

Bruk av datamaskiner og digitale redskaper

I forrige uke snakket vi om bruk som en relasjon mellom menneske og artefakt, i en situasjon, i en kontekst. Vi var opptatt av artefakten som redskap, som forlengelse av menneskets evner og handlingsrom. Vi skilte mellom hvor spesialisert redskaper er ved å bruke ulike begreper: verktøy og instrument. Vi kan snakke om digitale redskaper på samme måte, men det er en viktig forskjell mellom tradisjonelle redskaper, som en hammer, og redskaper som er digitale: de er ikke passive objekter, de gjør ting. Digitale redskaper er datamaskiner, og som andre maskiner kan de utføre oppgaver (hvis de tilføres energi). I notatet sitt skriver Svein om de seks typene enkle maskiner, som et utgangspunkt for å forstå mange av dagens innretninger. Vi kan gjøre noe av det samme for datamaskiner: prøve å forstå hva som er så spesielt med dem ved å ta utgangspunkt i det som kjennetegner det digitale.

For å snakke om hva som kjennetegner datamaskiner som maskiner, kan vi se på hvilke egenskaper de har som kan «forlenge» menneskelige egenskaper og hva som kjennetegner dem rent teknisk. Vi kan si at datamaskiner egner seg til å registrere, lagre, bearbeide og gjenfinne data. De kan huske, søke, strukturere, lete og behandle store mengder data mye mer effektivt enn mennesker, og egner seg til rutineoppgaver og oppgaver der store mengder data skal bearbeides på kort tid.

I utgangspunktet er datamaskinen en maskin som behandler data «input» om til data «output». Vi bruker ofte betegnelsen informasjonsteknologi (IT) fordi dataene representerer informasjon som menneskene er interessert i å bruke. Dataene må kunne registreres digitalt og ha en digital representasjon, og de behandles etter bestemte regler og metoder eller algoritmer. Behandlingen produserer «output», som også representeres digitalt. En kalkulator kan være et eksempel: input er tallene som skal regnes på og den regneoperasjonen som skal utføres, output er resultatet av regneoperasjonen.

De grunnleggende egenskapene til datamaskiner muliggjør og begrenser hvilke oppgaver datamaskiner er egnet til å utføre. Datamaskiner utfører først og fremst «kognitive» oppgaver, dvs. oppgaver som består i å behandle abstrakte, symbolske digitale representasjoner, og som kan kontrollere andre fysiske eller abstrakte operasjoner. Et enkelt eksempel er en termostat som måler temperaturen og vil slå av ovnen hvis temperaturen er høyere enn en gitt verdi og slå den på dersom temperaturen er lavere enn denne verdien. Termostaten kontrollerer ovnen. Roboter er datamaskiner som kan også utføre fysiske oppgaver ved å bevege seg (f.eks. robotstøvsuger) eller ha bevegelige deler (f.eks. bevegelig arm).

Nedenfor tar jeg for meg noen av egenskapene ved datamaskiner og «det digitale» som påvirker både design og bruk av digitale redskaper. Først må jeg si litt om «det digitale». Så tar jeg utgangspunkt i at datamaskiner kan karakteriseres ved sin input, prosessering, og output, og diskuterer hva de består i: data og algoritmer.

Det digitale

Utgangspunktet for digitalisering er forskjeller i volt (spenning) i strøm, der høy spenning defineres som 1 og lav spenning defineres som 0. Digitale signaler tolkes altså som logisk 1 eller logisk 0, noe som er en abstraksjon siden strøm er kontinuerlig og varierende. Den binære representasjonen: 1 eller 0, innfører altså et skille mellom høy og lav i det som egentlig er en kontinuerlig strøm. Dette er gjennomgående i digitale representasjoner: de er enten-eller dvs. 1 eller 0, uansett om det fenomenet de representerer har flytende overganger.

De logiske 1 og 0 inngår i de logiske portene som datamaskiner består av, og som setter sammen 1-ere og 0-er i logiske operasjoner (AND, OR, NAND osv.) i mange abstraksjonsnivåer. En datamaskin er derfor en abstrakt maskin, bygget opp med mange lag av abstraksjon fra strømspenning til det som møter brukerne og som de kan interagere med. Et eksempel er kommandoen *print*, som gjør at et dokument sendes til en printer og skrives ut uten at vi trenger å bekymre oss om hvordan alle delene er koblet sammen og kan interagere.

