UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i:	**INF 5150**	**Uangripelige IT-systemer**
Eksamensdag:	**17. desember 2003**	
Tid for eksamen:	**09.00 – 12.00**	
Oppgavesettet er på … side(r)	**4**	
Vedlegg:	**0**	
Tillatte hjelpemidler:	**Alle skriftlige dokumenter kan benyttes**	

Kontroller at oppgavesettet er komplett
før du begynner å besvare spørsmålene.

Denne "fasiten" er en redigert versjon av retteinstruksen til sensorene

Generelt ønsker vi at vi gjennomfører en indirekte evaluering, dvs. vi benytter ikke direkte A-F, men har laget et poengsystem der maksimum antall poeng er 100. Dette tilsvarer altså prosentene gitt i oppgaven. Altså skal en oppgave med verdi 40% gi maksimalt 40 poeng.

Vi regner med at vi etter poengsettingen harmoniserer våre evalueringer på denne 100 poengsskalaen før vi vurderer fordelingen av studenter over skalaen.

Til slutt setter vi grensene for A-F og avleder endelig karakter for den enkelte.

Se retteinstrukser for de enkelte oppgaver flettet inn i dette dokumentet.

Eksamen INF5150 høsten 2003 – ordinær eksamen 17. desember 2003

Den generelle kontekst for denne oppgaven er den samme som for dette kursets obligatoriske oppgave. Det dreier seg altså om en restaurant som serverer og leverer mat av ulik etnisk opprinnelse.

Det følgende er et sekvensdiagram i UML 2.0 som beskriver noen situasjoner hvor det bestilles mat over SMS (dvs. ”Short Message Service på mobiltelefon). Detaljer angående SMS er irrelevant i denne oppgaven, men poenget er at man bestiller mat asynkront fra en mobiltelefon.

