

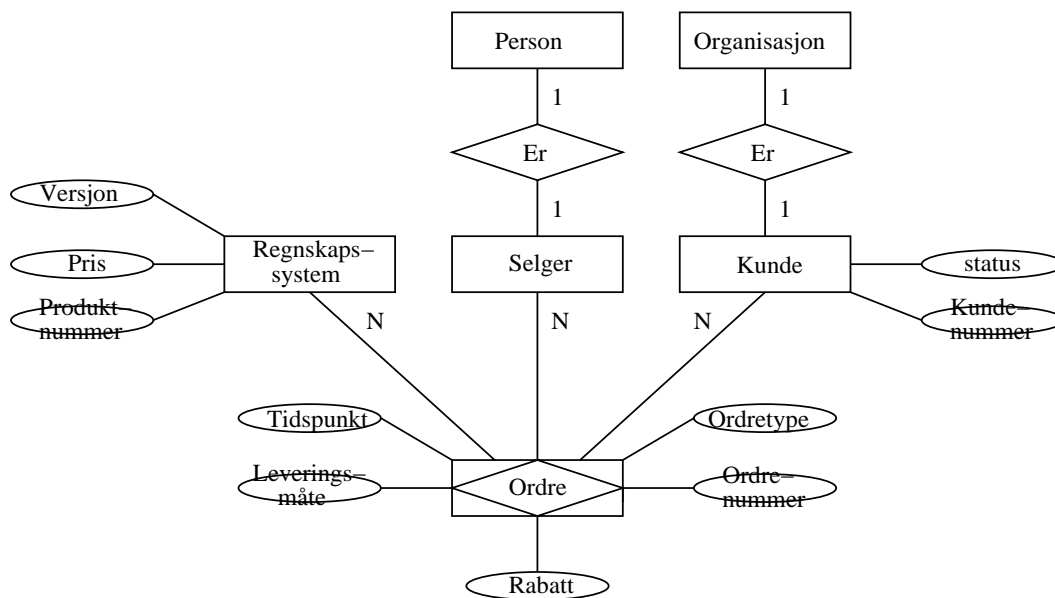
LØSNINGSFORSLAG

45160 SYSTEMERING 1

MANDAG 13. MAI 1996

Oppgave 1

a)

