

Norges teknisk-naturvitenskapelige universitet
Institutt for telematikk



EKSAMENSOPPGAVE I TTM4115 – SYSTEMERING AV DISTRIBUERTE SANNTIDSSYSTEMER

Faglig kontakt under eksamen: Rolv Bræk

Tlf.: 415 44 605

Eksamensdato: 29. mai 2009

Eksamenstid: 09:00-13:00

Studiepoeng: 7,5 SP

Tillatte hjelpemidler: A: Alle trykte og håndskrevne hjelpemidler tillatt. Alle kalkulatorer tillatt

Språkform:

Antall sider bokmål: 1

Antall sider engelsk: 1

Antall sider nynorsk: 1

Antall sider vedlegg: 3

Sensurdato¹: 23. juni 2007

¹ Merk! Studentene må primært gjøre seg kjent med sensur ved å oppsøke sensuoppslagene.

Bokmål (Eksamen utgjør 75% av sluttkarakteren.)

Noen av oppgavene referer seg til systemene som er beskrevet i vedlegg. Studer vedlegget først.

Oppgave 1. (25%) SDL

1. Gjør UML diagrammet i Figur 1 om til et SDL system diagram som tilsvarer Figur 1 så langt som mulig. Ta med definisjon av de signalene som fremgår av Figur 2.
2. Foreslå en indre struktur i *AcessAgent*, *GuideAgent* og *POIAgent* og begrunn valget. Det forutsettes at oppførselen til hver enkelt sesjon skal beskrives tilstandsorientert.
3. Definer oppførselen til *aat* rollen som en composite state i SDL-2000 slik at den tilfredsstiller sekvensdiagrammet i Figur 2 og har et *exit point* som kalles *accept* og ett som kalles *reject* som tilsvarer hhv akseptert og avvist bruker. (Bruk *none* input til å trigge transisjoner som skyldes input på grensesnitt som ikke er vist.)
4. Definer oppførselen til *cgt* rollen som en composite state i SDL-2000 slik at den tilfredsstiller sekvensdiagrammet i Figure 3.
5. Kontroller at rollen for *cgt* som du laget under punkt 4 er input konsistent. Gjør de endringene som eventuelt er nødvendige for at den skal bli input konsistent.

Oppgave 2. (25%) Systemdesign

1. Gjør rede for hvilke alternativ som finnes for å adressere og rute signaler i SDL. Hvilke alternativer egner seg i kollaborasjonene i Figur 2 og 3?
2. Definer en generell kollaborasjon som kan benyttes for opprettelse av sesjoner alle steder det er behov for dette i *CityGuide* systemet. Tegn MSC for typiske forløp.
3. Legg til collaboration uses i Figure 1 som viser hvor kollaborasjonen du definerte under punkt 2 over skal brukes.
4. Lag en SDL prosessgraf som viser oppførselen til *ut:PDA* fra og med opprettelse av *AccessSession* frem til *GuideSession* er opprettet. Bruk svarene på oppgave 1 punkt 3 og 4 som tilstander. (Hvis du ikke har løst disse, anta at de er tilgjengelig og vis hvordan de brukes)

Oppgave 3. (25%) Diverse

Gitt en omgivelse E og to systemer $S1$ og $S2$ beskrevet med prosessalgebra, CCS:

$$E = \tau; a'; ac; E + b; bc'; E$$

$$S1 = a; ac'; S1 + b'; bc; S1$$

$$S2 = a; ac'; S2 + \tau; b'; bc; S2$$

1. Foreta ekspansjon av uttrykket $E // S1$ og $E // S2$
2. Oppstår vranglås eller andre feil i noen av uttrykkene du fant (på punkt 1 over)?
3. Hva kan sies om ekvivalens mellom $S1$ og $S2$? Kan omgivelsen se forskjell? Er de observasjonsekvivalente?
4. Forklar hvordan SDL *save* virker. Hva blir slutttilstanden dersom prosessen i figur 4 starter i tilstand 1 med signalene $d;f;e;a;b$ (der d er først) i input køen?
5. Det antas at *CityGuide* systemet skal huske profilen og historien til registrerte brukere mellom hver innlogging. Forklar hvordan du vil ivareta dette.

English (The exam counts 75% towards the final grade.)

Some of the questions refer to the systems described in the appendix. Study the appendix first.

