

Teknologiforståelse som undervisningsfag på læreruddannelsen

Studerendes værdisætning af programmering som et badge of membership

Thilde Emilie Møller, Professionshøjskolen Absalon
Mads Middelboe Rehder, Københavns Professionshøjskole
Marie Falkesgaard Slot, Københavns Professionshøjskole
Lars Bo Andersen, Københavns Professionshøjskole

Abstract

Artiklen bidrager med viden om studerendes teknologifaglige arbejdsprocesser i undervisningsfaget teknologiforståelse på læreruddannelsen på første semester. I tre cases undersøges det, hvordan programmeringsfærdigheder værdisættes blandt de studerende, hvilket belyses med udgangspunkt i Andrea diSessas teori om *computational literacy*. Case 1 viser en gruppe engageret i en designproces, hvor programmering til deres store ærgrelse ligger uden for deres faglige rækkevidde. Case 2 viser, at det er muligt for en gruppe at forestille sig et programmeret udtryk, selvom færdighederne ikke er til det. Case 3 viser, hvordan en gruppe programmerer sig frem til en løsning med god understøttelse af medstuderende og facilitatorer af aktiviteten. Artiklen konkluderer, at der i udviklingen af en fagkultur eksisterer en specifik værdisætning af enkelte teknologier og færdigheder, hvor særligt programmering med diSessas betegnelse kan blive et *badge of membership* for de studerende. Dette må håndteres af både studerende og undervisere i udvikling af faget og fagligheden.

Engelsk abstract

The article contributes with knowledge regarding students' technological work processes in the teaching subject of technology comprehension in the first semester of the teacher education program. Through the examination of three cases, it is investigated how programming skills are highly valued among the students, illuminated by Andrea diSessa's theory of computational literacy. Case 1 demonstrates a group engaged in a design process where programming lies outside their academic project. Case 2 illustrates that it is possible for a group to envision a programmed expression, even if the skills are not yet acquired. Case 3 demonstrates how a group progresses towards a solution through the utilization of peers' skills in programming. The article concludes that in the development of an academic culture, there exists a specific valuation of some technologies and skills, where programming in particular becomes a badge of membership for the students (as defined by diSessa). This necessitates attention from both students and educators.



Indledning

Teknologiforståelse er et nyt undervisningsfag, der afprøves på en dansk læreruddannelse. Faget lægger sig i tråd med teknologiforståelse i folkeskolen, der blev afprøvet og afviklet som forsøgsfag på 46 skoler i perioden 2018 og 2021. Evalueringen herfra peger på, at lærerne finder det krævende at undervise i teknologiforståelse som selvstændigt fag, og at særligt computationel tankegang og programmering er komplicerede fagområder for lærerne i skolen (Rambøll, 2021).

Teknologiforståelse er en faglighed, der trækker på et bredt repertoire af humanistiske, samfundsfaglige og naturfaglige teorier og praksisser. Det er beskrevet i forskningslitteraturen om den danske forsøgsfaglighed, at der foregår en forhandling mellem flere forskellige teknologiforståelser, og at det er kompliceret at forene de humanistiske, samfundsfaglige og naturfaglige tilgange i et fælles genstandsfelt (Nørgård, 2020; Andersen, 2021; Danholt, 2021).

Faget er under udvikling og uden en etableret faglig progression fra grunduddannelse til videregående niveau. De studerende på teknologiforståelse på læreruddannelsen kommer med forskellige fagforståelser og forudsætninger for at deltage i undervisningen. I dette studie har de studerende forinden afsluttet et obligatorisk specialiseringsmodul i teknologiforståelse på læreruddannelsen. De har tidligere stiftet bekendtskab med fagligheden, men ud fra meget forskellige perspektiver i og med de har haft undervisning på forskellige hold med forskellige undervisere, som også er nye i fagligheden. De studerende er afsøgende på faget og fagligheden, og hvilken rolle de forskellige faglige elementer spiller samt deres indbyrdes styrkeforhold.

Artiklens formål er gennem en undersøgelse af de studerendes teknologifaglige arbejdsprocesser på første semester at få et indblik i, hvordan lærerstuderende engagerer sig i den nye faglighed og hvordan denne faglighed manifesterer sig i konkrete undervisningspraksisser, hvilket kan bruges i arbejdet med fagets videre udvikling.

Artiklen viser gennem tre cases, at der opstår en værdisætning af visse teknologier og at programmeringsfærdigheder spiller en særlig rolle i de studerendes engagement i et fag, der forsøger at sidestille humanistiske, samfundsfaglige og naturfaglige teorier og tilgange i et ligeværdigt forhold. Det er dette særlige, som artiklen behandler og analyserer. Artiklen er guidet af følgende forskningsspørgsmål:

Hvilken rolle spiller programmering i de studerendes arbejde på første semester i undervisningsfaget teknologiforståelse på læreruddannelsen?

De tre cases er udviklet på baggrund af en visuel etnografisk metode, hvor der foretages en eksplicit og gradvis indsnævring af et analytisk fokus. Dette fokus er indsnævret til at omhandle de studerendes oplevelser af og arbejde med programmering, og hvordan disse praksisser kan forstås.

For at kunne gribe den emergende teknologiforståelsesfaglighed og undersøge de studerendes arbejdsprocesser med særligt fokus på programmering bygges der på et teoretisk rammeværk, der retter sig mod programmering som særlig udtryksform i faglige kulturer. Artiklen trækker derfor på fysiker og uddannelsesforsker Andrea diSessa's teori om *computational literacy* (diSessa 2000) samt nyere bearbejdnings heraf fra Roland Hachmann (2023) og Nina Bonderup Dohn (2021).



Teknologiforståelse - et nyt undervisningsfag på læreruddannelsen

Undervisningsfaget teknologiforståelse på læreruddannelsen opererer med fire overordnede kompetenceområder: digital myndiggørelse, digitale arbejdsprocesser, computationel tankegang og teknologisk handleevne. Disse skal anskues som symfoniske og komplimentære og er overordnet set de samme kompetenceområder, som forsøgsfaget i grundskolen indeholder. I skolen hedder det ene kompetenceområde dog ”digitalt design” og ikke ”digitale arbejdsprocesser” og i læreruddannelsen er niveauet selvfølgelig på et videregående niveau og med et styrket fokus på lærerfaglige og professionsdidaktiske kompetencer.

