NORGES TEKNISK-
NATURVITENSKAPELIGE UNIVERSITET
INSTITUTT FOR DATATEKNIKK OG INFORMASJONSVITENSKAP

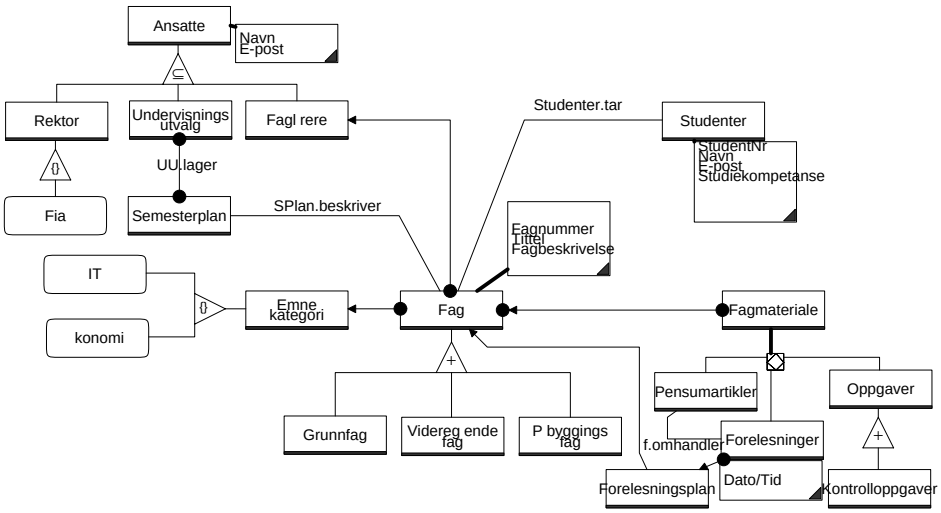
Faglig kontakt under eksamen:
Terje Brasethvik
Tlf: 73 59 36 71 / 90 95 91 85

EKSAMEN I FAG SIF 8035 INFORMASJONSSYSTEMER
Lørdag 20. mai 2000

Løsningsforslag

Vektleggingen av oppgavene er indikert med prosent (veiledende). Under en oppgave teller alle deloppgavene likt.

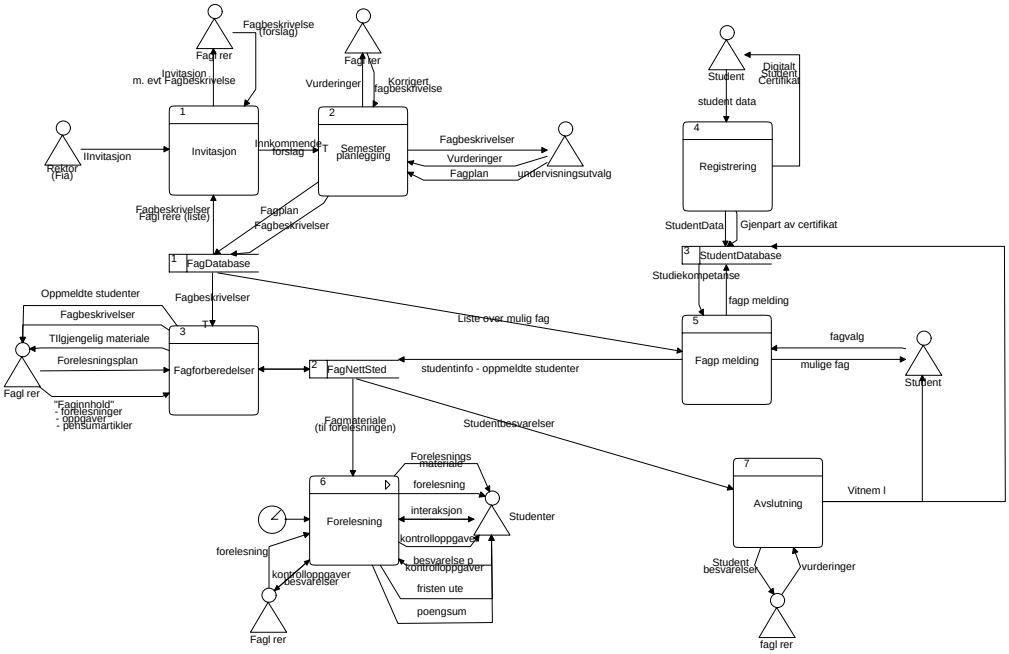
Oppgave 1 – Referent Modell (20%)



