Notat om begrepet “aktør” i Modul B i IN1030

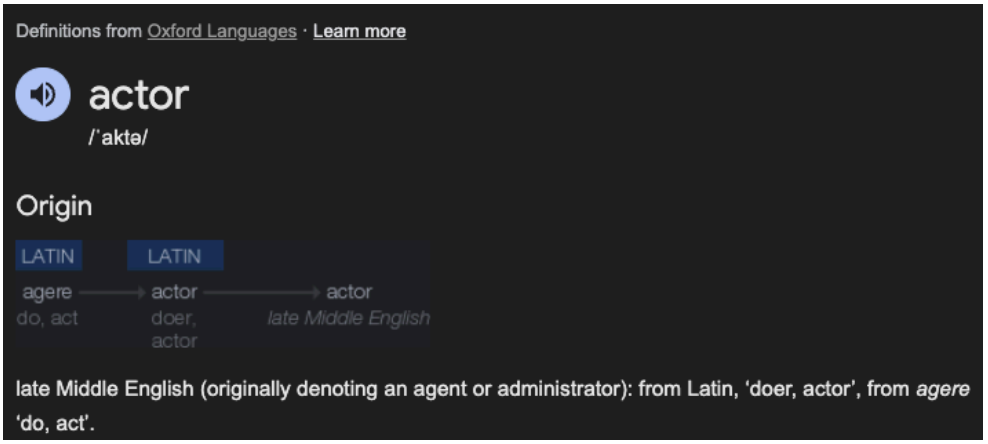
Gunnar Rye Bergersen, 26. mars 2026, version 1.2

Introduksjon

Hva som menes helt presist med begrepet “aktør” kan være en utfordring å fastslå, noe jeg har fått flere gode tilbakemeldinger fra studenter og gruppelærere på. Dette notatet belyser noen perspektiver som jeg håper kan være oppklarende.

Opphav

Ser man på opphavet til det engelske “actor” (Google søkestreng: “actor etymology”) får man oppgitt følgende:

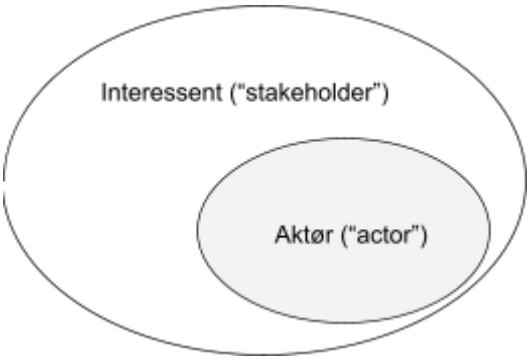


Det mest sentrale ved dette begrepet er tydeligvis det å *handle* eller *agere*, eller mer uformelt det å *gjøre* noe på en aktiv måte.

Interessent og aktør

I IN1030 introduserer vi begrepet “interessent” i Modul B og diskuterer av og til forskjeller og likheter mellom “aktør” og “interessent.” Forholdet mellom de to begrepene som vi har valgt å ha i kurset[1] fremgår av illustrasjonen under hvor alle aktører er interessenter, men ikke omvendt (se gjerne hva Digitaliseringsdirektoratets prosjektveiviser skriver om interessenter i konteksten av å gjennomføre gode IT prosjekter). I forhold til alle som har *interesser* vedrørende systemet, er det bare et fåtall som er *aktive* og *gjør* noe med eller på systemet.

[1] Merk at andre, f.eks. i industrien, kan bruke de to begrepene annerledes for gode grunner og vi har ingen definisjonsmakt i dette kurset til å bestemme hvordan andre skal bruke begrepet.



Illustrasjon: Venn diagram for interessent og aktør

Beskrivelser (definisjoner?) av aktør i Modul B

Gruppelærer My Schultheiss har vært grei og gjennomgått forelesninger, pensum, og lysark fra gruppetimer (med mer) og funnet flere beskrivelser av aktør i kurset:

1. "En aktør for et system representerer en rolle som et menneske eller et annet system som har et mål med systemet."
2. "En aktør er en som har en aktiv rolle med systemet og faktisk anvender systemet."
3. "Primær aktør har et eget mål i kommunikasjon med systemet."
4. "En aktør er noen som bruker eller interagerer med systemet, eller et system som brukes av systemet."
5. "En primæraktør er en bruker som har et eget mål med å anvende systemet." "En sekundæraktør hjelper primæraktøren med å oppnå sitt mål i systemet."

Samlet kan det sies at beskrivelsene er forholdsvis like, men at man får et noe forskjellig inntrykk av hva en aktør er avhengig av hvilken av de fem beskrivelsene man benytter. (Jeg er forøvrig bevisst forsiktig med å bruke begrepet "definisjon" om de fem beskrivelsene over da enhver kort definisjon av "aktør" vil skape en ny diskusjon om hva ordene i den korte definisjonen betyr, se gjerne hva den kjente fysikeren Feynman tenkte rundt dette temaet.)

