

Norges teknisk-naturvitenskapelige universitet  
Institutt for telematikk



**Løsningsforslag**  
**EKSAMENSOPPGAVE I TTM4115 – SYSTEMERING AV**  
**DISTRIBUERTE SANNTIDSSYSTEMER**

<b>Faglig kontakt under eksamen:</b>	Rolv Bræk
<b>Tlf.:</b>	415 44 605
<b>Eksamensdato:</b>	29. mai 2009
<b>Eksamenstid:</b>	09:00-13:00
<b>Studiepoeng:</b>	7,5 SP
<b>Tillatte hjelpemidler:</b>	A: Alle trykte og håndskrevne hjelpemidler tillatt. Alle kalkulatorer tillatt
<b>Språkform:</b>	
<b>Antall sider bokmål:</b>	1
<b>Antall sider engelsk:</b>	1
<b>Antall sider nynorsk:</b>	1
<b>Antall sider vedlegg:</b>	3
<b>Sensurdato<sup>1</sup>:</b>	<b>23. juni 2007</b>

---

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

## Bokmål (Eksamen utgjør 75% av slutt karakteren.)

Løsningene er gitt i den engelske delen

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 tilsvare 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 tilsvare 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:

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

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

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

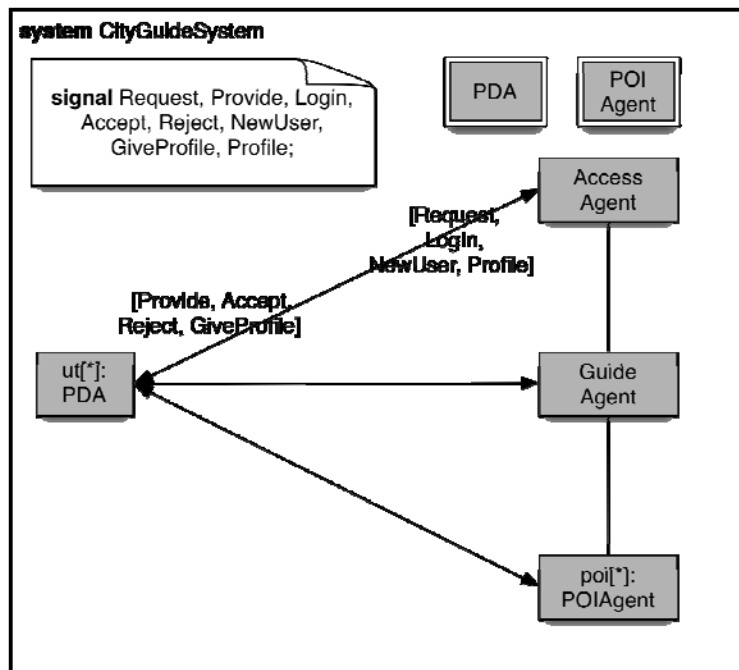
1. Foreta ekspansjon av uttrykket  $E // S1$  og  $E // S2$
2. Oppstår vranglås eller andre feil i noen av uttrykkene du fant (på punkt 1 over)?
3. Hva kan sies om ekvivalens mellom  $S1$  og  $S2$ ? Kan omgivelsen se forskjell? Er de observasjonsekvivalente?
4. Forklar hvordan SDL *save* virker. Hva 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.



The SDL system has the same inner structure as the given UML Class. In addition comes signal definitions and signals on the channels, as well as block type references.

SDL syntax errors like missing frame etc: -1-2.

Missing signals on channels and signal defs: -2

Missing signals on channel or signal defs: -1

Missing channels: -2

Missing type refs: -1

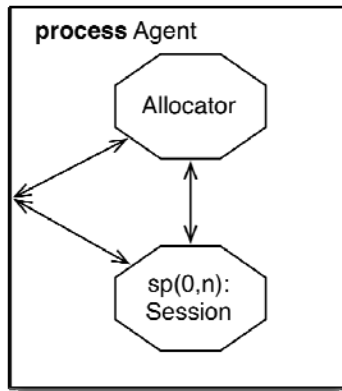
PDA in environment: -1

Missing cardinality: -1

Using stereotypes `<<block>>` on the blocks is incorrect (only used on UML classes instead of type references): -1

Package instead of system: -1

2. Propose inner structures for the *AccessAgent*, *GuideAgent* and *POIAgent* and justify your choices. Assume that the behaviour of each session is described in state oriented fashion.



Above is a general pattern that may be used for all of them. Since the sessions are to be modelled state oriented one will need a process for each session, and an allocator to manage requests for session setup and session release. One may either use a fixed pool of session processes that are dynamically allocated or one may create and stop session processes dynamically.

