



NORGES TEKNISK-
NATURVITENSKAPELIGE UNIVERSITET
INSTITUTT FOR DATATEKNIKK OG INFORMASJONSVITENSKAP

Faglig kontakt under eksamen:
Arne Sølberg
Tlf: 73 59 34 38 / 91 89 73 31

EKSAMEN I FAG SIF 8035 INFORMASJONSSYSTEMER

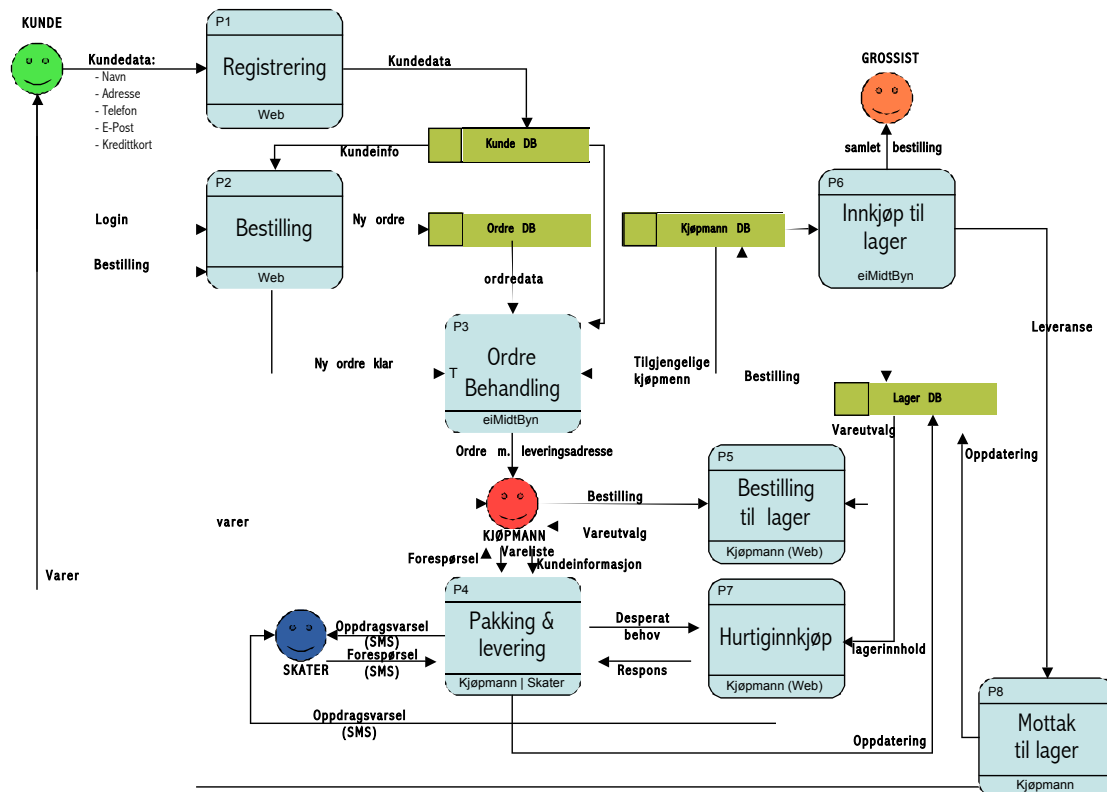
Mandag 14. mai 2001

Tid: kl. 0900-1400

Løsningsforslag

Oppgave 1 – Prosessmodell DFD (30%)

Studer case-beskrivelsen for eiMidtByn.com. Lag et overordnet Data-Flyt-Diagram (DFD) som viser den handelsprosessen som er beskrevet.