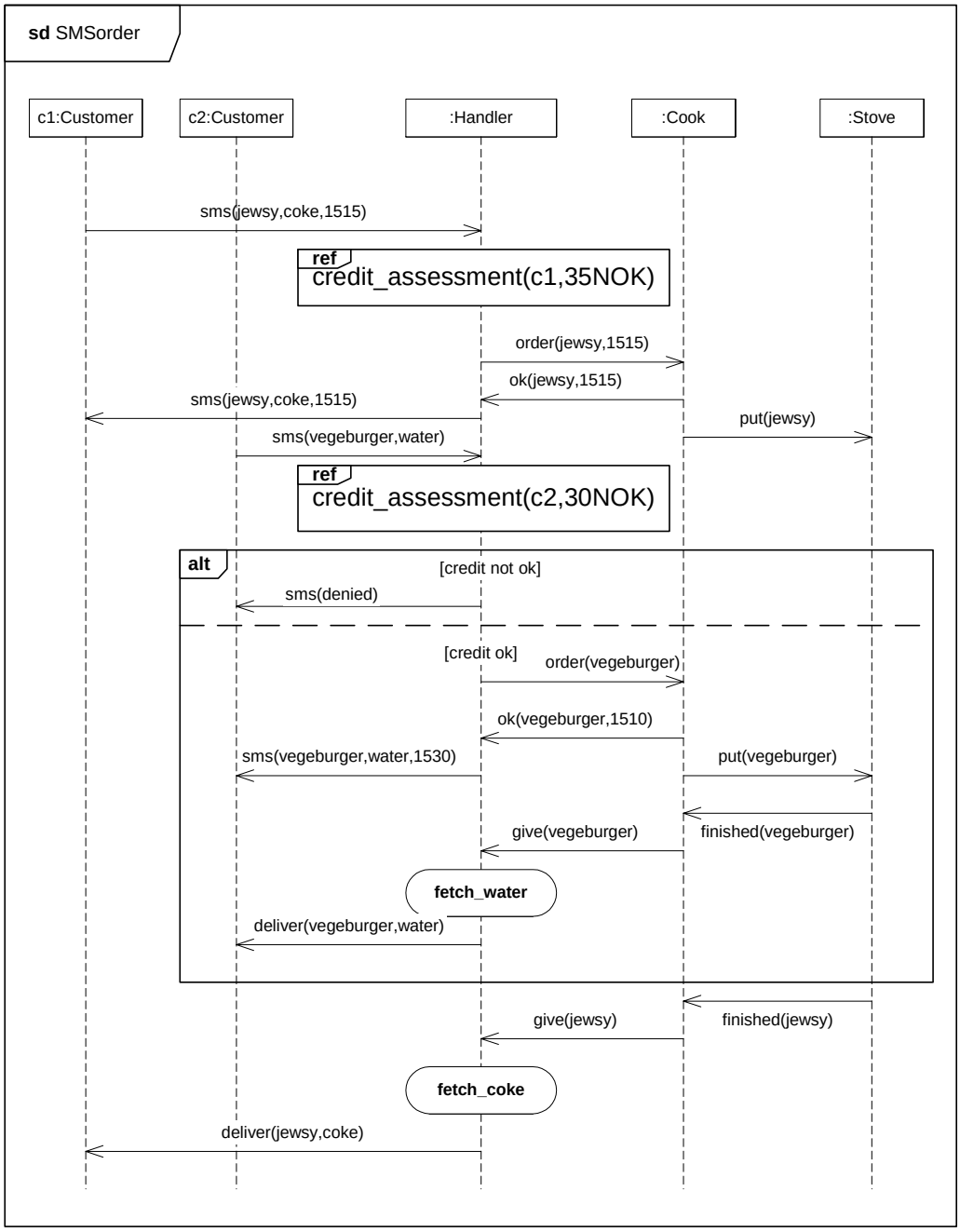


Figure 1 Sekvensdiagram for situasjoner der SMS brukes til å bestille mat

Kommentarer til sekvensdiagrammet i Figure 1.
SMS-meldingen inneholder mat, drikke og eventuelt det tidspunkt man ønsker å hente måltidet.

Hvis ikke tidspunkt er gitt, så vil man komme med en gang.
”fetch_water” og ”fetch_coke” er enkle aksjoner som Handleren gjør. Du kan bruke disse betegnelsene inne i transisjonene i tilstandsmaskinen du blir bedt om å lage.

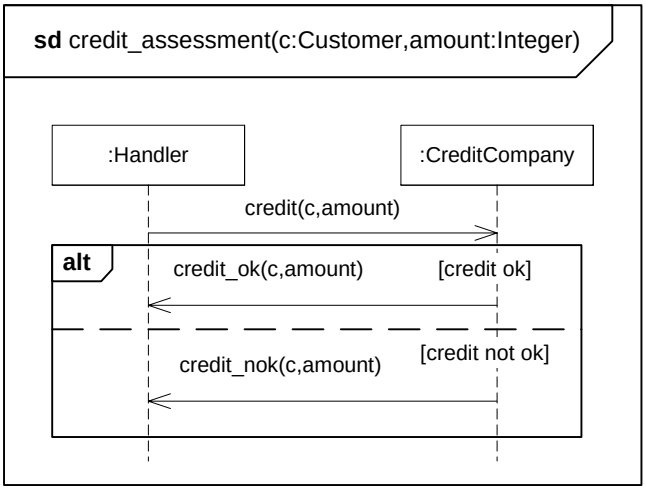


Figure 2 Veldig enkelt sekvensdiagram for credit_assessment

Oppgave 1 (40 %)

Deloppgave 1A (20 %)

Definer en tilstandsmaskin for kundebehandleren ”:Handler” slik at sekvensdiagrammet tilfredsstilles.
Vi har spesifisert credit_assessment svært enkelt i Figure 2. Lag gjerne tilstandsmaskinen slik at endringen blir godt avgrenset om credit_assessment fikk en mer komplisert spesifikasjon.

Retteinstruks 1A.