I studieordningen står, at de studerende på første semester skal arbejde med teknologifilosofi og teknologiforståelser, didaktiske grundpositioner og fagforståelser. De begynder her at arbejde med spørgsmålet: ”Hvad er teknologi?” De studerende skal derudover arbejde med basal computationel tankegang og teknologisk handleevne under spørgsmålet ”Hvad er en computer, et netværk og data?”. Tilmed skal de arbejde med designtænkning, kritisk design, æstetiske læreprocesser og teknologi, hvor fokus er på de studerendes egne processer og erfaringer. Første semester indeholder også et fokus på teknologiforestillinger og digital myndiggørelse herunder elevforståelser i teknologifagligheden samt et fokus på de studerendes egne erfaringer. Programmering er også repræsenteret i forbindelse med kompetenceområdet teknologisk handleevne, men overordnet set er programmering ikke i centrum på første semester.

Fagets kompetenceområder indeholder derved det multidisciplinære i form af både det materielle, designmæssige, myndiggørende og computationelle og det er en bærende pointe, at det aldrig alene er et mål, at de studerende skal kunne programmere. Intentionen i undervisningsfaget er således at skabe et symfonisk samspil mellem de fire kompetenceområder, og artiklens fokus på programmering er guidet af, at programmeringsfærdigheden gentagne gange viser sig interessant i de studerendes arbejde med faget.

Programmering optræder også i kompetenceområderne for teknologiforståelse i folkeskolen, hvor programmering også er placeret under kompetenceområdet teknologisk handleevne blandt kompetencer som fx opsætning af netværk, mens de mere analytiske aspekter af programmering som fx modellering og udvikling af algoritmer er placeret i et andet kompetenceområde, nemlig computationel tankegang. Det komplekse forhold mellem computationel tankegang og programmering afspejler sig både i kompetenceområderne for teknologiforståelse i folkeskolen og på læreruddannelsen.

Teori

Computationel tankegang og programmering

Computationel tankegang (CT) har i de seneste årtier fået fornyet opmærksomhed i uddannelsessektoren. Jeanette Wing (2006; 2011) genintroducerede begrebet i 2006 og argumenterede for, at alle fagdomæner burde opøve kompetencer til at formulere problemer på en måde, der gør dem håndterbare for en computer og computationelle beregninger. Dette er over tid blevet en central tankegang, som uddannelsespraksis og -politik i stigende grad har omfavnet, så det flere steder bliver betragtet som en grundlæggende kompetence, der skal styrkes gennem alle elevers skolegang.



CT er dog ikke nyt inden for uddannelse og kan spores tilbage til Seymour Paperts arbejde i 1980 med undersøgelse af forholdet mellem programmering og tænkning. Han argumenterede for, at elevernes arbejde med programmering kunne lette tænkning og læring på tværs af flere discipliner som matematik, naturvidenskab og litteratur (Papert, 1980). CT blev i den forbindelse forstået og praktiseret med et stærkt teknologisk fokus, der i høj grad afhang af programmeringsfærdigheder. Netop denne tilgang har været kritiseret for at være for snæver og i strid med den bredere opfattelse af CT som en tilgang, der fremmer analytisk og kritisk tænkning. I takt med den fornyede opmærksomhed i uddannelsessektoren, både politisk og i praksis, er CT i stigende grad blevet koblet til en række tænke- og problemløsningsfærdigheder, der gør brug af konceptuelle værktøjer som logik, algoritmer, nedbrydning og abstraktion, og er således i stigende grad særligt i Danmark blevet frakoblet de mere teknologisk orienterede programmeringsfærdigheder (Shute et al., 2017; Bocconi et al. 2018; 2022).

I en oversigt over forskellige computational thinking-begreber illustrer Nina Bonderup Dohn, hvordan disse er dybt indfildret i hinanden og ofte koblet til konkrete kompetencer og aktiviteter (Dohn, 2021). Her skelner hun blandt andet mellem analoge og digitale arbejdsprocesser, samt den grundlæggende forskel mellem at bruge programmer og kode programmer. Det betyder også, at hun skelner mellem CT som en analog tilgang, der kobles til ”brug af programmer”, mens computational literacy er placeret som en digital tilgang koblet til ”kodning af programmer”.

I det perspektiv kan CT i den danske uddannelsessektor overordnet forstås på flere måder og med forskellige grader af kobling til programmeringsfærdigheder overfor en mere generel tænke- og problemløsningsfærdighed.

Computational literacy – programmering på skemaet

Den amerikanske fysiker og uddannelsesforsker Andrea diSessa har udviklet en teori om computational literacy, der beskriver programmering som en særlig form for literacy. Denne literacy kan give adgang til de overordnede computationelle udtryksmuligheder, som den øvrige litteratur om computationel tankegang også beskæftiger sig med. Der opstår dog uenighed om, i hvilket omfang man skal betone programmeringskompetencer som del af de computationelle muligheder (Caeli & Bundsgaard, 2019). DiSessas begreb om computational literacy består af tre forbundne aspekter: det kognitive, det sociale og det materielle. Det kognitive aspekt er realiseringen af bestemte faglige forståelser og tænkemåder, som computationel tankegang fx er styret af. Det sociale aspekt handler om, hvordan vi forhandler og kommunikerer ved hjælp af især materialer og forstår materialer. Det materielle aspekt er bl.a. de materialer (fysiske og intellektuelle) eller ting, som vi tænker og udtrykker os med (fx gennem sproget) (diSessa, 2000).

I et computational literacy-perspektiv er programmering en fundamental og magtfuld literacy, der kan og ifølge diSessa bør læres tidligt, fordi programmering kan kobles til udtryksfuldhed og mere generiske tekstkompetencer i alle discipliner. Programmering kan ifølge diSessa forstås som den materielle basis (Dohn, 2021) i forbindelse med at kunne udtrykke sig. Her kommer materiel intelligens i spil, defineret ved det at kunne udtrykke sig relationelt og kognitivt gennem ting og tings udtryksfuldhed:

“We don’t always have ideas and then express them in the medium. We have ideas with the medium.” (diSessa, 2000, s. 116).