Kommentarer

- P1: Kundene må registrere seg før de kan handle
- P2: Kunder surfer på web og legger inn sin bestilling. Det er ikke spesifisert noe om hvordan dette foregår i CASE't. I virkeligheten må det designes en skikkelig web-butikk med vareutvalg, handlekurv etc. Dette er ikke modellert her.
- P3: Prosessen starter når en ny bestilling foreligger. Dette er markert med trigger (ikke forespurt). eiMidtbyn sjekker bestillingen, ser over tilgjengelige kjøpmenn og genererer en ordre til aktuell kjøpmann. Denne prosessen kan i prinsippet automatiseres, derfor er det her ikke modellert inn en eksplisitt ordrebehandler hos eiMidtbyn.
- P4: Kjøpmann pakker varene og sender oppdragsvarsel til aktuelle skatere. Første skater som møter leverer varene til kunden. Ved problemer vil skateren sende forespørsel til kjøpmann via sms. Det er et valg hvorvidt varer og sms-interaksjon skal modelleres i en DFD-modell på dette nivået. Varene er typisk fysisk/materiell flyt. Det tatt med her for å indikere at kunden faktisk mottar varene sine, og kan betraktes som en pakkseddel, kvittering e.l. SMS interaksjonen er i utgangspunktet kun interaksjon mellom to aktører og behøver slik sett ikke modelleres. Den er imidlertid tatt med ettersom det kan tenkes at sms funksjonaliteten skal støttes av en egen meldingsapplikasjon hos kjøpmannen.
- P5: Bestilling til lager. Kjøpmannen sjekker sin lagerbeholdning og legger inn den nye bestillingen
- P6: eiMidtbyn samler opp innkomne bestillinger og sender samlet bestilling til aktuelle grossister. At eiMidtbyn forhandler seg frem til egnede avtaler med grossister er ikke modellert her. Dette regnes for å være en manuell aktivitet og i allefall utenfor det som er beskrevet i case't.
- P7: Når kjøpmannen under pakking av varer sammenlikner varelisten med det han har av varer i hyllene, og oppdager et desperat behov for å kunne klargjøre denne ordren, så utfører han et hurtigkjøp ved å sjekke lagerbeholdningen hos andre kjøpmenn og sende skater'n på tur.
- P8: Kjøpmannen mottar varer fra grossist stabler disse på lager og oppdaterer lagerbeholdningen.

Aspekter vedrørende fakturering og regning/betaling er ikke modellert her. Dette står det forsvinnende lite om i case't og det er dermed ikke fokusert på det. Et alternativ er å la alle slike transaksjoner foregå elektronisk og la dette styres av P3 "ordrebehandling".

Oppgave 2 – Tamme og slemme problemer (20%)

I pensumartikkel P15 "Engineering design principles for unsurveyable systems" defineres begrepene tamme og slemme problemer/systemer.

- a) *Beskriv egenskapene og forklar forskjellen mellom tamme og slemme ("wicked") problemer*

Ikke alle systemer kan sies å være konstruktive. Dette gjør at en del systemutviklingsmetoder som f.eks. bygger på Langefors systemutviklingsprinsipp ikke kan anvendes på alle problemer. Dette har gjort at man skiller mellom tamme og slemme problemer.

Tamme er f.eks. enkle matematiske problemer der problemstillingen kan formuleres presist og basert på gitte lover eller formler. Slemme problemer er i kontrast vanskelig å formulere presist og der løsningene som oftest er underlagt "umulige" eller konflikterende krav.

I pensum er det listet en del egenskaper som skiller de tamme og slemme problemene:

- Slemme problemer kan ikke formuleres presist og entydig
- En formulering av et slemme problem tilsvarer en antakelse eller et utsagn om problemets løsning
- Slemme problemer kan alltid løses bedre/på en annen måte
- Alle slemme problemer kan betraktes som et symptom på et annet problem
- Ethvert slemme problem er i all vesentlighet unikt

Diskusjonen av disse finnes i pensum artikkel P15. Det kreves ingen utførlig diskusjon av disse, men det viktigste er at besvarelsen viser en forståelse for forskjellen mellom disse problem-typene og senere (i b) at løsningen på disse ikke ligger i variasjoner over tradisjonell, fasedelt systemutvikling eller forbedrede SW-engineering metoder. Prinsippene bak disse problemene og derfor også for løsningen av dem ligger på et annet nivå en faktisk innhold i systemutviklingsmetoder.

b) Gi en oversikt over prinsipper for løsning av slemme problemer

Prinsippene er også her definert i pensum artikkel P15. Igjen er disse bare listet her. Se kommentar i a):

- For slemme problemer finnes det ingen eksperter
- Ingen vil planleggs for
- Løsningen må være transparent
- Problemløseren til et slemme problem, bærer heller preg av å være en jordmor enn å være en lege som tilbyr en passende behandling
- Problemløseren er nødt til å utvise en form for forsiktig respektløshet (og optimisme) over problemet
- Det er viktig å bygge allianser
- Løsningen til et slemme problem er en politisk argumentative prosess

Oppgave 3 – Brukergrensesnitt (25%)

a) *Definer relevante mål for brukskvaliteten til systemet.*

I denne oppgaven ønsker vi å se om kandidaten kan si noe bedre enn at systemet skal være brukervennlig og lett å lære. Kap. 19.3 definerer 4: learnability, throughput, flexibility, attitude.

