

Dataprogram som ikke snakker samme språk

Det er matematisk umulig for dataprogram å gjøre feil. Men likevel er datatrøbbel noe vi sliter med hver dag, både i det offentlige og i det private.

Viten Petter Nielsen og Bjørn Erik Mørk Institutt for Informatikk, Universitetet i Oslo

Oppdatert: 12.des. 2013 14:27 Aftenposten Viten

Datamaskiner blir en viktigere og viktigere del av livene våre. De omgir oss omtrent i alt vi gjør, både på jobben og i fritiden, og de har dermed blitt en forutsetning for at vi skal få gjort det vi ønsker. Da er det naturligvis frustrerende når datamaskiner ikke fungerer, eller ikke oppfører seg slik vi forventer. Det skjer jo rett som det er. Men hvorfor opplever vi dette, mer enn 70 år etter vi fikk den første datamaskinen?

Selv om datamaskinene er blitt raskere, mindre og billigere, er vi fortsatt ikke kvitt barnesykdommene.

Hva skjedde da togtrafikken stoppet opp i [hele landet tirsdag 10. desember](#)? Hvorfor stopper T-banene i Oslo regelmessig [på grunn av signalfeil](#)? Hvorfor er vi i en situasjon hvor dataproblemer skal hindre politiet [fra å oppklare forbrytelser](#)? Hvorfor kunne brukere av Altinn logge seg inn [på Kenneth \(36\) sin selvangivelse](#)? Hva skyldes det at ustabile datasystemer førte til 840 uønskede hendelser som potensielt [truet pasientsikkerheten ved Ahus](#)? Hvordan kan veivesenet legge ut [40 000 personnumre på nettet](#)?

Feilfrie dataprogram

Det hender fra tid til annen at maskinvare bryter sammen, men dataprogrammer kan faktisk ikke gjøre feil dersom de er kodet riktig. Det er noe av det flotteste med datamaskiner, om vi ber dem om å gjøre noe to ganger så vil de alltid gjøre nøyaktig det samme. De kan ikke gjøre noe annet, og det er derfor vi kaller dem deterministiske. Det finnes ikke rom for feil og tilfeldigheter. De er faktisk så forutsigbare at de ikke kan lage tilfeldige tall. Og nettopp derfor vi kan bruke datamaskiner som førere av førerløse tog eller som autopiloter i fly. Og vi kan matematisk bevise at programmene vi lager aldri vil gjøre feil.

De første datamaskinene ble programmert av ingeniører og matematikere på universitetene som hadde behov for å utføre store og kompliserte beregninger, for eksempel for å lage grunnlaget for værmeldinger. Disse «brukerne» var eksperter på å dele opp store og kompliserte problemer til mindre og mer håndterbare, og på å definere disse i en strukturert form som kunne legges inn i datamaskinene.

Datamaskinene vi omgir oss med i dag er bygd opp på den samme måten. Selv om de ikke bare brukes til tunge matematiske beregninger er måten vi lager datasystemer i stor grad som før. Vi bryter opp store og kompliserte arbeidsoppgaver og krav til mindre og mer håndterbare deler. Dette har vi over tid laget raffinerte metoder for å gjøre, og vi er blitt flinke til å beregne hvor mye tid det tar å lage de enkelte systemene og de delene de består av.

På mange måter er datasystemene vi lager i dag en kjempesuksess dersom vi ser på størrelse og hvor mye funksjonalitet vi får for pengene sammenlignet med hva vi klarte for 5, 10 eller 20 år siden. Bare tenk på hvordan mobilene nå har en kapasitet som langt overgår det folk flest hadde på sin hjemme-PC for bare få år siden.

Fremskritt & problemer

Men så viser det seg da at vi ikke får det til allikevel. Hvorfor? Er det fordi datasystemene blir større og mer omfattende? Dersom vi nå er så flinke til å dele opp ting, da er det vel ikke noe problem å dele opp noe som er større? Det koster trolig mer, kanskje vi trenger noen flere ingeniører og kanskje det tar litt lengre tid, men det kan vi vel ta høyde for beregning av kostnadene? Dessverre er det ikke så enkelt.

Utfordringen i dag er at datasystemer ikke lengre er frittstående, men koblet sammen med andre. Et sykehus har for eksempel flere hundre forskjellige datasystemer i bruk, mens store sykehus som Oslo universitetssykehus har mer enn 1000.

Et annet eksempel er at de fleste store datasystemer i privat og offentlig sektor er basert på fødselsnummeret for å identifisere personer. På den ene siden muliggjør dette effektiv dataflyt mellom forskjellige instanser og opprydding i databaser når folk flytter eller dør. Men på den andre siden står både offentlig og privat sektor ovenfor en kjempeutfordring og svært store kostnader om fødselsnummeret må endres. Metodene vi har for å lage datasystemer er basert på flere forutsetninger som ikke passer med denne virkelighet. For det første at det alltid finnes en eier og en leverandør av et datasystem som har den hele og fulle kontrollen.