In the AccessAgent one might also have a registry holding user information.

In the CityGuide agent one needs a registry over POIs and other information.

Missing registries: -0,5

Missing multiplicity on sessions: -3

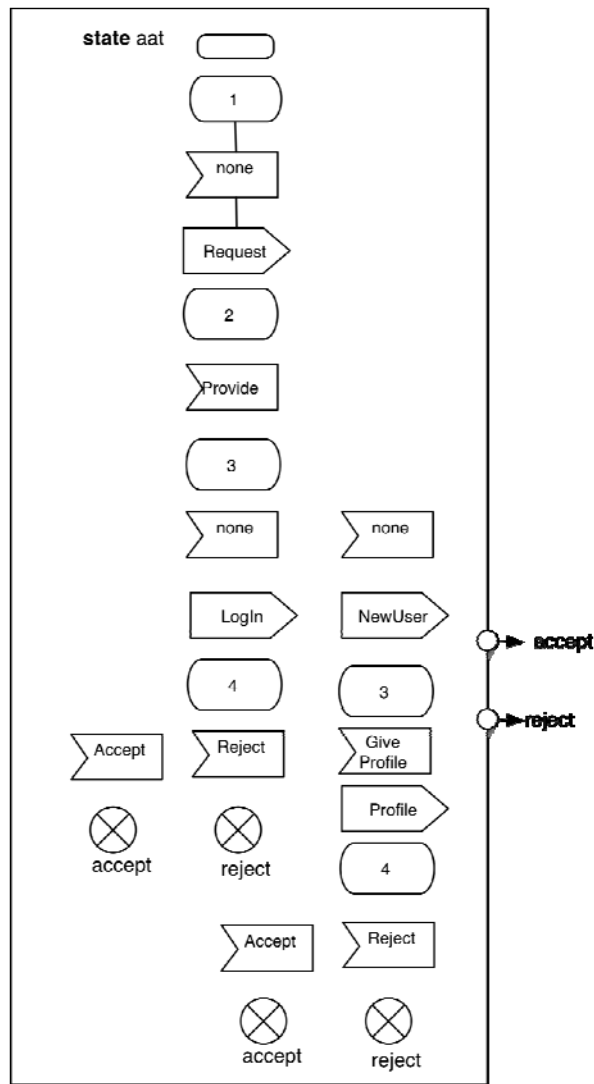
Missing allocator/manager: -3

SDL syntax: -1

Decomposition not according to method/Function decomposition: -1-2

Unnecessary processes for connection between blocks: -0.5-1

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.)



*Transition lines to be added above. Important here is to define it as a state with exit labels. Also to use none properly. In stead of starting with a “none” it may be better to send the “Request” on the initial transition.*

*SDL syntax errors like missing frame: -1-2*

*Missing none: -1*

*Inputs without states: -2*

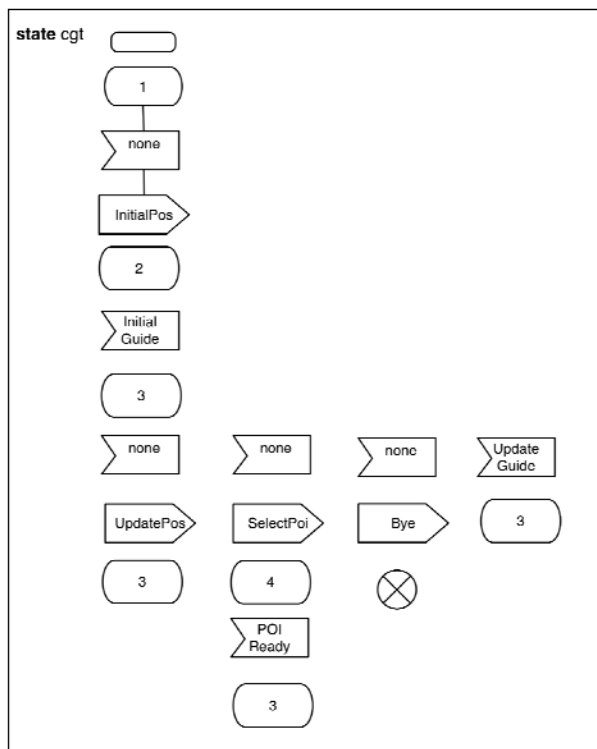
*Missing exit labels: -2*

*Process instead of composite state: -2*

*Not compliant with MSC/ Missing inputs or outputs: -1*

*Missing exit gates: -1*

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.



Transition lines to be added here. Note that no exit labels are needed since there is only one exit.