Kommandoen *print* er altså en abstraksjon og representerer et sett med moduler som gjør at datasystemet printer. Abstraksjon gjør det mulig å dele opp komplekse systemer i moduler, som kan studeres / designes hver for seg og der vi kan se bort fra kompleksiteten i andre moduler.

Abstraksjon er helt grunnleggende for digitale systemer / datamaskiner. Når vi abstraherer trekker vi fram noen karakteristika som viktige og trekker fra eller ser bort fra andre. Det gjelder også for dataene, som er digitale abstraksjoner. Et enkelt eksempel på en digital artefakt er et termometer, der vi måler feber som et tall. Tallet representerer hvor mye feber du har og anses ofte som «riktigere» enn hvor febril du føler deg.

Representasjoner

Data er digitale representasjoner som kan representere all slags informasjon digitalt. Datamaskinens representasjoner er digitale, dvs. binære og ikke kontinuerlige eller analoge, og de er egnet til å representere det som er mulig å måle og telle. Termostaten måler temperaturen og slår på ovnen hvis den er under et bestemt tall.

En sentral aktivitet for informatikere er å kategorisere data og definere datatypene som skal behandles og beregnes i datamaskinen. Dataene og behandlingen av dem kan ses som en modell av et fenomen, der vi tar med de datatypene som er viktige og bestemmer hvordan de skal behandles. Jeg kan bruke Majorstuen-krysset som eksempel. Vi tar med biler og trikker og busser, og samler dem i kategorien kjøretøy. Vi må kanskje dele biler opp i ambulanser og personbiler fordi det gjelder ulike regler for disse to typene biler. Og vi må bestemme hvor Majorstuen starter: kanskje er det ved lyskryssene på hver side av Majorstu-huset? Eller skal vi la grensen gå der køen om morgenen starter, som er langt forbi lyskryssene. Hva som er med og ikke, og hva som velges ut som viktig, kan ha betydning. Tar vi ikke med sykler i modellen, kan det ha konsekvenser for hvordan veibanen blir designet, gir vi trikken mer vekt enn andre kjøretøy, vil det ha konsekvenser for hvordan trafikken vil flyte i krysset. Et annet hverdagslig eksempel kan være skirenn for barn og ungdom, der fødselsår er det som brukes

for å definere deltaker-klasser, mens vi vet at i ungdommen varierer prestasjonene mer på grunn av høyde og vekt enn av fødselsår. Hva som velges og prioriteres har ofte konkrete konsekvenser for folk.

Modeller er representasjoner av fenomener der noen ting er med og andre ikke. Hva som inngår i modellene er viktig i og med at ikke alt kan representeres, og det gjelder alle slags kategoriseringer og klassifiseringer. Hvis klassifiseringene og kategoriseringene er knyttet til konkrete effekter vil de kunne ha store konsekvenser for folk, f.eks. vil det å ha en medisinsk diagnose være viktig for dine rettigheter som pasient og borger, f.eks. rett til sykepenge. Har du ingen diagnose, får du ikke sykepenge. Ofte inneholder klassifikasjonssystemer en restkategori, og denne kan endre seg over tid. Eksempler på dette er sykdommen AIDS, som først ble en diagnose etter mange år, og dagens diskusjon om ME er en diagnose.

Abstrakte representasjoner kan representere alle slags data. Abstraksjoner representerer konkrete objekter, strukturer og prosesser som fins i verden, og når abstraksjonene konkretiseres i representasjoner så blir de også en del av verden. I sykehus finnes f.eks. mennesker representert som pasienter både som innlagte personer og som pasientjournaler. Som pasient er det diagnose og symptomer som er viktige, og ikke så mye annet. Mange av dataene som samles inn om pasienten er bare indikatorer på sykdom eller behandling: indikatorer er det som kan måles og registreres som tegn på noe annet. Febermåling tas som indikator på hvor syk du er, blodprøver analyseres for å finne indikatorer på f.eks. betennelse i kroppen. Ofte er det ikke mulig å måle akkurat det man er ute etter, og man må ta til takke med det som er mulig å måle. En forsker fra Transportøkonomisk institutt fortalte en gang at for å beregne tungtransport gjennom Oslo brukte de målinger gjennom bomringen, men de kunne bare måle korte og lange biler, og måtte derfor beregne lastebiltransport basert på det de kunne måle.