Kamp om kontroll

Tidligere ble systemer typisk utviklet for bruk internt i en bedrift, mens i dag må systemene kobles sammen med eksisterende interne systemer i tillegg til eksterne systemer for å være relevante. For det andre at beslutninger som tas om et system er basert på tekniske og rasjonelle kriterier. I dag utvikler forskjellige aktører med til dels motstridende interesser de forskjellige delene i et system.

Da er det ikke enkelt å avgjøre hvem som skal gjøre hva, hvordan systemene kobles sammen og hvordan ansvar fordeles. Tekniske vurderinger er fortsatt viktige, men mye styres i realiteten av kampen om kontroll og ressurser. For det tredje er det ikke lengre slik at datasystemer skal løse veldefinerte og avgrensede regneoppgaver.

Datasystemer må i dag håndtere problemer som fremstår forskjellig for ulike brukere av systemet, og nye brukere og nye krav kommer til over tid. Sammenkoblingene trenger heller ikke nødvendigvis å være planlagt, men skjer fordi forskjellige systemer behøver hverandre og er tilgjengelige. Det er dette som gjør det mulig å kunne betale med kredittkort i nettbutikken og bruke kodebrikken fra nettbanken og mobiltelefonen for å sjekke ligningen på Altinn.

Veldig ofte kobler vi sammen gamle systemer som vi rett og slett ikke klarer å avvikle med nye systemer slik at vi kan leve med de gamle i noen år til. På den måten finnes det ikke noe overordnet plan som kan forsikre at de forskjellige systemene kobles hensiktsmessig med hverandre. Slike sammenkoblede systemer er ikke lengre deterministiske fordi vi rett og slett

ikke lengre har kontrollen og den fulle oversikten fordi enkeltsystemer utvikles av forskjellige aktører, på forskjellig sted og til forskjellig tid.

Ingen vaksine

Tar vi til oss at dagens datasystemer stadig blir større og mer sammenkoblede er vi nødt til å akseptere problemer som ikke kan analyseres fullt ut, brytes opp og ryddes vekk. Dette er kompleksitet vi må lære oss å leve med. Da er det viktig å erkjenne at datasystemene ikke alltid er verktøy og et virkemiddel vi kan løse problemene våre med, men at de faktisk også kan bli en kilde til utfordringer i seg selv.

Med økt sammenkobling vil risikoen for at små feil i enkeltsystemer forplanter seg og får store konsekvenser øke. Sammenkobling gjør det også vanskeligere å bytte ut gamle systemer da konsekvensene for andre systemer er vanskelig å forutsi.

Vi forventer i dag at datasystemene vi bruker skal være koblet sammen og fungere slik at det kan gi oss store fordeler i forhold til effektivitet. Men slik integrasjon har også en bakside som innebærer problemer fra tid til annen, problemer vi ikke kan bli kvitt.

«Standardiser eller dø!»

Det investeres årlig milliarder i IT-løsninger, og vi vil fremover investere store summer i datasystemer i offentlig og privat sektor i Norge. Som et eksempel, Helse Sør-Øst har pr. i dag rundt 3500 forskjellige datasystemer i drift på 12 forskjellige plattformer. Det var derfor naturlig at Viseadministrerende i Helse Sør-Øst holdt et foredrag om denne utfordringen på IT-tinget den 25. september 2013 med den betimelige [tittelen «Standardiser eller dø!»](#)

I foredraget beskrev han det omfattende arbeidet de har igangsatt for å standardisere og konsolidere systemene de bruker i sin virksomhet. Kompleksiteten i prosjekter av en slik størrelse er vanskelig å forestille seg. I tillegg til størrelsen er også slike fellessystemer virksomhetskritiske. Stopper de, så stopper virksomheten. Om vi skal lykkes med slike prosjekter må vi finne gode strategier som gjør at vi kan håndtere kompleksitet. Vi må helt grunnleggende analysere og lære av de feilene vi eller andre har gjort med systemer av slike dimensjoner.

Vi må lære at datasystemene vi innfører vil være vanskelige å koble fra og derfor leve lenge. Og vi må derfor stille konkrete krav om at de må være tilpasset endring, at de er fleksible og åpne. Vi må ta til oss den involverte kompleksiteten og gjøre reelle vurderinger om realismen i mange av dagens og fremtidens IT-prosjekter. Og sist, men ikke minst, vi må ikke la oss overraske at vi også fremover vil stå overfor IT-problemer som fører til togstans, t-baneforsinkelser, og utfordringer for politi så vel som i helsevesenet.

Publisert: 12.des. 2013 09:39