Question 1. (25%) SDL

1. Make an SDL system diagram that corresponds as closely as possible to the UML diagram in Figure 1. Include definitions of the signals that are used in Figure 2.
2. Propose inner structures for the *AcessAgent*, *GuideAgent* and *POIAgent* and justify your choices. Assume that the behaviour of each session is described in state oriented fashion.
3. Define the behaviour of the *aat* role as a composite state in SDL-2000 so that it satisfies the sequence diagram given in Figure 2 and has an *exit point* called *accept* and one called *reject* corresponding to accepted and rejected user respectively. (Use the *none* input to trigger transitions caused by input on interfaces that are not shown.)
4. Define the behaviour of the *cgt* role as a composite state in SDL-2000 so that it satisfy the sequence diagram in Figure 3.
5. Check that the *cgt* role you made under point 4 above is input consistent. If not, make it input consistent.

Question 2. (25%) System design

1. What are the alternative ways that SDL signals may be addressed and routed? Which alternatives are suitable in the collaborations described in Figure 2 and 3?
2. Define a general collaboration that can be used to establish sessions wherever needed in *CityGuide* system. Make MSC to describe typical cases.
3. Add collaboration uses in Figure 1 to show where the collaboration you defined under point 2 above is to be used.
4. Make an SDL-2000 process graph describing the initial behaviour of *ut:PDA* starting with establishing the *AccessSession* and ending with the *GuideSession* being established. Use the behaviours you defined under Question 1 point 3 and 4 as states. (In case you have not answered these, just assume they have been made and show how they are used.)

Question 3. (25%) Miscellaneous

Given an environment E and two systems $S1$ and $S2$ defined using process algebra, CCS:

$$E = \tau; a'; ac; E + b; bc'; E$$

$$S1 = a; ac'; S1 + b'; bc; S1$$

$$S2 = a; ac'; S2 + \tau; b'; bc; S2$$

1. Expand the expressions $E // S1$ and $E // S2$.
2. Are there deadlocks or other errors in any of the expanded expressions you developed under point 1 above?
3. What can be said about equivalence between $S1$ and $S2$? Can the environment see any difference? Are they observation equivalent?
4. Explain how SDL *save* works. What will be the final state if the process in Figure 4 is started in state 1 with the signals $d;f;e;a;b$ (where d is first) in the input queue?
5. Assume that the *CityGuide* system shall remember the profile and history of users when they are not logged in. Explain how this may be achieved.

Nynorsk (Eksamen utgjør 75% av sluttkarakteren.)

Nokre av oppgåvene referer seg til system som er skildra i vedlegg. Studer vedlegget fyrst.

Oppgåve 1. (25%) SDL

1. Gjer UML diagrammet i Figur 1 om til eit SDL system diagram som tilsvarar Figur 1 så langt som mogleg. Ta med definisjon av dei signala som er med i Figur 2.
2. Foreslå ein indre struktur i *AcessAgent*, *GuideAgent* og *POIAgent* og grunnlegg valet. Det er gjeve at oppførselen til kvar enkelt sesjon skal skildrast tilstandsorientert.
3. Definer oppførselen til *aat* rollen som ein composite state i SDL-2000 slik at den tilsvarar sekvensdiagrammet i Figur 2 og har eit *exit point* som heiter *accept* og eit som heiter *reject* som tilsvarar hhv akseptert og avvist brukar. (Bruk *none* input til å trigge transisjonar grunna input på grensesnitt som ikkje er synt.)
4. Definer oppførselen til *cgt* rollen som ein composite state i SDL-2000 slik at den tilsvarar sekvensdiagrammet i Figur 3.
5. Kontroller at rollen for *cgt* som du laga under punkt 4 er input konsistent. Gjer dei endringane som eventuelt er naudsynte for at den skal verte inputkonsistent.

Oppgåve 2. (25%) System design

1. Gjer greie for kva alternativ som finns for å adressere og rute signal i SDL. Kva alternativ eignar seg i kollaborasjonene i Figur 2 og 3?
2. Definer ein generell kollaborasjon som kan nyttast for oppretting av sesjonar alle stader det er behov for dette i *CityGuide* systemet. Teikn MSC for typiske tilfelle.
3. Legg til collaboration uses i Figur 1 som syner kor kollaborasjonen du definerte under punkt 2 over skal nyttast.
4. Lag ein SDL prosessgraf som syner oppførselen til *ut:PDA* frå og med oppretting av *AccessSession* fram til *GuideSession* er oppretta. Bruk svara på oppgåve 1 punkt 3 og 4 som tilstandar. (Dersom du ikkje har laga desse, gå ut frå at dei er tilgjengelege og syn korleis dei nyttast.)