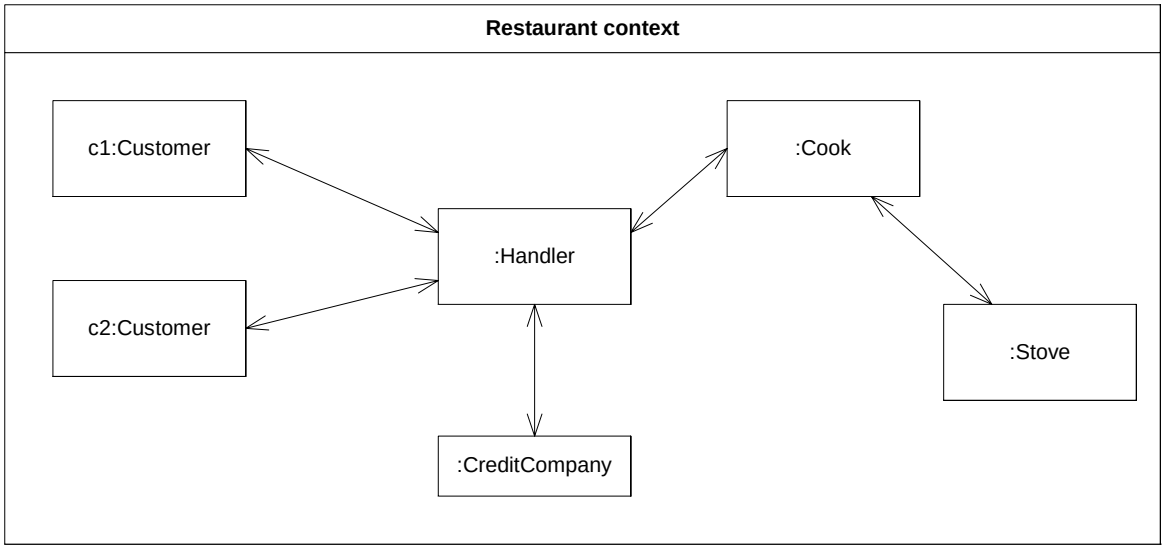
- Viktigste kriterium her er selvfølgelig at logikken i tilstandsmaskinen er korrekt. Vanskeligheten ligger i at Handleren ikke kan behandle én kunde av gangen fullstendig. Ideen er at Handleren vil kunne ta imot nye bestillinger mens maten til tidligere bestillinger blir laga.
- Elegante løsninger vil inneholde composite states.
- Spesielt bør man etter hintet i oppgaven modellere credit_assessment vha. en composite state.
- Løsningen bør selvfølgelig være såpass generell at den ikke utelukkende tilfredsstiller den gitte sekvens, men også andre liknende situasjoner

Deloppgave 1B (5 %)

Tegn en composite structure som kunne være konteksten til de gitte sekvensdiagrammer i Figure 1 og 2.

Retteinstruks 1B

- Dette bør være trivielt. Denne composite structure må altså inneholde parts som tilsvarer de Lifelines som er gitt og sammenhengen dem imellom må tilsvare den kommunikasjon som finnes i sekvensdiagrammene.



Deloppgave 1C (15 %)

I STAIRS-metodikken introduserer vi Refinement til å omfatte ”supplementing”, ”narrowing” og ”detailing”. Vi ønsker at du modifiserer sekvensdiagrammet SMSorder (med tilhørende credit_assessment) 3 ganger slik at de resulterende sekvensdiagrammer hver for seg viser resultatet av disse 3 ulike raffineringsbegreper. Du starter altså hver gang på SMSorder. Ett resulterende diagram skal være en ren "supplementing", ett skal være en ren "narrowing", og ett skal være en ren "detailing".