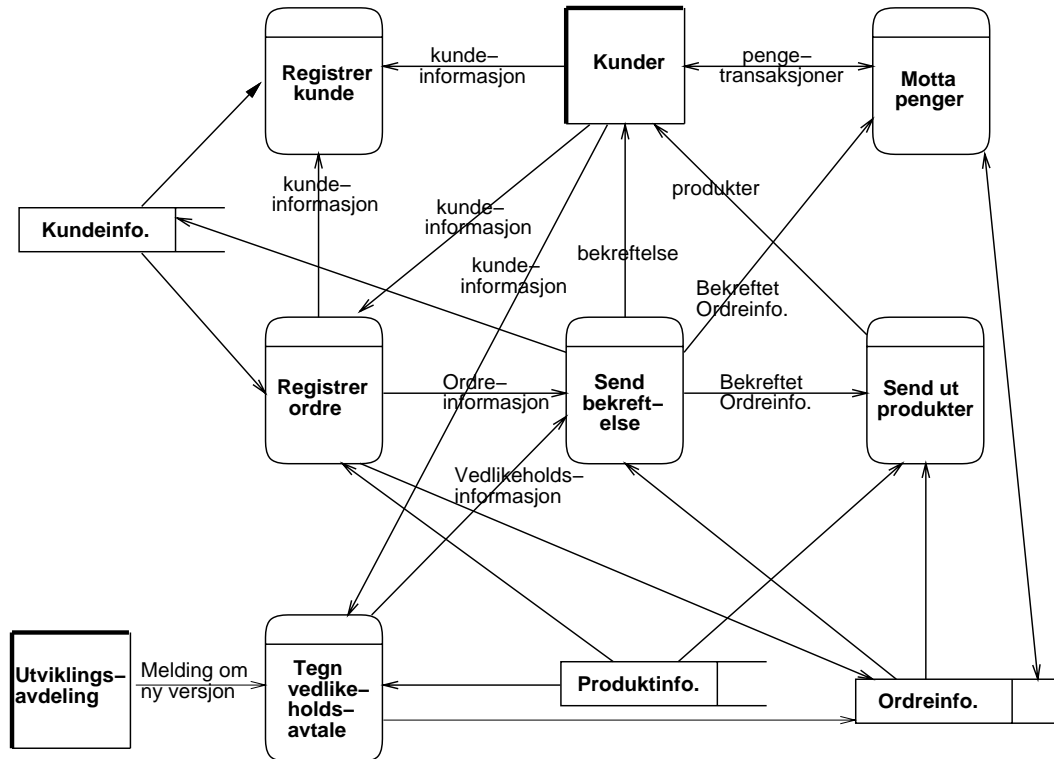


Kjerneentiteten ORDRE i diagrammet er både en entitet og en relasjon. Dette er derfor markert ved å bruke både et rektangel og en diamant. ORDRE har derfor relasjoner til andre entiteter når den selv er en relasjon. ORDRE som entitet er relatert til andre entiteter via relasjoner. ORDRE som entitet har også attributter, f.eks. ORDRETYPE forteller om ORDRE er en salgsordre eller en vedlikeholdsordre.

ER diagrammet over er bare en skisse. Selgere vil selvfølgelig ha en mengde attributter. Noen av disse attributtene vil dekkes av at de også er PERSONER. En kunde vil også ha mange attributter, hvorav mange vil dekkes av ORGANISAJON.

Ordrebekreftelse, kontrakt osv. er unødvendig, fordi de vil inneholde de samme opplysningene som en ORDRE i ER-diagrammet.

b)



Beskrivelsen nedenfor forteller litt mer om hva hver prosess gjør:

Registrere ordre Registerer ordre fra kunder som vil ha regnskapssystemer. Tar seg også av alle endringer i ordren.

Send bekreftelse Sender ut bekreftelse som inneholder relevante informasjoner om ordren.

Registrer kunde Legger inn nye kunder i systemet. Systemet har neppe de samme kundene hele tiden, slik at denne prosessen er nødvendig. Men denne prosessen kan selvfølgelig gjøres i andre systemer som f.eks. tar seg av markedsføring.

Send ut produkter Prosessen passer på at kunden får de produktene han ber om.

Motta penger Tar seg av regningsutsendelse og overføring av penger. Hvis pengene ikke kommer sender systemet ut en puring.

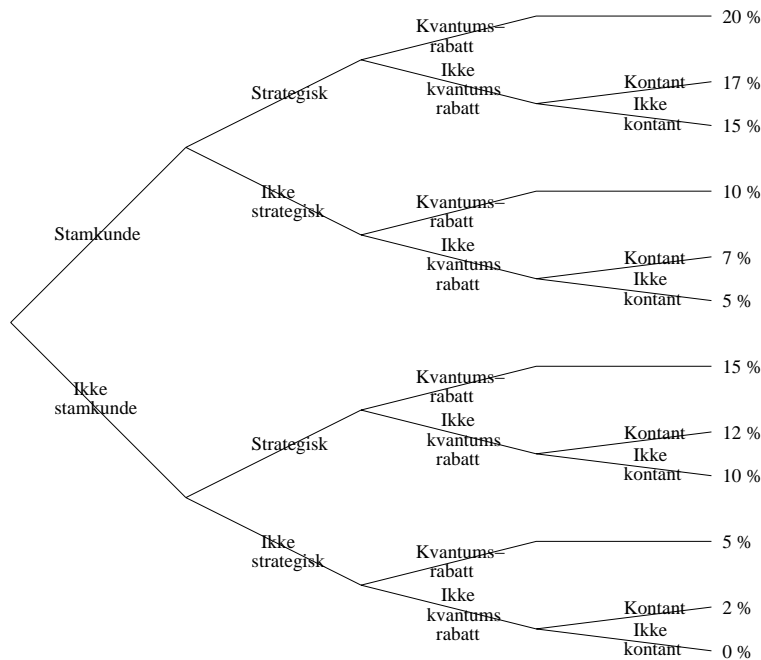
Tegn vedlikeholdsavtale Vedlikeholdsvilkårene avtales med kunden. Prosessen sørger for at vilkårene oppfylles.