- I modellen er det lagt vekt på å ha med få begrep. Grovt sett tre "kategorier" av begrep: Fagpersonale (*Rektor, Undervisningsutvalg* og *Faglærere*), *Fag* og *Fagmateriale*.
- Det mest sentrale begrepet er *Fag*. Det meste er sentrert rundt eller relatert til dette begrepet.
- Alle fag er delt inn i *Grunnfag, Videregående fag* eller *Påbyggingsfag*. Disse er ikke overlappende. I tillegg har alle fag en *Emnekategori*.
- *Fagmateriale* består av *pensumartikler*, *forelesninger* og *oppgaver*.
- Alle forelesningene er beskrevet i *Forelesningsplanen*. Forelesningsplanen kunne vært utelatt som eget begrep ettersom alle forelesningene selv har dato/tid for når de finner sted, og selv har en referanse til pensum som omhandles.

Oppgave 2 – Prosessmodellering (DFD) (30%)

a) Overordnet dataflytdiagram (DFD).

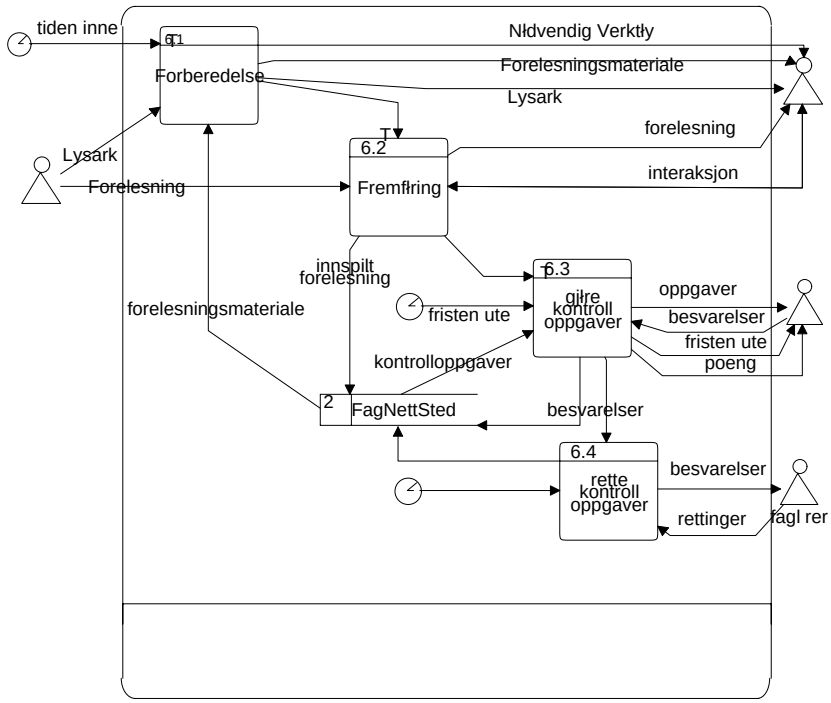


- Modellen viser hovedoppgavene til systemet gjennom et semester. Semesteret starter ved at det sendes invitasjon til potensielle faglærere (P1) og avsluttes ved at faglærer skriver de endelige vurderinger og sender vitnemål til studentene (P7).

- 3 datalagre er i bruk:
  Fagdatabasen som brukes av undervisningsutvalg og rektor for å planlegge fag. Inneholder fagbeskrivelser og semesterets fagplan.
  FagNettsted: ”det opprettes et nettsted for alle fag” som inneholder alt fagmateriale både fra årets og tidligere utgaver av faget.
  StudentDatabase: Alle studenter som er registrert og deres studiekompetanse og progresjon/vitnemål.

- Tidsaspektet for prosessene er forsøkt indikert med timere og triggere. (Ikke forespurt.)

b) Dekomponer aktiviteten som omhandler selve "forelesningen" ett nivå ned.



c) En fundamental forskjell mellom DFD-baserte og objekt/klasse-baserte metoder ligger i måten man dekomponerer systemet på. Forklar forskjellen og poengter hvorfor DFD-diagrammer på denne måten er "inkompatible" med objektorienterte diagrammer.