Formuleringer av det samme kan være:

- a. Mengden opplæring som trengs
- b. Tidsbruk (relativt til før)
- c. Behandlingskapasitet (relativ)
- d. Feilraten ved bruk (relativt/absolutt)
- e. Tilfredshet/trygghet (kvalitativt, f.eks. rating i spørreundersøkelse)

Det er naturlig å angi en enhet for målet og om målet er relativt eller absolutt. I oppgave 3 er det et pluss dersom en kommenterer disse målene, selv om det kan være unaturlig å gå rett på disse målene ved evaluering av først-utkastet til design.

b) *Ta utgangspunkt i case-beskrivelsen og lag en modell av oppgavene som kjøpmennene har knyttet til leveranser. Bruk hierarkisk oppgave-analyse (HTA) eller den foreleste TML-notasjon. Hvordan kan Data-Flyt-Diagrammet laget i oppgave 1 være til hjelp? Forklar hvordan denne oppgavemodellen kan være til hjelp i resten av utviklingsprosessen.*

Tre elementer er viktige i besvarelsen:

- i. Lese relevante opplysninger ut av case-beskrivelsen (og DFD)
- ii. Bruke modelleringskonstruksjoner som nedbrytning, sekvens og valg
- iii. Bruk av oppgavemodellen i design og evaluering

Relevante opplysninger går både på avgrensninger og konkrete oppgaver ifm. håndtering av bestillinger. Formelt riktig bruk av notasjonen tillegges liten vekt, poenget er dekomponering, sekvens og valg/alternative forløp. Merk hvordan oppgave 1 fører til at del-oppgaver for forskjellige bestillinger kan flettes. Dette er vesentlig for å fange opp kompleksiteten i kjøpmannens hverdag. Merk også hvordan noen oppgaver gjentas for et sett enheter. Oppgave 3 inngår som en deloppgave to steder, og er derfor løftet ut (slags generalisering).

Hierarkisk oppgavemodell (tre deler):

1. Håndtere leveranser (ingen begrensning)
 - motta bestillinger
 - håndtere leveranse (for hver bestilling, sekvens)
 - håndtere 1 bestilling
2. Håndtere 1 bestilling (sekvens)
 - motta bestilling (inkl. adresse og leveringstidspunkt)
 - forberede leveranse (for hver vare i leveransen, sekvens):
 - forebered vare (valg):
 - dersom vare mangler (sekvens)
 - finn nærliggende kjøpmann
 - handle vare elektronisk
 - håndtere ungdomsoppdrag (oppg. 3)
 - hent vare i lokalt lager
 - pakk vare
 - håndtere ungdomsoppdrag (oppg. 3)
3. Håndtere ungdomsoppdrag (sekvens)
 - sende ut SMS om oppdrag
 - gi leveranse til første ungdom
 - håndtere avvik, f.eks. kunde ikke hjemme, vare ikke tilstede (repeterende sekvens)
 - motta SMS
 - svare på SMS

Det vil være naturlig å lage en liten domenemodell, med begrepene bestilling/leveranse, vare, lager(sted) og skater, men de er vant til at dette er en del av domenemodellering/statisk modellering, som er utelatt fra denne eksamenen. Det er ofte vanskelig å unngå å anta noe om systemets oppgaver ifm. gjennomføringen, f.eks. om systemet mottar SMS'er for kjøpmannen eller om han/hun bruker mobilen i tillegg til systemet. Det bør legges vekt på at oppgavene i modellen baserer seg på caset og ikke på et tenkt design som ennå ikke er gjort/bedt om.

Dataflytdiagrammet kan "generere" oppgaver på to måter:

- 1) navngitte prosesser tilsvarer overordnede oppgaver
- 2) hendelser i form av innflyt trigger utførelse av oppgave, f.eks. valg av ett av flere alternativ

Oppgavemodellen vil selvsagt styre designet, men også gi verdifull input til evaluering, f.eks. for generering av konkrete oppgaver for gjennomføring i tester, og evt. om det er behov for logging av "virkelige data" forut for evaluering, for å gjøre denne mer realistisk.

