas, C., & Fry, B. (2007). *Processing: a programming handbook for visual designers and artists*. Mit Press.
- Rees, A., García-Peñalvo, F. J., Jormanainen, I., Tuul, M., & Reimann, D. (2016). *An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers*. Belgium: TACCLE3 Consortium. doi:10.5281/zenodo.165123.
- Resig, J., Fry, B., & Reas, C. (2008). *Processing.js*.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... others. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60–67.
- Resnick, M. (1996). StarLogo: an environment for decentralized modeling and decentralized thinking. In *Conference companion on Human factors in computing systems ACM* (pp. 11–12).
- Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691.
- Seneviratne, O. (2017). Making Computer Science Attractive to High School Girls with Computational Thinking Approaches: A Case Study. In *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 21–32). Springer, Cham.
- 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., & Wilensky, U. (2009). Learning Electricity with NIELS: Thinking with Electrons and Thinking in Levels. *International Journal of Computers for Mathematical Learning*, 14(1), 21–50.
- Shiffman, D. (2008). *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*, Burlington, USA.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*.

- Stevens, R., Jona, K., Penney, L., Champion, D., Ramey, K. E., Hilppö, J., ... Penuel, W. (2016). FUSE: An alternative infrastructure for empowering learners in schools. Singapore: *International Society of the Learning Sciences*.
- Tedre, M., & Denning, P. J. (2017). The long quest for computational thinking. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 120-129). ACM.
- The digital promise (2017). Computational thinking for a computational world. The digital promise. *Accelerating Innovation in Education* (pp. 11-13).
- Valla, J. M., & Williams, W. M. (2012). Increasing achievement and higher-education representation of under-represented groups in science, technology, engineering, and mathematics fields: A review of current K-12 intervention programs. *Journal of women and minorities in science and engineering*, 18(1).
- Wagh, A, & Wilensky, U. (2014). Seeing patterns of change: Supporting student noticing in building models of natural selection. In: *Proceedings of 2014 constructionism*. Vienna.
- Wagh, Aditi, & Wilensky, U. (2013). Leveling the playing field: Making multi-level evolutionary processes accessible through participatory simulations. In *Proceedings of the Biannual Conference of Computer-Supported Collaborative Learning (CSCL), Madison, Wisconsin*.
- Weintrop, D., Hjorth, A., Brady, C., & Wilensky, U. (2016). NetLogo Web: Bringing Turtles to the Cloud. *Constructionism 2016*, Bangkok, Thailand.
- Weintrop, D. (2015). Blocks, text, and the space between: The role of representations in novice programming environments. In *Visual Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on* (pp. 301–302). IEEE.
- Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 199–208). ACM.
- Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 18(1), 3.
- 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). ACM.
- Wilensky, U. (1999). NetLogo: Center for connected learning and computer-based modeling. *Northwestern University, Evanston, IL*.
- Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. *Constructionism 2010*, Paris, France.
- Wilkerson-Jerde, M. H., & Wilensky, U. (2010). Restructuring change, interpreting changes: The deltatick modeling and analysis toolkit. *Proceedings of Constructionism*.
- Wing, J. (2011). Research notebook: Computational thinking—What and why? *The Link Magazine*, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from <http://link.cs.cmu.edu/article.php?a=600>
- 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), 5.
- Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 205-220). Springer, Cham.

A.4 Tentativ dansk taksonomi for CT i uddannelser

I regi af It-vest arbejder Syddansk Universitet, Aalborg Universitet og Aarhus Universitet på at formulere en dansk taksonomi for CT i universitetsuddannelser.

Taksonomien er bevidst holdt på et forholdsvis generelt niveau, idet tanken er, at taksonomien skal være en ramme på tværs af uddannelser, men en ramme som også skal specialiseres til den konkrete kontekst.

Der arbejdes "bottom-up" med dette, i første omgang med udgangspunkt i hhv. humanistiske og samfundsvidenskabelige cand.it.-uddannelser, men det er tanken efterfølgende at udbrede perspektivet til at dække universitetsuddannelser generelt.

Foreløbig har arbejdet resulteret i formulering af (tentative versioner af)

1. Forudsætningskrav til studerende (og tilbud, så de kan opfyldes/efterleves)
2. CT-kompetencebeskrivelse for humanistiske it-uddannelser
3. Principper for organisatorisk integrering af CT-faglighed
4. Tentativ taksonomi/ramme for CT i humanistiske it-uddannelser

A.4.1 Forudsætningskrav til studerende (og tilbud så de kan opfyldes/efterleves)

Studerende antages at have en forståelse for programmering og programmeringssprog (tilsvarende ungdomsuddannelsernes Informatik på C-niveau).