Oppgåve 3. (25%) Diverse

Gjeve ein omgjevnad E og to system $S1$ og $S2$ skildra med prosessalgebra, CCS:

$$E = \tau; a'; ac; E + b; bc'; E$$

$$S1 = a; ac'; S1 + b'; bc; S1$$

$$S2 = a; ac'; S2 + \tau; b'; bc; S2$$

1. Gjer ekspansjon av uttrykket $E // S1$ og $E // S2$.
2. Oppstår vranglås eller andre feil i nokon av uttrykka du fant (på punkt 1 over)?
3. Kva kan seiast om ekvivalens mellom $S1$ og $S2$? Kan omgjevnaden sjå skilnad? Er dei observasjonsekvivalente?
4. Forklar verkemåten til SDL *save*. Kva vert slutttilstanden dersom prosessen i figur 4 startar i tilstand 1 med signala $d;f;e;a;b$ (der d er først) i input køa?
5. Gå ut frå at *CityGuide* systemet skal huske profilen og historia til registrerte brukarar mellom kvar innlogging. Forklar korleis du vil ivareta dette.

Vedlegg/ Appendix

Bokmål

CityGuide systemet

Et system for turistguiding i byen, *CityGuide*, er vist med UML i Figure 1:

- Hver bruker har en *PDA* (eller *smartphone*) med GPS mottaker.
- *AccessAgent* benyttes til registrering av nye brukere og pålogging av eksisterende brukere i henhold til kollaborasjonen *AccessSession*, beskrevet i Figure 2.
- *GuideAgent* står for guiding på overordnet (by) nivå i henhold til kollaborasjonen *GuideSession*, beskrevet i Figure 3.
- *POIAgent* står for detaljert guiding i forbindelse med en severdighet (POI) i henhold til kollaborasjonen *POISession* (ikke detaljert).

Hver bruker har en profil og en historie som brukes av *GuideAgent* til å velge hvilke POI som presenteres. For å forenkle diagrammene er disse og andre data utelatt.

CityGuide systemet skal kunne betjene mange brukere samtidig slik at det vil være mange parallelle forløp av kollaborasjonene i Figur 1. De forskjellige kollaborasjonene i Figur 1 blir startet og avsluttet dynamisk.

I tillegg til kollaborasjonene vist i Figur 1 finnes det kollaborasjoner for å opprette og avslutte sesjoner, som ikke er vist.

English

The Cityguide system

A system for tourist guiding in the city, *CityGuide* is depicted using UML in Figure 1:

- Each user has a *PDA* (or *smartphone*) with GPS receiver.
- *AccessAgent* is used to register new users and to log on existing users according to the collaboration *AccessSession*, described in Figure 2.
- *GuideAgent* is responsible for guiding at the city level according to the collaboration *GuideSession*, described in Figure 3.
- *POIAgent* is responsible for detailed guiding around a point of interest (POI) according to the collaboration *POISession* (not detailed).

Each user has a profile and a history that the *GuideAgent* uses to select which POI to present to the user. In order to simplify the diagrams, this and other data has been omitted.

The *CityGuide* system shall be able to serve many users concurrently so there will be many occurrences of the collaborations in Figure 1 running concurrently. The various collaboration occurrences are started and stopped dynamically.

In addition to the collaborations shown in Figure 1 there are collaborations to start and stop sessions that are not shown.