Materiel intelligens peger på, at det er i relationen mellem menneske og ting opstår det, som diSessa kalder ideer med ”ting”. Med ting mener diSessa computerteknologiens hardware og software, de fysiske og ikke-fysiske ting, som vi har, og tilmed indgår fx også sproget som en mere abstrakt ting, som tidligere præsenteret.

I Hachmanns undersøgelse og analyser af teknologiforståelsesfagligheden i skolen fremhæves især det materielle aspekt af diSessas tre aspekter af computational literacy, og der argumenteres for, at



materialitet spiller en vigtig rolle i forhold til de stof- og indholdskriterier, der er afsættet for undervisningen (Hachmann, 2023).

“Computational literacy handler ikke blot om at kunne læse, skrive eller regne på en computer, men derimod om helt bestemte praksisser og aktiviteter, hvor forskellige typer af computere eller software indgår som en del af deltagelsen. For eksempel er kodning (ting vi gør) en computational literacy-aktivitet i et softwareprogram (ikke-fysisk ting vi har), der afvikles på en computer (fysisk ting vi har).” (Hachmann, 2023, s.83).

Hachmann uddyber literacy-begrebet med fokus på menneskelige forudsætninger, materialiteter og den sociale kontekst, og computational literacy som færdigheder og dispositioner, der bringes i spil i en teknologimedieret og situeret kontekst med det sigte at opnå et værdsat mål. Derfor bør undervisning i computational literacy ifølge Hachmann bygge på perspektiver, der indlejrer diSessas tre aspekter af computational literacy (Hachmann, 2023).

Dohn vægter også den konkrete kontekst – det situerede – højt i forbindelse med at definere computational literacy. Hun eksemplificerer med læsning, hvor akademiske tekster generelt anses som mere literate, hvorimod slangjargon i en konkret kontekst kan være mere literate (Dohn, 2021). Ligeledes forholder det sig med programmering ved hjælp af forskellige kodesprog. Dohn foreslår alligevel et kontinuum for computational literacy, der bygger videre på ideen om, at hvert niveau forudsætter de tidligere og hvor analog computational tankegang (problemløsning) er det mest basale niveau (Dohn, 2021).

Tool-rich cultures og badges of membership

DiSessa fremhæver, at alle fagkulturer, der har opnået en høj udtrykskraft, er ”tool-rich cultures”, idet de har udviklet mange ”ting at tænke med”. Den afgørende pointe for diSessa (2000) er, at disse ”tools” (teknologier) er udviklet til at understøtte kulturens faglige udtryksevne og aldrig udgør et mål i sig selv. Det er også derfor, at klassiske læremidler ifølge diSessa kan opleves som kunstige og skoleagtige for elever, idet teknologierne ikke er udviklet som del af børnenes egen gryende fagkultur, men derimod introduceret udefra. Programmering, herimod, giver eleverne et værktøj til selv at udtrykke nye ideer og tilmed selv at udvikle egne programmer og værktøjer til faglige formål, fx et program til at tegne cirkler med en virtuel skildpadde.

“These tools are in a deep sense owned by their communities. The principles by which the tools work are frequently community property and taught to newcomers. Many are fabricated by and for the community.” (diSessa, 2000, s. 40).

Et andet forhold ved teknologiernes rolle i fagkulturer er, hvordan udvalgte teknologier bliver særligt identitetsgivende for ens sociale tilhørshold til kulturen, og at der er følelser forbundet til disse teknologier. Der kan fx være stolthed forbundet med mestring af en særlig teknologi, men også følelsen af ydmygelse eller at være pinlig berørt, hvis man omvendt er nybegynder og ikke mestrer teknologien. Ifølge diSessa (2000) giver mestring af de særligt identitetsgivende teknologier en elev, studerende eller fagperson et ’badge of membership’ i den faglige kultur.

“Tools are badges of membership, symbols of commitment and accomplishment, frequently tinged with affects such as pride and sometimes (for beginners) embarrassment.” (diSessa, 2000, s. 39).

I artiklens analyse af de studerendes arbejdsprocesser er det således tydeligt, at selvom der er mange mulige tolkninger af, hvilken rolle programmering spiller, så udgør programmeringskompetencer et særligt ’badge of membership’ for de studerende.



Med diSessas brede teoretiske tilgang til at kunne udtrykke sig computationelt samt nyere bearbejdnings heraf fra Hachmann og Dohn analyserer artiklen således de studerendes teknologifaglige arbejdsprocesser, for derigennem at få indblik i hvilken rolle programmeringskompetencer spiller undervejs, og hvordan disse kompetencer kan anskues fra de studerendes perspektiv.

Metode

Artiklen bygger på filmede observationer af de lærerstuderendes teknologifaglige arbejdsprocesser i mindre grupper i to undervisningsmoduler, samt en eksamen som blev afholdt som workshop for medstuderende og eksterne deltagere. Observationerne blev foretaget på undervisningens første modul og dækker derved de studerendes første møde med fagligheden teknologiforståelse. Det empiriske materiale har et omfang af 5 timer og 30 minutters video og undervejs har der været foretaget uformelle interviews med studerende. Denne artikel fremhæver tre cases, hvor tre grupper af studerende på 1. semester skal løse forskellige designopgaver med teknologi.