De studerende skal kunne argumentere gennem tilvejebringelse af interaktive elementer af en horisontal prototype.

It-vest udbyder ressourcer til studerendes egen opkvalificering (website, der samler gode eksisterende ressourcer til opkvalificering).

A.4.2 CT-kompetencebeskrivelse for humanistiske it-uddannelser

Studieordningernes kompetenceprofiler for humanistiske it-uddannelser suppleres med følgende:

Den studerende skal kunne

- anvende computermødelser til at opnå ny erkendelse af eget fagfelt
- forstå computationelle tankeprocesser
- gennemføre computationelle tankeprocesser og praksisser
- tilegne sig et sprog (verbal og programmering) for at kunne forberede og konstruere digitale artefakter
- argumentere for designprocessen og den endelige løsning gennem proces og data
- konstruere eller redesigne et interaktivt digitalt artefakt til et proof of concept
- kritisk evaluere egne og andres design i forhold til de etiske (og politiske) aspekter i forhold til egen praksis, fællesskab og samfund.

A.4.3 Principper for organisatorisk integrering af CT-faglighed

Det er vigtigt, at ledelsen på universiteterne forstår CE's (Computational Empowerment) vigtighed, prioriterer dens betydning OG er klar til at bruge ressourcer på dens udvikling i de respektive uddannelser.

Fagligheden integreres gennem uddannelsens forskellige fag.

Undervisere, der ikke nødvendigvis har den teknologiske ekspertise, teams op med kollegaer med stærkere it-kompetencer.

Undervisere, der ikke nødvendigvis har den humanistiske ekspertise, teams op med kollegaer med stærkere humanistiske kompetencer.

A.4.4 Tentativ taksonomi for CT i humanistiske it-uddannelser

Den tentative taksonomi er formuleret med afsæt i fem temaer. De fire sidstnævnte er CT-praksisser ekstraheret fra artiklen *Defining Computational Thinking for Mathematics and Science Classrooms* (Weintrop et al., 2016).

De fem tentative temaer er:

- Problem framing
- Data og informationsprocesser
- Modellering og simulering
- Computational undersøgelse og problemløsning
- Systemtænkning

De fem temaer udfoldes yderligere i det følgende:

A.4.4.1 Problem framing

- Forståelse for partcipatoriske designprocesser
- Empatisk domæneforståelse (problemer opstår ikke i tomrum)
- Evnen til på baggrund af viden at formulere et problem, der kan undersøges og løses
- Erkende og undersøge forforståelser som baggrund for problemløsning
- Iteration; problem framing foregår gennem processen.

A.4.4.2 Data og informationsprocesser

- Skabe knotwork og networks af stakeholders som udgangspunkt for at erkende verden
- Forstå den eksisterende teknologi og forholdet mellem teknologi og brugskontekst (praksis)
- Forståelse for data og kvalificering af data
- Samle, skabe, analysere, manipulere, transformere og visualisere data.

A.4.4.3 Modellering og simulering

- Brug af computationelle modeller til at forstå begreber samt at finde og teste løsninger
- Designe, konstruere og vurdere computationelle modeller
- Kunne forstå og beskrive computermodellens begrænsninger.

A.4.4.4 Computational undersøgelse og problemløsning

- Forberede problemer til computationelle løsninger i samarbejde med brugerne
- Mock-up og prototyping herunder collaborative prototyping (eksternalisering af systemperspektiver)
- Vælge effektive og relevante computationelle værktøjer
- Vurdere forskellige tilgange/løsninger til et problem
- Udvikle modulære computationelle løsninger
- Undersøge, hvilke andre problemer løsningen skaber
- Skabe computationelle abstraktioner
- Undersøge, om løsningen har en tilsigtet effekt på det valgte problem
- Fejlfinding og -retning.

A.4.4.5 Systemtænkning

- Forståelse for teknologihistorie og filosofi
- Analyse/forståelse af komplekse systemer på forskellige abstraktionsniveauer
- Forstå, beskrive og definere komplekse systemer i termer af fænomener og begreber samt deres relationer i systemet.