Retteinstruks 1C

- Poenget med supplementering er altså at det legges til sekvenser, men at ingen av de eksisterende endres. Vanskeligheten er det siste. Det er altså ikke korrekt å øke sekvensdiagrammet med ett måltid til etter de to måltidene som er angitt. Den enkleste strategien er vel å legge inn noe mer alternativer. F.eks. at det også kan skje en sms(denied) allerede etter første credit_assessment.
- Poenget ved narrowing er at noen sekvenser kuttes ut. Typisk vil det skje ved at ikke alle alternativer beholdes. En ide i vår sammenheng er at kun sms(denied)-alternativet etter andre credit_assessment beholdes. Det vil kunne tilsvare in implementasjon der kun én kunde handteres av gangen helt til leveransen av maten. Ikke effektivt, men altså en lovlig raffinering. En annen mulighet er kun å velge ett av alternativene i credit_assessment, f.eks. at kun credit ok beskrives. Dette vil tilsvare en implementasjon som betyr at alle kunder aksepteres uten kredittvurdering. En narrowing skal ikke legge til sekvenser.
- Poenget med detailing (detaljering) er at enten lifelines deles opp ved decomposition, eller at signalene deles opp ved interface refinement. Hvis de tar utgangspunkt i den obligatoriske oppgaven blir vel den vanligste løsningen å gjøre en decomposition av :Cook, men vi ser gjerne at det også resulterer i translasjon av meldinger f.eks. ved at :Cook produserer maten i flere vendinger slik vi har vist i STAIRS og at det da trengs en logisk oversetting fra disse detaljerte meldinger til den enkle meldingen som er vist i det opprinnelige diagram.
- Vi vil akseptere at de forklarer hva de ville gjøre om de ikke får tid til å lage fullstendige diagrammer, men må jo gi de som lager fullverdige diagrammer noe kreditt for det.

Oppgave 2 (40%)

Gjør en analyse av risikoer forbundet med SMS på vegne av en restaurantkunde.

Deloppgave 2A (10%)

Identifiser og verdisett minst fire aktiva. Relater disse. Spesifiser riskevalueringskriterier med hensyn til disse.

Retteinstruks oppgave 2A

Aktiva er det som skal beskyttes. Siden denne analysen gjennomføres på vegne av en restaurantkunde så skal aktiva være ting som er av verdi for en restaurantkunde i det aktuelle system. Eksempler på slike er "pengebeholdning", "helse", "personlig informasjon". Verdien av et aktivum kan være angitt kvantitativ (1000 kr) eller kvalitativt (liten).

Aktiva er ofte relatert. For å forenkle analyseprosessen er det viktig å spesifisere disse relasjonene. For eksempel, er en trussel mot kundens "tenner" (som er et aktivum for kunden) også en trussel mot kundens "helse" på grunn av den opplagte avhengigheten mellom tenner og helse. Disse relasjonene kan spesifiseres i et klassediagram eller i et CORAS asset-diagram (asset er en stereotype av UML klasse).

Riskevalueringskriteriene skal karakterisere hva man er villig til å akseptere mht risk. Det er altså en slags kravspekk som karakteriserer under hvilke betingelser behandling er nødvendig. En risk består av tre ting, nemlig en uønsket hendelse, en frekvens og en konsekvens. Frekvensen angir hvor ofte riskikoen forekommer; konsekvensen hvilket tap risikoen medfører for det aktuelle aktivum når den forekommer. Disse verdiene kan måles kvantitativt eller kvalitativt. På grunnlag av frekvens og konsekvens beregnes risikoverdi.

Eksempel på utfylt ``risk evaluation criteria table´´

Criteria ID	Applied for stakeholder	Applied for assets	Criteria description
C1	**Sykehus**	**Pasient, Helsepersonell**	**Aksepteres kun hvis *riskverdi* er mindre eller lik *liten***
...			...

I en slik tabell er første kolonne en unik identifikator. Annen kolonne angir interessent. I vårt tilfelle skal det være restaurantkunde. Tredje kolonne skal inneholde et eller flere av de identifiserte aktiva, og ingen "nye" aktiva. Fjerde kolonne skal karakterisere hva som er akseptabelt. Det kan enten gjøres i form av risikoverdi som over, men da må de ulike risikoverdiene være definert i form

av en funksjon gir en risikoverdi for et par av konsekvens og frekvens, eller direkte i form av konsekvens og frekvens.

Kvalitativ risikoanalysematrise

verditap	***sannsynlighet***			
	nesten sikkert	***sannsynlig***	***moderat***	***usannsynlig***
ignorerbart	Høy risiko	Middels risiko	Liten risiko	Liten risiko
lite	Høy risiko	Høy risiko	Middels risiko	Liten risiko
moderat	Ekstrem risiko	Høy risiko	Høy risiko	Middels risiko
stort	Ekstrem risiko	Ekstrem risiko	Høy risiko	Høy risiko
meget stort	Ekstrem risiko	Ekstrem risiko	Ekstrem risiko	Høy risiko

Deloppgave 2B (20%)

Gjennomfør en modellbasert HasOp-analyse med utgangspunkt i sekvensdiagrammet SMSorder og ledeordene (forsinket, endret, feil, misbruk). Identifiser minst 10 ulike uønskede hendelser (hvorav minst to for hvert enkelt ledeord).
HasOp-analysen skal dokumenteres i en tabell, med kolonner for: identifikator, interessent, aktiva, item, ledeord, trussel, og uønsket hendelse.