På baggrund af vores teoretiske optagethed af de studerendes arbejde med teknologifaglige problemstillinger har vi valgt at trække på visuelle etnografiske metoder, der via kameraet kan synliggøre og stille skarpt på de studerendes handlinger med teknologien. Foruden kameraets naturlige egenskab, hvor det rettes og fokuseres undervejs som en *synliggørelse af forskerinteressen* (Møhl, 2003), er kameraet også led i en fortløbende og allerede begyndende analyse, der følger interaktioner med materialiteter og teknologier. Inspireret af ideen om et *filmet feltarbejde* og *walking with video* (Rehder, 2016; Winther, 2013; Pink, 2007), skaber det visuelle materiale et produktivt udgangspunkt for det efterfølgende analytiske arbejde i det kollaborative forskningsarbejde. Det visuelle materiale gør det muligt for forskergruppen at gense og diskutere detaljerede observationer, som har været fokuseret på programmering (Pink & Morgan, 2013; Grimshaw & Ravetz, 2009) i fællesskab, og på den baggrund udarbejde de skrevne *tykke beskrivelser* (Geertz, 1994). Det visuelle materiale har alene været tiltænkt intern analyse, og ikke som materiale til en særskilt publiceret film, hvorfor små redigerede sammenklip har fungeret som indledning til undersøgelsen og udformningen af de skrevne cases (Crawford, 1992).

Adgang til felten og de studerende er løbende blevet forhandlet, som det altid er tilfældet i etnografisk arbejde (Hammersley & Atkinson, 1995). Da vi som forskere også er kolleger med underviserne, og ligeledes har en tæt relation til tilblivelsen af undervisningsfaget, har tilliden mellem os og de studerende været et konstant anliggende for vores arbejde. Tilliden har netop været nødvendig for, at vi kunne overvære de studerendes arbejde og få indblik i deres arbejdsprocesser og refleksioner over undervisningen i uformelle samtaler.

Case 1 – Analoge materialer og æstetik

Case 1 udspiller sig to måneder inde i de lærerstuderendes første semester med teknologiforståelse. Aktiviteten er italesat som en "designudfordring" med et tinkering-aspekt, rammesat som aktiviteter der udforsker redskaber og strategier som understøtter legende, iterative eksperimenter med både fysiske og computationelle materialer (Presicce, 2017). Fire studerende som er tilfældigt sat sammen ved aktivitetens begyndelse, skal designe begyndelsen på en "Storm P.-maskine", dvs. en kørebane, som er skabt af forskellige "mikrouniverser" og teknologier. De må i udgangspunkt selv bestemme, om deres start skal være digital eller analog eller en kombination. Gruppen mødes i laboratoriet for at vælge en digital teknologi, men efter lidt snak frem og tilbage i håbet om at de kan finde en fælles teknologi at gå videre med, splitter gruppen op i to; to studerende går videre med at programmere en LEGO-spike robot, mens gruppen vi følger, vil arbejde med et "blåt tema", en verden eller en hav-fortælling, hvor en kugle skal bevæge sig gennem en Flexi trax bane. Den "tekniske del" af banen består i, at en snor med en bold for enden bindes til loftet og med den præcise afmålte afstand svinges, så den rammer en anden kugle, der sætter banen



i gang. De to studerende henter masser af forskellige materialer og afprøver hvad piberenser, standard-plastic forme og tape kan bruges til, mens de hele tiden diskuterer og omgør deres beslutninger. De får ideer sammen ved at tage fat, røre, sætte sammen, samle, skille ad, lave om og designe gennem alle de forskellige materialer som har ret forskellige teksturer (hårdt, blødt, sart mv).

Rie	"Jeg glæder mig til at bygge det, træer, pap og sådan noget."
Dorte	"Ja, at gøre det æstetisk pænt."
Rie	"Hvad skal vi lave efter?"
Underviser	"En Storm P-maskine."
Rie	"En Storm P maskine?"
Underviser	"Du ved sådan en maskine, du ved sådan en skinne, så sker der noget og så sker der noget andet..."
Dorte	"Jeg har lige set sådan en i et madprogram jeg ser, i Bagedysten! Og vi skal bygge den! Det er lang tid vi har til det, og det er kun sjovt, hvis det lykkes, det er ikke sjovt hvis den stopper undervejs."
Underviser:	"Så I har starten? Et eller andet der sætter det i gang, I må godt være... I må jo godt kode, så man fx klapper, og så går det i gang?"
Rie	"Vi skal lave en tinkering, ha, vi er de første, vi skal finde ud af det."
Dorte	"Jeg ved ikke, hvad jeg har lavet. Er det for grimt? Ja, det er for grimt, vi tilføjer bare noget for at gøre det."
Rie	"Den skal jo sættes i gang, skal vi ikke have noget heroppe, det kan vi godt."
Dorte	"Jeg er glad for alt det vi tog."
Rie:	"Ja."

Det er to veloplagte og materiale-jagtende studerende, der går i gang med at skabe "et blåt univers". De bruger tid på at planlægge deres design, kontekstualisere deres materiale-ideer og koble dem til kravet om, at det er deres bane-element, der starter den fælles Storm P.-maskine. Et spændingspunkt er, at de selv skal bygge bane-elementet, og at de har en stærk opfattelse af, at den fælles bevægelse eller rytme gennem maskinen "kun er sjov, hvis det lykkes, det er ikke sjovt hvis den stopper undervejs". De arbejder med materialer, som de kender til - "jeg elsker Flexi trax" - og de vil gerne bygge og "gøre det pænt", siger de.

Men kombinationen af deres forståelse af, at maskinen ikke må gå i stå, og underviseren, der siger, at de "godt må kode" udløser en usikkerhed mellem de to studerende om, hvorvidt deres faglige projekt nu også er stærkt nok. Da der er fem minutter tilbage af aktiviteten, inden alle de mange gruppers baner skal sættes sammen, går der en form for "maker-panik" igennem gruppen. De leder hektisk i materialekasserne og bygger frem og tilbage på banen, mens de samtidig forsøger at få den hvide bold til at ramme en sort bold, der skal rulle ned gennem banen. De bliver i tvivl om kvaliteten af det æstetiske udtryk.



Det som før var at “gøre det æstetisk pænt” veksles til, at de synes, de har lavet et “grimt” udtryk, hvor de bare “tilføjer noget for at gøre det”. Som en slags materiale-hævn over banen legitimerer de deres valg af materialer. Selvom de fra designprocessens begyndelse har glædet sig til at arbejde med Flexi trax og “at pynte” og “jeg er glad for alt det vi tog”, er der en lidt tung stemning. De ser deres medstuderendes færdige arbejde med deres del af maskinen og kan se, at nogle af de andre grupper er lykkedes med at programmere små robotter beregnet til undervisning (Ozobot, Dash og LegoWeDo).