Sommerville definerer ikke aktør eksplisitt, men introduserer begrepet i seksjonen "use cases" (4.4.3) i relasjon til Figur 4.15: "actors in the process, who may be human or other systems, are represented as stick figures." Som eksempel på relevante aktører i forhold til Mentcare caset i boka nevnes sykepleier, resepsjonist, doktor og leder. Dette er eksempler som er konsistent med de fem beskrivelsene av aktører over. Det kan også være relevant å se hvordan industrien (offentlig og privat sektor) bruker begrepet aktør. Digitaliseringsdirektoratet redegjør for hovedaktører i kontekst av digital postkasse, samt mulige roller som eksisterer for dette caset.

Hvorfor modellere aktører i kravspesifikasjonen?

Av og til kan det være lurt å ta et steg tilbake å vurdere hvorfor man i det hele tatt bør innføre et spesifikt begrep som "aktør." Kontekst er som følger:

- Man har hatt en større gjennomgang av brede sosiotekniske konsekvenser av å bygge systemet og man har satt seg inn i en del lovverk i tillegg (Modul A).
- Man har deretter fått jobbet med å forstå krav til systemet ved å snakke med bl.a. interessenter og man har muligens etablert en prosessmodell man tror kommer til å fungere (første del av Modul B).

- Tiden er nå inne for å bli såpass konkret slik at man kan starte å planlegge hvordan skrive kode for et system basert på det man tror man nå vet. Påfølgende steg vil være å prøve dette ut (hva som til slutt viser seg å fungere finner man ut senere).

Hva er nå kritisk å gjøre? Vel, vi vet at vi må bygge et system og dette systemet kommer til å interagere med omverden, enten via brukere eller via andre datasystemer. En konkret plan for hvordan denne interaksjonen kommer til å foregå er nødvendig.

Hvem har nytte av hver spesifikk modell som lages?

Modellering av systemet (Kapittel 5 i Sommerville) beskriver mye brukte modeller i UML på flere abstraksjonsnivåer. Noen modeller er veldig overordnet og primært nyttige når man kommuniserer tanker og ideer til andre for innspill. Andre modeller er veldig detaljert for å vise helt konkret hvordan man to systemer skal kommunisere slik at en utvikler har tilstrekkelig informasjon for å lage en løsning som fungerer. Man modellerer i tillegg for å prøve å forstå selv hva man skal gjøre og for å vurdere alternativer for hvordan man kan redusere den kompleksiteten som alltid eksisterer når man bygger store IT systemer.

De modeller som vi diskuterer i dette kurset har til felles at de vurderes i forhold til om de er *nyttige* for noen. De kan f.eks. være nyttige for interessenter eller bruksdesignere, arkitekter, sikkerhetseksperter eller programmerere – men det er lurt å tenke igjennom hvem det er man lager en modell for. Hvilke behov eksisterer for denne personen, gruppen eller organisasjonen man har i tankene når man modellerer?

Ved å modellere interaksjonen mellom systemet og brukere, samt andre systemer som systemet under utvikling benytter eller benyttes av, er grunntanken at man prioriterer å gjøre en aktivitet som er kritisk for at man kan starte å utvikle noe i det hele tatt. Denne aktiviteten er å modellere de som agerer, eller samhandler, med systemet, nemlig aktørene.

Hva med en utvikler eller et utviklerteam som aktør?

Jeg har merket meg at mange også ser utviklerne i prosjektet som aktører. Dette er helt riktig når man snakker om utviklingsprosessen, men når konteksten er at vi skal modellere de som aktivt *bruker* systemet blir ting litt annerledes. Det er forholdsvis uvanlig å modellere utviklernes handlinger i systemet sammenlignet med å modellere sluttbrukernes handlinger. Jeg har ikke funnet eksempler på utviklere som aktører som må modelleres for et system i verken Sommerville eller i tidligere eksamensoppgaver, noe som illustrerer poenget mitt. Use cases som involverer utviklere er for øvrig ofte også mer komplekse enn det som jeg mener bør dekkes i et introduksjonskurs.[2] Merk også svakheten med å (mekanisk) sjekke ut om et begrep er dekket av en spesifikk definisjon – noen begrep er vanskelige å dekke ved kun en kort setning, noe som gjør at man av og til benytter konkrete eksempler på hva et begrep er ment å omfatte isteden, se gjerne Wikipedia for en oversikt over forskjellige typer definisjoner.

[2] Et tenkt use case hvor en utvikler er en bruker kan f.eks. være når en (vanlig) bruker har prøvd å gjøre noe på systemet som feiler. En utvikler i andre eller tredje linje support må deretter logge seg inn i med administratorrettigheter i det samme systemet og rette opp problemet. Det finnes også gode use cases rundt produksjonssetting av kode for å håndtere feilsituasjoner som oppstår ved at man oppdaterer systemet til en ny versjon mens en bruker er inne og utfører en endring.