Retteinstruks oppgave 2B

Her skal det konstrueres en tabell med kolonner som angitt over. Identifikator er en unik identifikator for den aktuelle raden i tabellen. Interessent skal være "resturantkunde". Aktiva skal være et aktivum identifisert under Deloppgave 2A. "item" er en referanse til et syntaktisk element i SMSorder. Siden analysen fokuserer på risiko mht SMS så er meldinger mellom kundene og handler naturlige "items", men et "item" kan også (til nød) være en instanslinje for eksempel. Ideen er så å anvende ledeordene på "items". For eksempel, når ledeordet "forsinket" anvendes på en melding så skal man prøve å identifisere trusler og uønskede hendelser som kan oppstå fordi den aktuelle meldingen er forsinket i forhold til hvordan den er representert i SMSorder. Det kan være fra 0 til mange slike. Kolonnen for ledeord dokumenterer hvilket ledeord (forsinket, endret, feil, misbruk) som ble brukt for det par av trussel/uønsket hendelse som den aktuelle raden angir. Trussel skal være en "trussel". En uønsket hendelse oppstår ved at en trussel utnytter en sårbarhet. Den uønskede hendelsen skal kunne lede til at det aktuelle aktivum går tapt eller reduseres i verdi.

Deloppgave 2C (10%)

Organiser de identifiserte uønskede hendelsene i form av ett eller flere feiltrær. Hvis man bruker mer en ett feiltre må dette begrunnes. Tilordn kvantitative frekvensverdier til løvnodene. Kalkuler med utgangspunkt i frekvensverdiene på løvnodene resulterende frekvensverdier for rotnodene.

Retteinstruks oppgave 2C

Fire steg i en feiltreanalyse

STEG 1: Identifikasjon av topphendelse

I en sikkerhetsanalyse er dette en uønska hendelse

STEG 2: Konstruksjon av feiltre

Topphendelsen plasseres på toppen

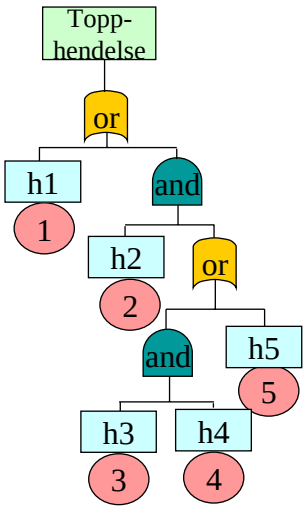
Deretter klassifiseres topphendelsen i et antall mer spesialiserte hendelser

Disse hendelsene kobles til topphendelsen gjennom en logisk port

Resten av treet konstrueres på samme måte med utgangspunkt i hver ny hendelse inntil ønsket detaljnivå nås

STEG 3: Identifikasjon av minimale snittsamlinger

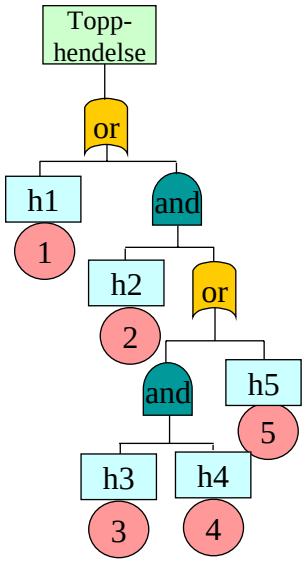
STEG 4: Kvalitativ eller kvantitativ analyse av feiltre