Den svære start

De to studerende, der endnu ikke kan programmere, giver til sidst udtryk for, at de gerne vil kunne det sprog, den literacy, som de kan se nogle af deres medstuderende mestrer. De har arbejdet med analoge materialer, men opdager til sidst, at de er udenfor et mere prestigefyldt fællessprog.

Da de studerende kun er i begyndelsen af første modul, har de endnu ikke lært at programmere, hvilket stemmer overens med, hvad der forventes af dem. Derfor er de to studerendes følelse af skuffelse interessant, fordi det viser et situeret kompetence-hierarki i det tidlige møde med teknologiforståelsesfaget, der kan opstå omkring programmering i undervisningssituationer.

De studerende i denne case kunne f.eks. godt have forfulgt eller stået fast på en tinkering-tilgang og holdt sig til at designe et analogt fantasifuldt havunivers. De havde med det opstillede faglige projekt slet ikke brug for programmering. Til slut giver de alligevel udtryk for, at de ikke er tilfredse med resultatet af deres arbejde, fordi de opdager, at de er de eneste, der ikke har kastet sig ud i et projekt med inddragelse af programmering.

Det interessante er, at det først er i slutningen af deres arbejde med delen til Storm P.-maskinen, at denne følelse kommer til udtryk. Det er her, de sammenligner deres arbejde med deres medstuderendes arbejde og særligt får øjnene op for, at deres design mangler noget centralt i forhold til det faglige fællesskab, i og med de andre studerende har forsøgt sig med programmerbare elementer. Dette forhold stemmer overens med diSessas beskrivelse af *badges of membership* som særligt identitetsgivende for ens sociale tilhørsforhold ved mestring af de særligt identitetsgivende teknologier.

Case 2 – Motivation for programmering

De studerende skal i en tinkering-workshop bygge en “shitty-robot”, som de skal præsentere for resten af deres hold sidst på dagen. En shitty robot er en robot, der løser et problem, som ingen har og som løser problemet på en dårlig og sjov måde. De studerendes shitty-robot i denne undervisningsaktivitet skal bl.a. indeholde programmering og et digitalt element.

Som start på arbejdet samler gruppen forskellige teknologier, som de hver især påtænker at bruge til deres fælles robot. Der er en høj grad af åbenhed i opstarten af ideskabelsen. De tager udgangspunkt i nogle af de teknologier, som de har fundet. De samler et par barbermaskiner, en gammel filmrulle og en gammel lampe. En får ideen til, at lampen kan dreje rundt, mens en anden vil montere en større gammel filmrulle ovenpå lampen, så barbermaskinerne kan sættes på denne. Endelig får en tredje ideen om at skabe en gruppebarbermaskine, der skal gøre op med, at barbering er privat og intimt, men i stedet noget venner kan gøre sammen med en gruppebarbermaskine.

Allerede tidligt i deres arbejde med at bygge robotten vil de gerne kunne kontrollere hastigheden på den motor, der skal bruges til at dreje barbermaskinerne rundt på lampen. De har således et behov for at styre motoren og ønsker at programmere noget til at styre motoren, som samtidig kan være det digitale element i deres robot.



- Bjørn "Prøv at høre her, det der med at kunne programmere noget ik? Det skal vi jo bruge hele weekenden på at lære..."
- Underviser 1 "Ja ja så nu må det jo bliver sådan... altså, som det nu bliver... Der er jo heller ikke et krav om programmering, bare om et digitalt element."
- Bjørn "Jamen det er jo det, der er problemet. Jeg kender ikke det her kit, det vil jo tage mig... at sætte mig ind i hvordan man programmere noget som helst..."
- Underviser 2 "Hvad er det du skal bruge det til?"
- Bjørn "At skrue op og ned for strøm (...) Du skal bare passe på med de der små kredsløb, hvis du trækker for meget strøm ud af dem."
- Karl "Hvis vi trækker slideren til USB, og USB til den der motor, så kan vi jo selv styre det..."
- Bjørn "Jamen hvad gør LittleBit i den der sammenhæng?"
- Karl "LittleBit'en gør jeg ikke noget ved. Det eneste jeg gør at trække en ledning fra LittleBit ud til en motor, så vi kan slide den. Det er helt fint, der kommer ikke noget kredsløb."
- Underviser 2 "Ja der er en slider til LittleBits, og en USB-port men... "
- Bjørn "Jamen der er jo ikke strøm på den slider... slideren er jo bare en modstand. Hvad har det med LittleBit at gøre? Jeg forstår det ikke."

Gennem de næste fem minutter viser en af de studerende en af de løsninger, han havde tænkt. Den opgiver de. Dernæst præsenterer en underviser en løsning, hun har tænkt på, men den må også forkastes. Efter dette bryder gruppen ud i tre grupper. I grupperne undersøger de tre mulige teknologier, som måske kan tilføre deres robot det programmerbare element, som de håber på. Teknologierne de forsøger sig med, er Lego Spike, Micro:Bit og MakeyMakey.

- Bjørn "Selve det mekaniske, det har vi styr på, det har vi bare ikke samlet. Men det der er udfordringen..."
- Underviser2 "... det er at få den der til at køre. Altså den der med USB... "
- Bjørn "Næ, nej nej det er at tilføre en eller anden form for digital styring af den."
- Underviser2 "Nåå på den måde..."
- Bjørn "Ja for det her er jo bare at montere den her, sætte vores hjul på, montere barbermaskinen og så sætte sig i en gruppe. Fordi det er gruppebarbering."
- Underviser2 "[griner]" .



- Bjørn "Og så kan vi sidde her med vores skæg [læner sig fremad mod robotten med hagen først] rundt om den."
- Underviser2 "[griner] Det er fanne hyggeligt. Den skal virke!"
- Bjørn "Men vi har bare svært ved at få det digitale element ind i den."