- c) *Ved siden av oppgavemodellen, hva slags informasjon i caset er nyttig for å lage et godt første-utkast til design? Hva slags type prototyp er det naturlig å lage basert på designet, og hvilke evalueringsteknikker er relevante for å prøve den ut?*

Fra caset identifiseres:

- a. brukertypen kjøpmann, som skal designe for
- b. plattformen som brukergrensesnittet skal kjøre på, inkl. inn- og ut-enheter, f.eks. at grensesnittet er tastaturløst
- c. omgivelser som brukergrensesnittet skal inngå i, som f.eks. kan regnes å være hektisk, med liten tid til å fokusere på grensesnittet

Informasjon som ikke finnes i caset er karakteristika ved brukermassen, retningslinjer og standarder for design for den aktuelle plattform og hvordan en butikk ser ut.

To aspekter ved designet synes naturlig å få undersøkt gjennom evaluering, og dette styrer hva slags prototyp som er naturlig å lage:

- 1) Validering av oppgavene, dvs. relevansen, innholdet og strukturen
- 2) Om designet er tilpasset omgivelsene og de andre aktivitetene i butikken

Den første typen evaluering burde kunne gjennomføres med papirprototyping og et representativt utvalg kjøpmenn. På dette tidspunkt er det ikke så viktig med ekspertvurderinger av designet, siden det er validering av oppgavene som er fokuset. Evalueringen er ikke eksperiment-orientert, og kan godt være basert på høyttenkning.

Den andre typen krever en noe mer vertikal prototyp med langt høyere "fidelity", f.eks. for å teste om kompleksiteten til skjermbildene er passe, om inputmulighetene er tilpasset pekeskjerm etc. I dette tilfellet er det naturlig med felt-testing i en butikk med mer logging/opptak av hva som skjer, både av grensesnittet og av omgivelsene.

Merk at det er litt tidlig å evaluere mht. kvalitetsmålene formulert i deloppgave a).

Oppgave 4 – ERP systemer, SAP (25%)

SAP R/3 skal brukes til å realisere eiMidtByn's innkjøpstjenester.

- a) *SAP utviklingsmetode*

Fra kirchmer, "Phases of a Business Process Oriented (BPO) implementation", henter vi følgende oversikt over faser og aktiviteter:

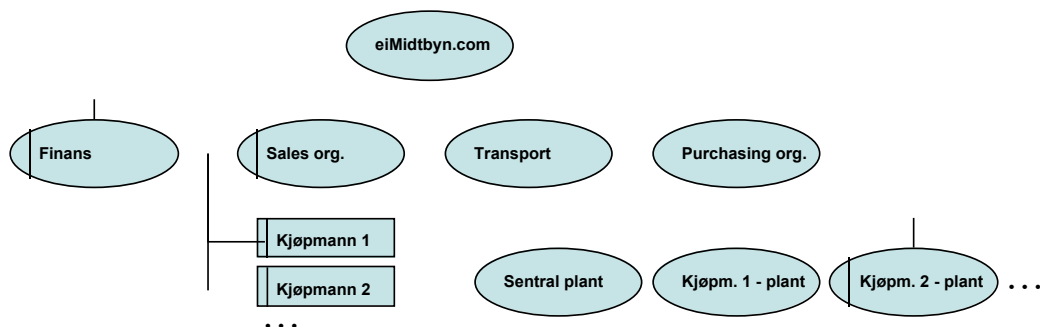
- Strategi-basert (utvikling av) BPO-konseptet

- Bestem prosjektets scope , ("target system"), dvs. hva som skal støttes av ERP systemet og likeledes hva som ikke skal støttes av ERP systemet
- Utvikling av *produkt* og *markeds*modeller
- Definer forretningsprosessene (*prosess*-modeller)
- Standard software-basert (utvikling av) BPO konsept
 - Velg/finn relevante biter av referansemodellen
 - Detaljer prosessbeskrivelsen på/ned til transaksjonsnivå
 - Bestem ansvarlig organisasjonsenhet for de ulike steg i prosessmodellen
- BPO implementation
 - IT aktiviteter
 - Organisasjonsaktiviteter
 - "Live Start" -> overgang fra build-time til run-time operation
- Optimalisering av forretningsprosessen etter at man har gått "Live" (Continuous Process Improvement).
 - Gjennomgang/revidering av rammekrav til systemet
 - Vurder oppfyllelse av strategiske mål
 - Tilpasning av BPO implementasjonen
 - Tilpasning av BPO konseptet

Svaret bør inneholde en kortfattet beskrivelse av hva som er relevant – og innholdet i de aktuelle modellene for eiMidtbyns tilfelle.