Momenter å legge vekt på:

- På et overordnet nivå er det viktigst å få med de funksjonelle aspektene som har konsekvenser for interaksjonen mellom systemet og omverdenen. Den interne virkemåten til systemet er ikke så viktig. Spesielt vil oversettelse til japanske ikke ha så mye interesse på et overordnet nivå, siden dette har med implementasjonen å gjøre.
- Mye informasjon i problembeskrivelsen er ikke relevant for å lage en oversikt over prosessene i ordresystemet: hvordan datamaskinene er plassert, hvem som arbeider i Kjekkk & Grei AS, at underleverdørene i Japan må samarbeide med de store japanske bilfabrikkene. . .
- Systemet vil få interaksjon med andre systemer i bedriften. F.eks. vil det sikkert finnes informasjon om de ulike produktene bedriften selger i andre systemer.

Oppgave 2 (20%)

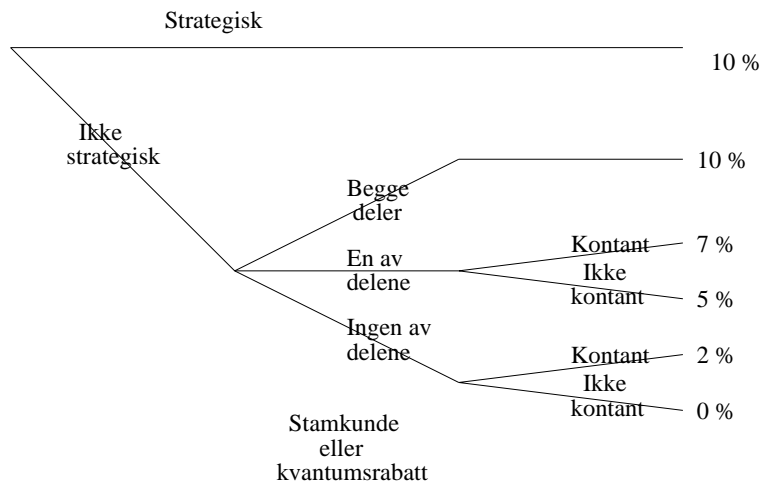
a)



Oppgaveteksten er litt uklar, slik at det ikke er helt klart om f.eks. Vilhelm Erobreren jr. jr. gir sin kvantumsrabatt i tillegg til rabatt for strategiske kunder som blant annet Dutta Yndig gir. Det er mulig å ta hensyn til denne komplikasjonen ved å legge hvem

selgeren er i beslutningstreet, men dette vil gjøre treet unødvendig komplisert. En bedre løsning er å bruke det samme beslutningstreet som over, men å heller markere at noen av utfallene ikke er mulig hvis en strengere tolkning legges til grunn. Dette kommentaren viser noe av utfordringen med modellering: poenget er å få tak i det vesentlige uten at oversikten går tapt, noe som bestandig er en avveining.

b)

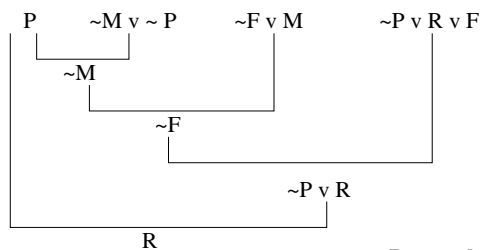


Oppgave 3 (20%)

a)

- P KG selger flere produkter
- R KG gir rabatt
- F KG forbedrer produktene sine
- M KG kommer for sent på markedet

Uttrykkene i teksten:	P	
	$P \rightarrow (R \vee F)$	$\sim P \vee R \vee F$
	$F \rightarrow M$	$\sim F \vee M$
	$M \rightarrow \sim P$	$\sim M \vee \sim P$



Det er altså mulig å slutte seg til at KG gir rabatt

b)

G Kundene blir glade

$G \rightarrow R$

For å vise at tillegget er konsistent, er det nok å vise at negasjonen av tillegget er inkonsistent. Negasjonen blir i dette tilfellet:

$\sim(\sim G \vee R)$ G $\sim R$

Vi tar også med oss R fra oppgave 3 (a). Resolusjon viser dermed:

$\sim R$ G $\sim R$
└──────────────────┘
□

Oppgave 4 (20%)

- Se pensum i kapittel 3.6: finne forgreningssannsynlighetene for hver transaksjon. Finn hvor mye hver transaksjon bruker av ressuser. Kan så beregne hvor mye last applikasjonen påtrykker et system.
- Se pensum i kapittel 3.5.