Hele gruppen og de to undervisere står nu sammen om motoren og diskuterer mulighederne. De har ikke fundet en programmerbar løsning til at styre hastigheden på den motor, som drejer barbermaskinerne rundt, hvilket har været et fokus i 30 minutter. De arbejder derfor videre på at få den mekaniske del af robotten til at virke og dermed få deres gruppebarberingsmaskine klar til præsentationen for de andre på holdet.

Ved præsentationen demonstreres deres robot, og en fra gruppen får klippet skægget ved præsentationen til stor begejstring for holdet. De præsenterer det som et nederlag, at de "ikke har arbejdet med kodning og bits", men forklarer de mange forsøg de har gjort, samt hvordan de endte med at finde et batteri med lav nok strømstyrke til, at motoren kørte den passende hastighed.

Materiel intelligens og faglig anerkendelse

I case 2 går de studerende på jagt efter noget, som kan bruges til deres robot, uden at de på forhånd har diskuteret, hvilken robot de vil bygge, eller hvad de vil lede efter. Det betyder, at de får, hvad diSessa beskriver som "ideas with the medium" (diSessa, 2000, s. 116), altså ideer som opstår sammen med materialerne og ikke nogen, som udtrykkes med materialerne. diSessa beskriver, at *material intelligence* opstår i samskabelse med eksterne materialiteter, som i hans kontekst betyder computerhardware (diSessa, 2000).

I gruppens arbejde med robotten er det tydeligt, at de tre aspekter af computationel literacy fra diSessa, det kognitive, det relationelle og det sociale, alle spiller en afgørende rolle i de studerendes læreproces, idet de alle bidrager med materialer, som indgår i robotten og via disse materialer får en indgangsbillet til den kollaborative, sociale og mentale konstruktion af den fælles robot. Deltagelse i arbejdsprocessen bliver således konkret materialiseret gennem teknologiske bidrag til den fælles proces.

Ud over at de arbejder med analoge materialer vil gruppen løse opgaven og kravet om et programmerbart og et digitalt element. Underviseren påpeger, at trods opgaveformuleringen vil det også være i orden, hvis der ikke indgår programmering men blot et digitalt element. De studerende vil dog fortsat gerne programmere en styringsmekanisme til motoren. Dette er ikke den letteste løsning rent teknisk og kræver tilmed, at de studerende kaster sig ud i at lære noget nyt, som de endnu ikke mestrer. Udover at dette kan ses i lyset af, at de studerende forsøger at løse opgaven, ser vi igen, at visse teknologier kan udgøre *badges of membership*, der er særligt værdisat i fagkulturen.

Modsat case 1, hvor de studerende i slutningen af processen opdager, hvad programmering kan tilføre i forhold til et udtryk, har de studerende i denne case fra begyndelsen af processen et tydeligt billede på, hvad programmering vil kunne tilføre deres robot.

Case 3 – Musikprogrammering på micro:bit

Denne case adskiller sig fra de to forrige, idet den tager udgangspunkt i de studerendes modulprøve. Den er rammesat som en konferencedag, hvor de studerende skal afvikle en halv times workshop for eksterne deltagere inklusive deres medstuderende. Den didaktiske rammesætning er altså udarbejdet af de studerende senere på semestret end de to første cases, men blikket er også her rettet mod



arbejdsprocesserne. Vi følger en gruppe deltagere i denne workshop bestående af to personer, en studerende og en ekstern deltager.

Den ene workshop-underviser rammesætter undervisningsaktiviteten og placerer en computer og et micro:bit toolkit på deltagernes bord. En micro:bit er en mini computer, som kan modtage input via en række sensorer, USB-port og Bluetooth. Deltagerne får et stykke papir med noder fra et kendt klassisk musikstykke og et flot designet A3-ark med titlen på workshoppen. Workshop-underviseren fortæller dem, at de på arket skal sætte klistermærker i felterne "dekomposition", "abstraktion", "algoritmer" og "mønstergenkendelse", når de anvender nogle af disse CT-kompetencer i arbejdsprocessen med at programmere en micro:bit til at spille musikken fra nodearket.

De to deltagere begynder at pakke deres micro:bit ud af æsken. Den ene workshop-underviser forklarer, at de til at starte med skal have en ledning til at forbinde computer med micro:bit og giver dem et USB-stik. Den studerende i gruppen udtrykker, at de gerne vil have lov til at lege lidt med det selv, da de ikke har erfaringer med at programmere micro:bits.

Deltagerne bruger meget af tiden på at undersøge, hvordan de får forbundet deres micro:bit til en computer. De tager fat i forskellige kabler og hardware. Workshop-underviseren henvender sig et par gange for at sige, at de gerne må spørge om hjælp.

Han lægger et keyboard til dem, som de kan forbinde til deres micro:bit, og senere lægger en anden workshop-underviser en manual til dette keyboard frem foran dem.

Efter at deltagere i noget tid uden succes har forsøgt at forbinde keyboard, micro:bit og computer, bryder workshop-underviseren ind og hjælper dem. Han siger, at de har forbundet micro:bit rigtigt til computeren, men at de skal droppe keyboardet og i stedet koncentrere sig om at forbinde en højtaler til deres micro:bit, så der kan komme lyd ud.

Han finder en højtaler til dem, men lader dem selv sætte højtaleren til micro:bit. Han forklarer, at de skal bruge en af de to knapper, A eller B, på deres micro:bit, når de skal have den til at spille. Efterfølgende taler han dem gennem processen med at downloade en kode fra programmet MakeCode til micro:bit.

Mia	"Så vores kode laver vi herinde og sender til micro:bitten."
Workshop-underviser	"Yes."
Søren	"Okay."

Kodefilen lægges over på deres micro:bit og man ser den procentvis downloades. Mens den downloades, reflekterer den ene deltager, der er studerende på faget, højt over læreprocessen, mens hun sidder med klistermærkearket i hænderne. Hun tilføjer, at det ikke er første gang i løbet af semestret, at de arbejder med CT-kompetencebegrebet mønstergenkendelse.