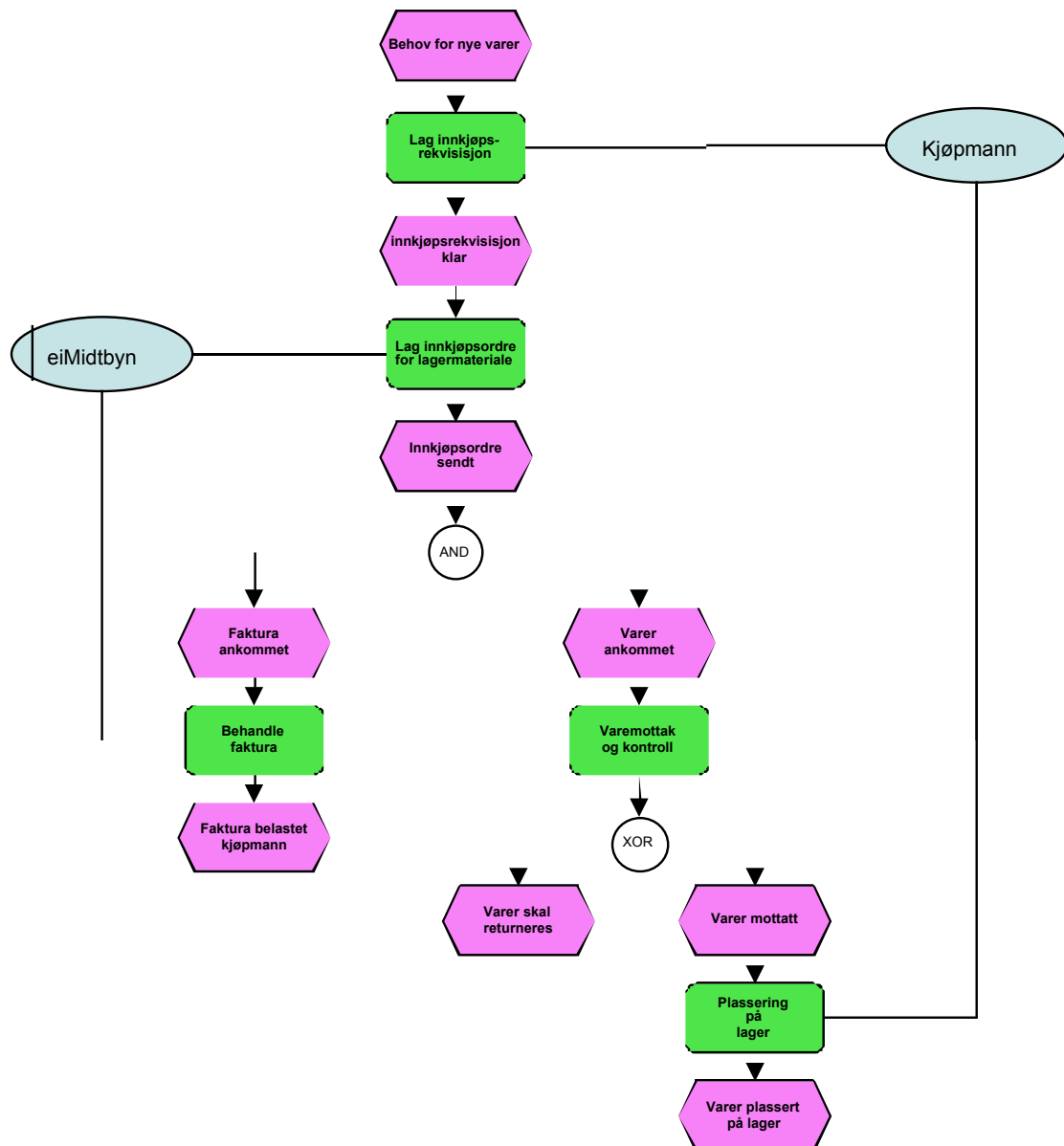
b) *Modellering av innkjøpsprosessen i SAP*

- **Organisasjonsmodell (ARIS Organizational view):**



Organisasjonsmodellen kan settes opp på flere måter. Den mest naturlige er å la de enkelte kjøpmennene være plants (altså varehus som mottar og lagrer varer), men dette har vi snakket lite om i forelesningene. Bruk skjønn.

- *EPC-modell (ARIS Control view):*



Modellen er basert på SAP procurement logistics modellen, men er grovt forenklet for tilpasning til eiMidtbyn.com. Modellen gjør en del antakelser ift. Case't, f.eks. at det skal sendes regning og at varer kan bli returnert. Disse antakelsen stammer fra den opprinnelige SAP procurement logistics modellen. Modellen omfatter ikke lyninnkjøp.