Det finnes en rekke sensorer som kan registrere og måle ting vi ikke kan sanse (elektromagnetiske bølger, f.eks. infrarødt lys, radiosignaler). Men også her finner vi at sensorene ikke alltid måler direkte det vi er ute etter. En fuktsensor i en potteplante måler hvor mye strøm som går i kretsen. Strømmen i kretsen endres med fuktigheten i jorda og går lettere gjennom fuktig jord. På forhånd må vi vite hva motstanden er for hhv. fuktig og tørr jord, og forskjellen mellom disse er utgangspunkt for beregningen av fuktighet. Fuktsensoren måler hvor mye strøm som slipper gjennom kretsen, og motstanden i kretsen brukes til å beregne hvor fuktig jorda er. Mange sensorer bruker fysiske og kjemiske egenskaper ved fenomenet som utgangspunkt for å sette opp en strømkrets der en kan måle om og hvordan motstanden endres, og der endringen i motstand brukes som data, som indikator, på det som sensoren skal måle.

Representasjoner i datamaskiner er ofte designet for å være forståelige for mennesker, for å lette interaksjonen og kontrollen av datamaskinen. Internt i datamaskinene er det annerledes, f.eks. har ikke robotgressklipperen noe kunnskap om gress, eller robotstøvsugereren noe kunnskap om rengjøring og støv. Det er menneskene – brukerne – som lager mening ut av det datamaskinen utfører og gir mening til dataene som utgjør input og output.

Prosess

Datamaskiner gjør ting, og i bunn og grunn er de maskiner som lager resultater («output») av de dataene de får inn («input») etter bestemte regler. Datamaskinen er en abstraksjon av denne prosessen. Vi kaller beskrivelsen av hva som skal gjøres med dataene for algoritmer, dvs. strukturen for den prosesseringen datamaskinen gjør. Vi kan si at som andre maskiner er datamaskiner avhengig av at algoritmene er riktige og at de utfører algoritmene riktig, dvs. at de fungerer som de skal, slik at vi kan stole på resultatet. En kalkulator må regne riktig for at vi vil bruke den.

I de fleste datamaskiner bestemmer algoritmene det som skal gjøres med input-data for å produsere den ønskede output. Algoritmene er derfor avhengig av dataene – og omvendt: hvis vi bare kan telle målinger av korte og lange biler, er det disse tallene som må legges til grunn for å finne ut hvor mange lastebiler som kjører gjennom bomringen.

Datamaskiner er ofte designet som generelle maskiner, dvs. at samme maskin kan gjøre mange forskjellige oppgaver, og at man ikke kan se på den hvilke oppgaver den kan gjøre. Dette gjør det vanskeligere å ha samme kontroll som manuelle redskaper og enkle maskiner gir, siden disse er mulig å gjennomskue, forstå og kontrollere. Kalkulatoren kan vi normalt forstå og sjekke, men skatteberegningen er mye mer komplisert og vanskeligere å kontrollere. Store, komplekse systemer, som Skatteetetatens systemer, får data fra mange kilder, og de inngår i flere regnestykker, og både datainnsamlingen og beregningene skjer uten at vi trenger å gjøre noe. Skatteetetatens systemer fungerer som en autonom teknologi, som tar beslutninger og handler på egenhånd, uten eksplisitt kommando fra mennesker.

Bruk av digitale redskaper

Bruken av digitale redskaper kan beskrives på ulike nivåer: vi interagerer med det fysiske grensesnittet, men må også forholde oss til modellene og dataene – som er abstrakte representasjoner. Både den konkrete, fysiske interaksjonen og den abstrakte tolkningen er viktig for bruken av digitale redskaper.

For den fysiske interaksjonen gjelder at grensesnittet er forståelig og tilgjengelig: vi må forstå hva vi skal trykke på eller hva vi skal gjøre, og vi må være i stand til å utføre disse konkrete operasjonene. Interaksjonsdesign handler om å lage interaksjonsmekanismer som er forståelige og brukbare. Universell design handler om hvordan vi skal designe interaksjonsmekanismene slik at de også kan brukes av folk med nedsatt funksjonsevne uten at de må ta i bruk ekstra utstyr. Interaksjonsdesign handler også om at det maskinen responderer med må være forståelig, slik at vi forstår om og at vi har gjort ting riktig. For å snakke om denne konkrete siden ved bruk av digitale redskaper, kan vi bruke innsikt fra bruk av andre konkrete redskaper og maskiner.