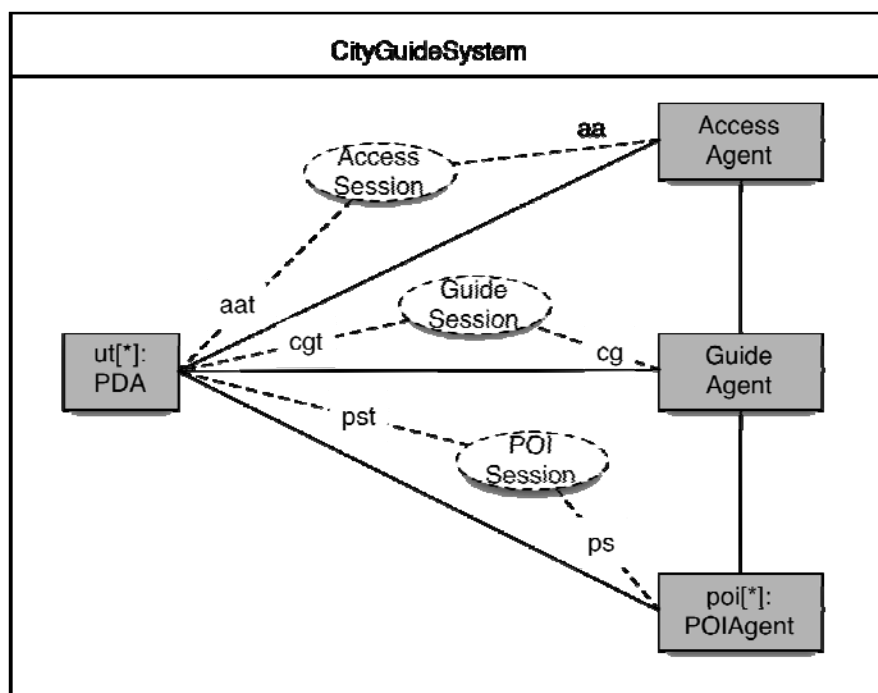


Figure 1 The CityGuide system

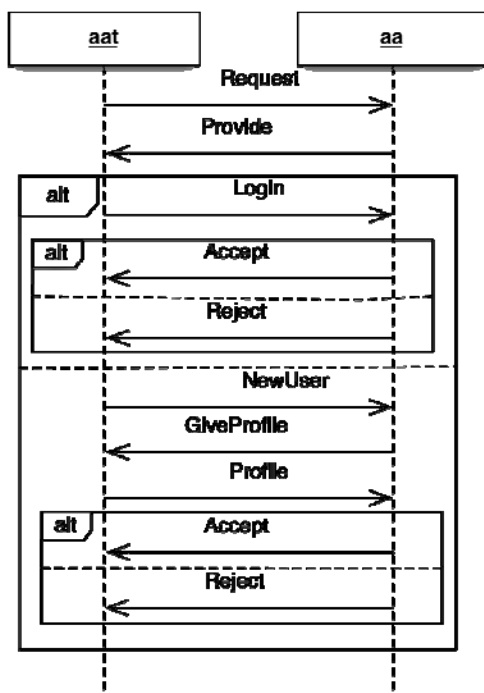
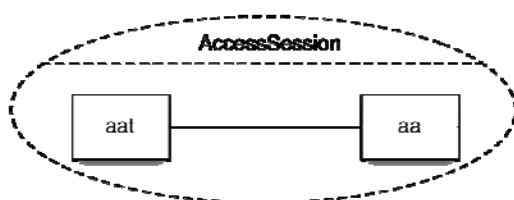


Figure 2 Access Session

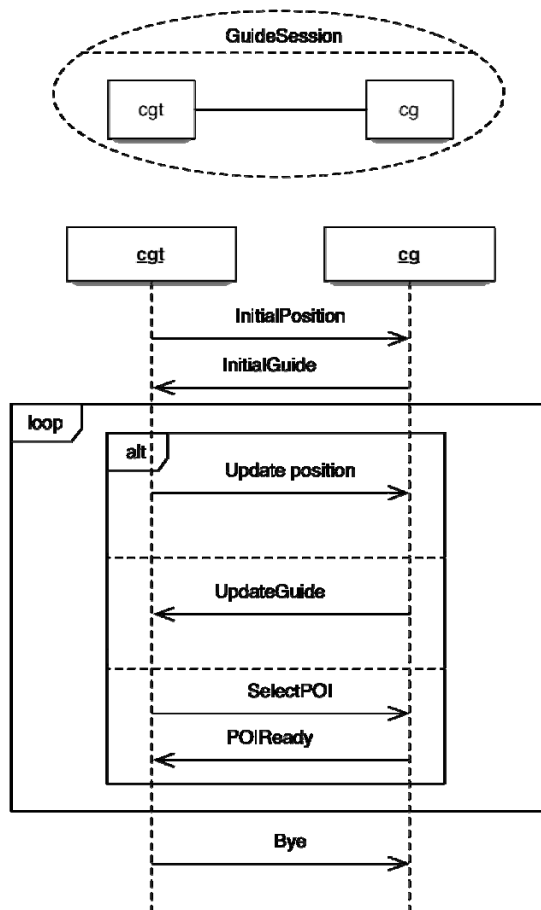


Figure 3 GuideSession

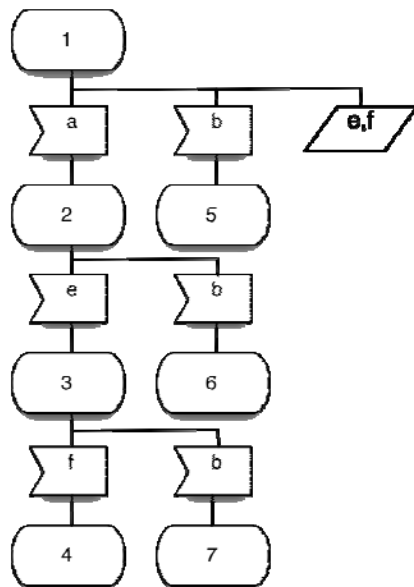
process P

Figure 4 Process P