- I DFD modellerer man prosessene – eller funksjonene – som systemet skal utføre og datataene som flyter mellom prosesser. Man ønsker å skille ut data (datalager, dataflyt) fra prosessenes logikk. I noen dialekter av DFD skiller man også på forskjellige typer flyt, f.ek.s data vs. kontrollflyt. Prosessene i DFD starter ved ankomst av en flyt, de kjører seg ferdig og "forsvinner" når de har sent all nødvendig utflyt. Prosessene har ingen tilstand eller noe liv utover dette.

  I DFD-modeller er all dekomponering fokusert på prosessene – en ren funksjonell dekomponering.

- I objekter eller klasser, ønsker man i utgangspunktet å gjøre det motsatte, ved å *kapsle inn* data, operasjoner på dataene og den lokale tilstanden til dataene inne i objektet/klassen. I disse metodene er det objektet/klassen som dekomponeres – man dekomponerer entitetene ("tingene") fra den verden modellen beskriver ("subject domain decomposition")

Se forøvrig artikkel *P24* – ("A survey of Structured and Object Oriented Software Spectification Methods and Techniques", R. Wieringa, ACM Computing Surveys, vol. 30, no. 4, December 1998.)

Oppgave 3 – SAP-innføring og prosessmodeller (20%)

a) Forklar hva som er spesielt med måten prosessmodellering brukes på i SAP-prosjekter. Hva er de såkalte "Business Blueprints"?

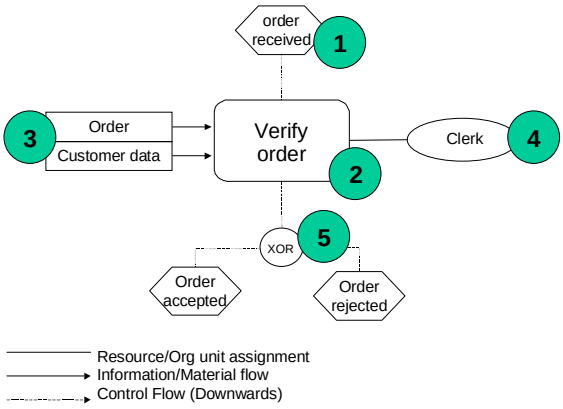
I SAP prosjekter tar man ofte utgangspunkt i eksisterende prosessbeskrivelser – gjerne hentet fra SAP's arkiv med såkalte "best business practices" eller business blueprints. Disse brukes enten som "inspirasjon"/innspill til egen prosessforbedring, eller som referansepunkt når man spesifiserer den forretningsprosessen som SAP skal støtte.

Prosessmodellene som finnes i SAP's arkiv har alle en "direkte" mapping til en underliggende implementasjon i SAP. Når man har formulert en ønsket prosessmodell sammenlikner man derfor med modellene som finnes i arkiv, noe som skal gi et godt utgangspunkt for rask konfigurering av SAP. En viktig aktivitet er sammenlikning av ens egen prosess (enten nåværende eller fremtidig) med de referanseprosesser som finnes.

Det som er spesielt med denne prosessen er mao. at man tar utgangspunkt i eksisterende "best practices" prosessmodeller og at modellene brukes direkte til å konfigurere SAP, noe som skal gi en "rask og billig" implementasjon.

b) Sammenlign modellene og forklar kort hva de forskjellige elementene i modellene betyr.

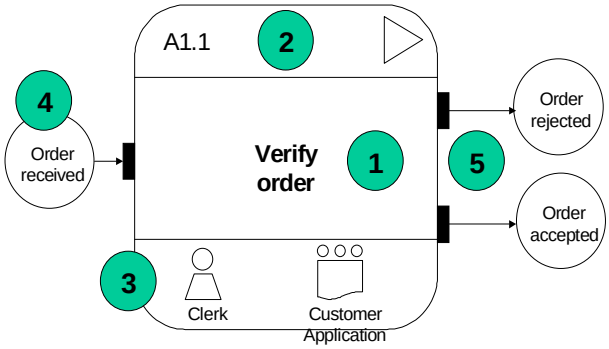
- EPC – Event Process Chain



1. Hendelse ("Event")
2. Funksjon
3. Informasjons-, Materiell- eller Ressurs-objekt
4. Organisasjonsenhet
5. Logisk operator (XOR) som beskriver forholdet mellom hendelser og funksjoner

De tre forskjellige flyt-typer er angitt på figuren.