Mia	"Der var noget der gik op for dig. En mønstergenkendelse. [deltageren sætter klistermærket på bingopladen] Den tror jeg altså vi har haft nogle stykker af."
Workshop-underviser	"Hvis I ser på den [peger på micro:bitten], der står A og B. Hvad var det for en kode I havde skrevet til? "
Mia	"A, ja men det er vi med på [hun tager micro:bitten i hænderne] nu er det noget med... "
Søren	"Nu skal vi bare trykke på den, så skulle den gerne gøre noget".



- Mia "[trykker på A] Der sker så ikke en skid... skal vi lige prøve [hun tager fat om ledningen til højttaleren] må man godt tage den?"
- Workshop-underviser "Mm."
- Mia "Sætte den herover bare for at se hvad der sker. [rykker USB kablet og trykker på A og musikken spiller]."
- Søren "Okay."
- Mia "OKAY [glad og henrykt]."

Deltagerne får kodet deres musikstykke færdigt og får downloadet det til micro:bit. Til sidst skal alle høre de to gruppers melodier og den ene deltager trykker på A på micro:bit, som spiller deres musikstykke. Der bliver klappet, grinet og hujet.

Mozart – lyden af succes

I denne case om programmering og micro:bit lykkedes den studerende og hendes gruppemedlem med at anvende programmering i deres problemløsning. De får ideer i deres arbejde med materialer (materiel intelligens) ligesom i case 2, her forstået både som materialer og sprog. Det er en central pointe, at det netop er gennem workshop-underviserens (medstuderende) understøttende facilitering gennem fysiske materialer og gennem sproget, at deltagerne får ideer og til sidst lykkes.

Gruppen bliver guidet til at reflektere med CT-begreber over deres arbejde med programmering. Gennem materialer som papir og klistermærker samt påmindelser fra workshop-underviserne, sætter deltagerne klistermærker med CT-begreber som mønstergenkendelse, abstraktion, dekomposition og algoritmer på refleksionsarket. Det er især deltageren, som er studerende på faget, der viser, at de anvender de computationelle begreber til at løse deres problem, idet hun reflekterer verbalt over deres praksis, og bliver samtidig bevidst om egen mestring af denne faglighed.

I disse refleksioner genkendes det, som diSessa betegner som det kognitive aspekt af computational literacy. Dette kognitive aspekt kommer her til udtryk i tæt samspil med både det sociale og det relationelle aspekt af computational literacy, idet disse verbale refleksioner er stramt styret af CT-begreber på klistermærkerne i deres samarbejde om at kode musik. Da Mozart til slut toner fra deres micro:bit, er det lyden af succes og opnåelsen af *badges of membership* i forhold til at have færdigheder til at skabe et computationelt artefakt, der har aspekter af noget udtryksfuldt. På denne måde viser denne case også, hvordan de studerende, der faciliterer workshoppen lykkedes med at skabe det tiltænkte symfoniske samspil mellem kompetenceområder i fagligheder, idet det ikke er programmering i sig selv, der på noget tidspunkt har været målet i denne workshop, men formålet har været med støtte og vejledning at vise deltagerne i workshoppen, at programmering kan åbne nye muligheder for at skabe udtryk, og handler om at lære at tænke på nye måder.

Konklusion

I teknologiforståelse som undervisningsfag på læreruddannelsen skal de studerende opnå faglige kompetencer gennem arbejdet med områder som digital myndiggørelse, designfaglighed, teknologisk handleevne og computationel tankegang, som de fleste ikke har forudsætninger for, når de starter med faget. I de tre cases vises, hvordan de studerende på forskellige måder forstår programmering som en særligt attraktiv, men også vanskelig kompetence i deres første møde med den nye faglighed.



I case 1 arbejder de studerende med computationel tankegang som tænkeredskab eller metode til problemløsning af designopgaven. De studerende har delt sig op i to grupper ud fra, hvem der kan programmere, og hvem der ikke kan. Casen følger dem, der ikke kan, og det viser sig, at de til slut i processen giver udtryk for, at deres begyndelse på Storm P.-maskinen ikke har samme niveau som deres medstuderende, selvom der ikke på noget tidspunkt foretages en summativ evaluering. De får via deres medstuderendes arbejde med programmering af teknologier til den fælles Storm P.-maskine en oplevelse af, at det at kunne programmere er af høj værdisætning i fagligheden, da det via programmering er muligt at skabe nye udtryk og produkter.

I case 2 insisterer de studerende på at inkludere programmering i deres opgaveløsning. Med en tinkering-tilgang leger de sig frem til ideer om, hvad deres robot skal kunne. De tester, afprøver og finder hurtigt ud af, at de ønsker, at deres robot skal kunne dreje rundt i en vis hastighed, som skal programmeres. Men de mangler færdighederne til at skabe dette udtryk, hvilket skaber frustration i gruppen. De studerende har fra begyndelsen af arbejdet med robotten kompetencer i forhold til at kunne forestille sig og tænke muligheder med programmering, men denne case viser, at det at kunne tænke sig frem til en mulig løsning ikke er det samme som at kunne udføre den.

I case 3 arbejder den studerende og hendes gruppemedlem med computationel tankegang i form af problemløsning og programmering af micro:bit. Deres arbejde skal de løbende reflektere over og kategorisere inden for de forskellige CT-elementer. Læreprocessen omhandler altså både computationel tankegang og computational literacy. Det lykkedes dem at programmere en micro:bit til at spille Mozart til stor begejstring for alle herunder deres workshop-undervisere (medstuderende), som undervejs har ydet støtte og hjælp i forhold til, at de kunne nå i mål. Programmering viser sig tydeligt her som midlet til at skabe et nyt udtryk.

Gennem de tre cases vises det, hvilken betydning computational literacy har for de studerende, og hvordan programmering i nogle tilfælde bliver et badge of membership. Programmeringsfærdigheder bør ifølge studieordningen ikke være og er ikke særligt prioriteret i undervisningen på første semester, hvilket betyder, at der er en forskel i færdigheder mellem de få, som allerede har dem eller de som af egen vilje indbygger programmeringsfærdigheder i deres didaktiske opgaver som i case 3. De studerendes forudsætninger bliver således afgørende i forhold til de mange muligheder for at koble programmering og de øvrige kompetenceområder, hvorved programmering enten kan komme til at stå meget centralt eller mere på linje med andre færdigheder og tilgange i faget. Faget er nyt og derfor er det helt naturligt, at de studerende kommer med forskellige forestillinger om bl.a. programmering, hvilket er med til at definere fagkulturen fra start. Vi har set alle fire kompetenceområder i spil i vores observationer på første semester, men det tyder på, at undervisere i teknologiforståelse skal navigere i, hvilke teknologier og kompetencer, der på godt og ondt bliver særligt identitetsgivende og kommer til at udgøre badges of membership.