Det som skiller digitale redskaper fra andre, fysiske og manuelle redskaper er at god bruk ofte også forutsetter at vi kan tolke og forstå modellene og representasjonene som er implementert. For å forstå kalkulatoren og godta dens resultater, er det nødvendig å kunne regne – et

overslag i hodet vil lett kunne si om resultatet virker riktig. Er det for varmt eller kaldt i rommet, kan vi manipulere termostatens temperaturgrense. For mer kompliserte systemer, med flere typer data og vanskeligere beregninger, kreves mer arbeid for å sjekke at dataene og beregningene er riktige. Vi må forstå modellen som ligger til grunn for beregningene, og vi må vite hvilke data som inngår og hvor de kommer fra, for å kunne kontrollere resultatet.

Dette at vi både må forstå de konkrete interaksjonsmekanismene og de abstrakte modellene gjør at bruk av digitale redskaper er mer krevende enn andre typer redskaper og maskiner. Usynlige eller autonome systemer er derfor veldig vanskelige å forstå og kontrollere. Dessuten medfører de en fornyet diskusjon om begrepene bruk og bruker: kan du sies å være bruker av et system du ikke interagerer med? Skatteetatens systemer har en veldig liten interaksjon med borgerne om skatten, men vi kan likevel sies å være pålagt å bruke Skatteetatens systemer. Hvordan kan vi snakke om bruken av en selv-kjørende bil? Vi kan også diskutere bruk av robotstøvsugere der det er få interaksjonsmekanismer og vi må interagere ved å late som vi er en hindring (sette foten foran for å få dem til å skifte retning).

Oppsummert kan vi si at for å forstå og beskrive bruk av digitale redskaper kan vi hente innsikt fra bruk av manuelle, fysiske redskaper, som fingerferdighet og håndlag. Vi må også ta med kunnskap om bruk av maskiner, dvs. artefakter som utfører oppgaver. Bruken av hverdagslige maskiner som vaskemaskiner kan gi innsikt i arbeidet med å gi input, velge behandling, og motta output. Vi kan erfare hvor viktig tilbakemeldingene fra vaskemaskinen er for bruken. Her kan vi også se hvordan vi forstår hva vi putter inn i maskinen har innflytelse på hvilken behandling vi kan (bør) velge, og at input og behandling må tilpasses hverandre for å gi ønsket resultat. Det tredje og mest kompliserte aspektet ved digitale redskaper er å forstå abstraksjonen som er representert i datamaskinen: hvordan fenomenet som representeres er modellert, hvilke datatyper som er mulige, hvordan de behandles, hvordan datatypene og behandlingen samspiller, og hvordan resultatet skal tolkes. Her kan vi ta med at god forståelse av hva som ikke er med i modellen og hvilke data(typer) som ikke tas med i beregning kan gi verdifull kunnskap og forståelse av effektene av bruken av digitale redskaper / datamaskiner. Å se hva som mangler krever imidlertid mye kunnskap.

Kobler vi dette til diskusjonen om hva slags kunnskaper og ferdigheter vi må ha for å bruke digitale redskaper, kan vi se at vi trenger et stort repertoar av kompetanse. For å operere et grensesnitt trenger vi ferdigheter, det Svein i sitt notat kaller «knowing-how»: vi må vite hvor vi skal trykke og hva vi kan gjøre med redskaper. For å gjøre bruk av resultatet av datamaskinens behandling trengs det kunnskap om fenomenet som er representert i maskinens modeller og data, med vekt på det Svein kaller «knowing-that»: vi må skjønne hva det betyr. Noen ganger kan vi skjønne hvordan en maskin virker ved å analysere erfaringene over tid, slik at knowing-that kan utvikles etter erfaring med knowing-how. Når vi går på kurs for å lære oss et system er det ofte grensesnittet det fokuseres på (knowing-how), men vi får også vite hvordan dataene og beregningene skal tolkes (knowing-that). Vi kan sammenlikne med å kjøre bil med gir: i starten er giringen vrien og krever at vi har oppmerksomheten på selve kjøringen, men etterhvert går giringen automatisk og vi kan bruke mer av oppmerksomheten på kjøringen-i-trafikken og på trafikkbildet.