- APM – Action Port Model



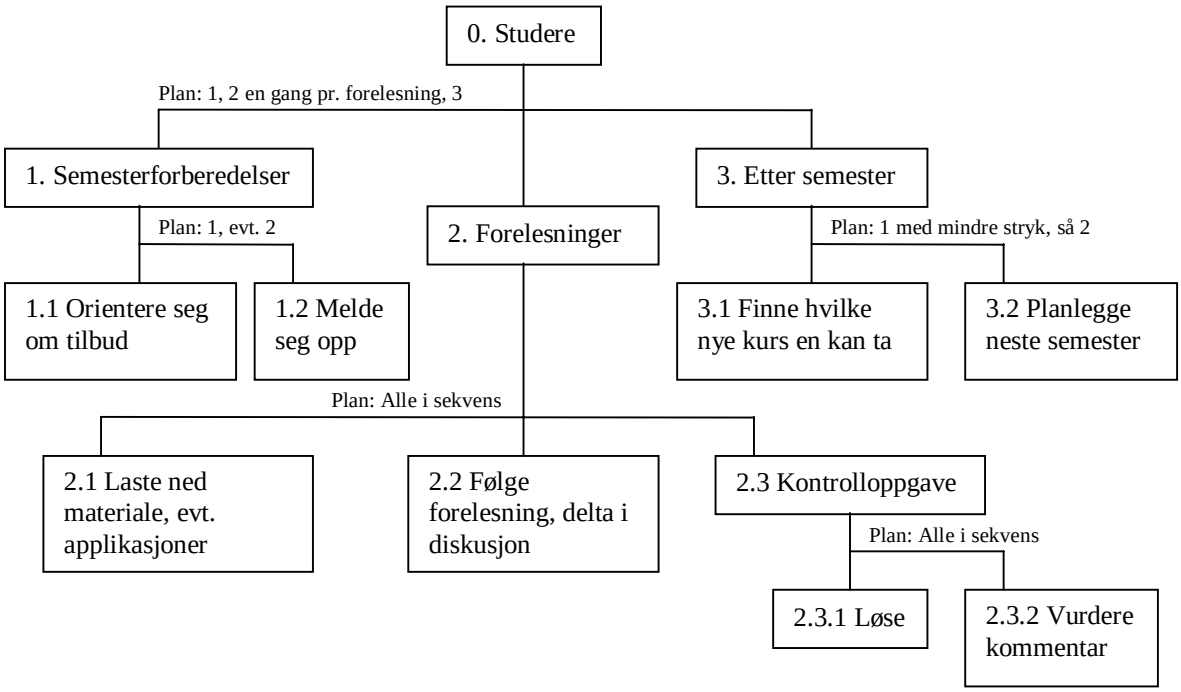
1. Aktivitet
2. Attibutter til aktiviteten, her aktivitetsnummer + dekomponeringsikon (viser at den er dekomponert)
3. Ressurslinje: lister alle ressurser som aktiviteten trenger. 2 ressurser er vist: 'clerk' - av typen *organisational actor* og 'customer application' - av typen *software tool*.
4. "Condition" – conditions brukes for å si noe om betingelsene for at en prosess kan starte eller stoppe. Conditions kan brukes til å knytte sammen prosessmodell fragmenter. Betingelsene refererer til domenet som modellen beskriver(UoD) og spesielt til informasjonsobjekt-ressurser.
5. Port signatur. Både inn og utporter i APM angis som XOR av AND porter.

APM er et språk beregnet på arbeidsflyt og skal kunne kjøres i en arbeidsflyt omgivelse (Workflow Management System). APM fokuserer dermed på betingelsene må være oppfylt for at en prosess kan starte og stoppe (pre og post kondisjoner), samt hvilke ressurser som må være tilgjengelig for at prosessen kan utføres. Start og stopp betingelsene uttrykkes logisk vha. av portsignaturer på innganger og utganger. Ressursene angis ihht. til et gitt ressurshierarki. Ressursene dekker både konkrete og generelle aktører, verktøy og objekter og kan brukes til både menneskelig og programvare basert resurstildeling. APM språket kan dekomponeres i aktivitetene, tilsvarende DFD modeller.

Oppgave 4 – Brukergrensesnitt (30%)

a) *Ta utgangspunkt i caset og lag en oppgaveanalyse for en student gjennom et helt semester. Bruk HTA-notasjonen og suppler med tekst.*

Det sentrale her er den hierarkiske strukturen og den naturlige tredeling før, under og etter et semester. Det er forsåvidt greit å gjøre 3. som en del av forberedelsene. I HTA-notasjonen beskrives sekvensbeskrankninger og betingelser som tekst på hvert nivå, men det er ikke så nøye akkurat hvor, bare det er med.