Denne artikel har undersøgt, hvordan oplevelser af tilhørsforhold, engagement og faglighed kan påvirke praksis i teknologiforståelse som forsøgsfaglighed på læreruddannelsen. Artiklen kan bidrage til at identificere og forstå, hvordan undervisere og uddannelsesinstitutioner bedst kan støtte og udnytte disse oplevelser af tilhørsforhold og engagement i en sådan tilblivende faglig praksis. På baggrund af disse analyser, som undersøger de studerendes teknologiforståelsesfaglige arbejdsprocesser, opstår et behov for yderligere forskning i hvad der påvirker studerendes motivation, engagement og værdisætninger i faget.

Referencer

Andersen, L. B. (2021). Krydsende teknologiforståelser i teori og praksis: Fra problem til potentiale. *Learning Tech*, 10, 100–126. <https://doi.org/10.7146/lt.v6i10.125621>



- Bocconi, S., Chiocciariello, A., & Earp, J. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education. Report prepared for the Nordic@ BETT2018 Steering Group. <https://doi.org/10.17471/54007>
- Bocconi, S., Chiocciariello, A., Kampylis, P., Dagiene, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Jasute, E., & Malagoli, C. (2022). Reviewing computational thinking in compulsory education: state of play and practices from computing education (JRC128347). Publications Office of the European Union. <https://data.europa.eu/doi/10.2760/126955>
- Caeli, E. N., & Bundsgaard, J. (2019). Datalogisk tænkning og teknologiforståelse i folkeskolen tur-retur. *Tidsskriftet Læring og Medier (LOM)*, 11(19), 30. <https://doi.org/10.7146/lom.v11i19.110919>
- Crawford, P. I. (1992). Film as discourse: the invention of anthropological realities. I *Film as ethnography* (s.66-82). Manchester University Press.
- Danholt, P. (2021). Technology understanding in a more-than-human world. *Learning Tech*, 10, 169–190. <https://doi.org/10.7146/lt.v6i10.125722>
- diSessa, Andrea (2000): *Changing Minds. Computer, Learning and Literacy*. Cambridge Massachusetts: MIT Press.
- Dohn, N. B. (2021). Computational Thinking—Indplacering i et landskab af it-begreber. I Dohn, N. B., Mitchell, R., Chongtay, R., & Dohn, N. B. (Red.). *Computational thinking: Teoretiske, empiriske og didaktiske perspektiver* (s. 31–58). Samfundslitteratur.
- Geertz, C. (1994). Thick description: Toward an interpretive theory of culture. *Readings in the philosophy of social science*, 213-231. Basic.
- Grimshaw, A., & Ravetz, A. (2009). *Observational Cinema - Anthropology, Film, and the Exploration of Social Life*. Indiana University Press.
- Hachmann, R. (2023). Computational Literacy. Kognitive, sociale og materielle aspekter ved teknologiforståelser i skolen. *Learning Tech – Tidsskrift for læreruddannelse, didaktik og teknologi*, (13), 78-99. DOI: [10.7146/lt.v8i13.132969](https://doi.org/10.7146/lt.v8i13.132969)
- Hammersley, M., & Atkinson, P. (1995). *Ethnography - Principles in Practice* (Second Edition). Routledge.
- Møhl, P. (2003). Synliggørelsen: med kameraet i felten. In K. Hastrup (Ed.), *Ind i verden: En grundbog i antropologisk metode* (pp. 162-183). Hans Reitzels Forlag.
- Nørgård, R. T. (2020). Teknologifantasi. *Tidsskrift for læreruddannelse og skole*, 40(117), 65–79.
- Papert, S. (1980). *Mindstorms; Children, Computers and Powerful Ideas*. New York: Basic Book.
- Pink, S. (2007). Walking with video. *Visual Studies*, 22(3), 240-252.
- Pink, S., & Morgan, J. (2013). Short-Term Ethnography: Intense Routes to Knowing. *Symbolic Interaction*, 36(3), 351-361. doi: [10.1002/symb.66](https://doi.org/10.1002/symb.66)
- Presicce, C. (2017). *Explorations in computational tinkering*. Thesis: MIT, School of Architecture and Planning. URI:<http://hdl.handle.net/1721.1/114066>
- Rambøll. (2021). Slutevaluering: Forsøg med teknologiforståelse i folkeskolens obligatoriske undervisning (s. 99). Børne og Undervisningsministeriet. <https://www.uvm.dk/aktuelt/nyheder/uvm/2021/okt/211004-erfaringerne-fra-forsoeg-med-teknologiforstaelse-i-folkeskolen-er-landet>
- Rehder, M. M. (2016). *Søskendenævær: et fænomenologisk inspireret studie af unge adskilte søskendes hverdag med afsæt i teknologier, materialiteter og kropslige erfaringer*: ph.d.-afhandling. DPU, Aarhus Universitet.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2011). Research notebook: Computational Thinking – what and why? *The Link Magazine*, 6.
- Winther, I. W. (2013). Forskerens fornemmelse for øen: en kropsligt indvævet forskningspraksis. In K. Rasmussen (Ed.), *Visuelle tilgange og metoder i tværfaglige pædagogiske studier: en antologi baseret på erfaringer og indblik fra forskning, udviklingsarbejde og undervisning* (1. ed., pp. 309-336). Roskilde Universitetsforlag.



Forfattere

Thilde Emilie Møller

Lektor
Professionshøjskolen Absalon



Mads Middelboe Rehder

Lektor
Københavns Professionshøjskole



Marie Falkesgaard Slot

Docent
Københavns Professionshøjskole



Lars Bo Andersen

Docent
Københavns Professionshøjskole