Kvantitativ frekvensestimering for feiltrær

- En minimal snittsamling er en minimal mengde basishendelser som er tilstrekkelig for at rothendelsen skal inntreffe
- De minimal snittsamlingene for feiltreet til høyre er {1}, {2,5}, {2,3,4}
- Hvis alle hendelser er uavhengige er sannsynligheten for en minste snittsamling lik produktet av sannsynlighetene for dens basishendelser
- For et feiltre med n minimale snittsamlinger med sannsynligheter p1, p2, … pn er sannsynligheten for rothendelsen

  1-((1-p1)*(1-p2)*….*(1-pn))



Symboler i et feiltre

og: Alle hendelser under porten må inntreffe for at henselsen over porten skal inntreffe

eller: Minst en av hendelsene under porten må inntreffe for at hendelsen over porten skal inntreffe

hendelse: Hendelse som resulterer fra en kombinasjon av hendelser gjennom en logisk port

basishendelse: Hendelse som ikke krever videre dekomponering

uferdig hendelse: Hendelse som ikke er ført tilbake til sin årsak; tas som input men årsak kan være ukjent

fortsettelse i annet tre: Indikerer at resten av treet fortsetter et annet sted

fortsettelse av et annet tre: Indikerer at treet er en fortsettelse av et annet tre

Her er vi ikke nøye på syntaksen. Det essensielle er å få frem hvordan løvhendelsene relaterer seg til rothendelsene ved hjelp av "og" og "eller" porter.

Oppgave 3 (20 %)

I Figure 3 viser vi en tilstandsmaskin som modellerer kokken ":Cook". Vurdér om denne tilstandsmaskinen er en god implementasjon relativt til sekvensdiagrammet SMSorder. Hvordan ville du evt. forbedre den?

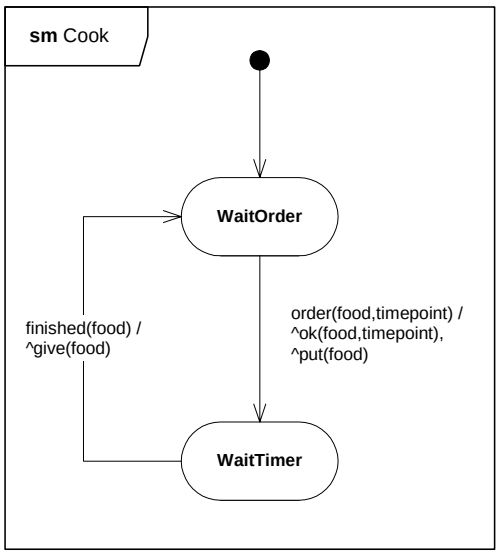


Figure 3 En tilstandsmaskin for Cook

Kommentarer til Figure 3:.
Vi har med vilje utelatt deklarasjoner av parametere og lokale variabler. Du behøver heller ikke bekymre deg om det, men det er ønskelig at du kommenterer evt. nye begreper og betegnelser som du introduserer i et forsøk på å lage en forbedring.
Den lille hatten foran signaler betyr kun en meldingssending.

Retteinstruks Oppgave 3

- Det er meningen at det skal utføres en partiell modelchecking ved å sjekke :Cook linja i sekvensdiagrammet mot tilstandsmaskinen. Det vil altså skjære seg ved andre ’order’ signal fordi da er :Cook i WaitTimer og får altså et signal den ikke har definert.
- Det er ikke så lett å fikse. Vi ønsker at også :Cook skal kunne handtere flere ordrer samtidig og det vil altså bety at han må på et eller annet vis bufre ordrene mens han venter/lager maten. Vi kan tenke oss en liste i data som representerer dette bufferet. Men for å få kun vårt eksempel til å gå trengs jo tross alt ikke mer enn 2 elementer i databufferet.
- En litt mer kreativ løsning er å la :Cook generere assistenter ad lib. Hver assistent tar seg av en ordre etter omtrent den tilstandsmaskinen som er gitt. Som avslutning på ordren, terminerer assistenten. Dette er en løsning som går teknisk og er elegant, men som faktisk implementasjon i restauranten ville sikkert søknadene til å bli assistent bli få hvis assistentene ble terminert etter oppfylt første ordre (:-).
- Oppgaven evalueres rimeligvis etter generalitet, korrekthet og kreativitet (og kanskje forståelighet) prioritert etter den rekkefølgen.