En TML-modell vil ha samme struktur, men representere sekvens og valg grafisk. Det er også naturlig å legge inn referanser til den statiske modellen, og evt. dataflyt, f.eks. fra 3.1 til 3.2.

b) Hvilke ulike representasjoner av designet er det naturlig å bruke, for å støtte dialogen med disse to gruppene? Hvilke kobling er det mellom disse representasjonene og de som brukes av de andre utviklerne, f.eks. modellene fra oppgavene 1 og 2?

Mot sluttbrukere er det naturlig å bruke uformelle representasjoner som skisser, scenarier, snapshots og storyboards. Disse er lette å forstå og fremmer dialogen. En bør unngå å lage strøkne skjermbilde-eksempler tidlig i designfasen, da disse kan binde dialogen og hindre nyttige kommentarer. Etterhvert vil prototyper være aktuelle, for å få tilbakemelding på f.eks. interaksjonsteknikker og helhet.

Mot "de andre" utviklerne kan mer formelle representasjoner brukes, da utviklerne ofte har mer trening i formelle notasjoner, og det da er lettere å sammenstille designet med spesifikasjonen/konstruksjonen av resten av systemet. Det bør være et samspill mellom systemutvikling og brukergrensesnittdesign, og i mange tilfeller er det riktig å gi designeren "rett", dersom systemspesifikasjonen ikke er konsistent med brukergrensesnittdesignet. Argumentet er typisk at "slik forstår brukeren verden, og dersom en følger systemmodellen vil brukskvaliteten/effektiviteten forringes". I andre tilfeller er systemet å oppfatte som en absolutt rammebetingelse, og det er viktig å tilpasse designet. Uansett er eventuelle forskjeller viktig å oppdaget og det er lettest dersom mer formelle modeller brukes:

1) Den statisk modell vil dekke begreper som studenten kanskje har en mental modell av, eller en oppfatning om, som det er naturlig å utnytte i brukergrensesnittdesignet. F.eks. kan begrepet "vintereksamen" være forvirrende i en landsdel hvor det kommer mer snø etter påske enn før. I vårt tilfelle har ingen eksplisitt brukergrensesnittmodell for begreper, men disse vil ofte inngå som

en naturlig del av en oppgavemodell. I tillegg vil også de konkrete representasjonene gjenspeile statisk begreper. En kan også benytte tradisjonelle modeller, men passe på at de er orientert more brukeren og validert gjennom konkrete representasjoner som skjermbilder og prototyper.

2) Prosessmodellen vil overlappe oppgavemodellen vår, f.eks. er det naturlig å finne igjen de prosessene hvor studenten er aktør, høyt oppe i oppgavehierarkiet vårt. Det er viktig å finne ut om den naturlige oppgavestrukturen for en aktør er inkonsistent med de overordnede prosessene som aktøren deltar i, både når det gjelder nedbrytning, sekvensiering og informasjonsflyt.

c) Bobby planlegger å utføre både "formative" og "summative" evalueringer i prosjektet. Hva skiller disse to typene og når er det naturlig å bruke dem i dette prosjektet?

Fra boka (Preece, Glossary: side 713 og 721 respektivt):

"*formative evaluation* : an evaluation that takes place before actual implementation and which influences the development of the product.
summative evaluation : an evaluation which takes place after implementation and has the aim of testing the proper functioning of a product."

Formative evalueringer er evalueringer som er ment å påvirke designprosessen, f.eks. ved at funnene brukes til å foreslå nye designalternativer. Summative evalueringer skal gi en kvalitativ og kvantitativ vurdering av et "sluttprodukt", og kan være nyttig for å test korrekt realisering og estimere besparelser ved bruk og evt. ressurser til hjelpetjenester. Forskjellen mellom disse to typene evalueringer påvirker naturlig nok både evalueringsteknikkene som er aktuelle og tidspunktet det er aktuelt å bruke dem.

For Bobby er det naturlig å utføre en summativ evaluering av dagens produkt, før nydesignet påbegynnes, og en etter at designet er "ferdig". Imellom disse bør Bobby bruke formative evalueringer for å vurdere designalternativer og stimulere til generering av designforslag.