SDL syntax errors like missing frame, missing start/exit: -1-2

Missing none: -1

Inputs without states/omitted states: -2

Behaviour errors/not logical behaviour: -2

5. Check that the *cgt* role you made under point 4 above is input consistent. If not, make it input consistent.

Transition 3-4 is not input consistent. Must receive *UpdateGuide* in state 4. Note that consistency of the exit transition depends on the next state in the diagram where the composite state is used. This state should also be prepared to receive *UpdateGuide*.

Finner ikke inkonsistens grunnet utelatte tilstander (SDL feil), ellers rett resonnement: -3

Dårlige eller uriktige forklaringer: -1-3

Behandler none som et synlig signal: -2

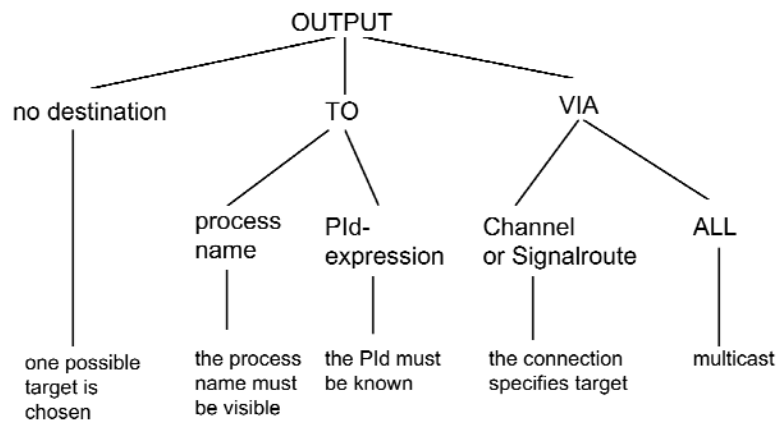
Helgardering ved å legge til for mange signaler/ mange *updateGuide* fra same tilstand:-3

Not pointing to the precise error or not saying how to correct it: -5

## 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?

Alternatives are listed in foil 26 of *Basic Sdl*.



*TO* addressing is generally needed here since sessions are created dynamically and one needs to address the individual session processes. Note that the collaborations in fig 2 and 3 only describe the behavior after sessions has been established. For session setup one may use *Via* towards the allocator.

Omitted one kind: -1,2 (1 for no destination, 2 for *To* or *VIA*)

Has to consider the solution proposed in Q1.2 when assessing which method to use. If correct given the solution, then: -0

Not explained what to use: -5

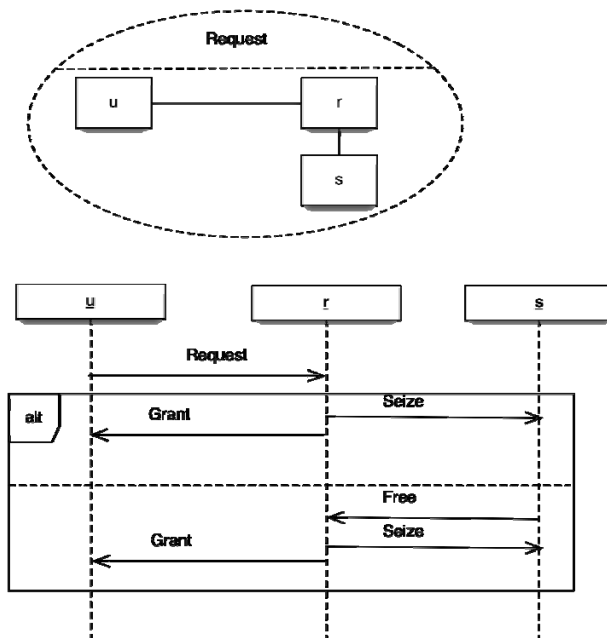
Not explained any addressing schemes: -5

Not used *TO* towards PDA: -2

Indirect routing, remote routing is not asked for, but gives credit if nothing else: +2

- Define a general collaboration that can be used to establish sessions wherever needed in *CityGuide* system. Make MSC to describe typical cases.

Something similar to *RoleRequest* in *ActorFrame*, or the solution to exam 2008.



In the above solution UML Sequence diagrams are used, but we asked for MSC so the candidates should use the MSC syntax. UML is acceptable however.

This solution assumes that the session processes are created at startup and then allocated dynamically. Creating the sessions (s) dynamically would be equally correct. The only problem is to handle any persistent data specific to a session object. One does not really need

to consider the r-s interactions here, so it is OK to omit the s role and only show the u-r interactions.

Omitting the collaboration: -2

MSC errors like missing frame, missing labels etc: -1

Very incomplete MSC: -1

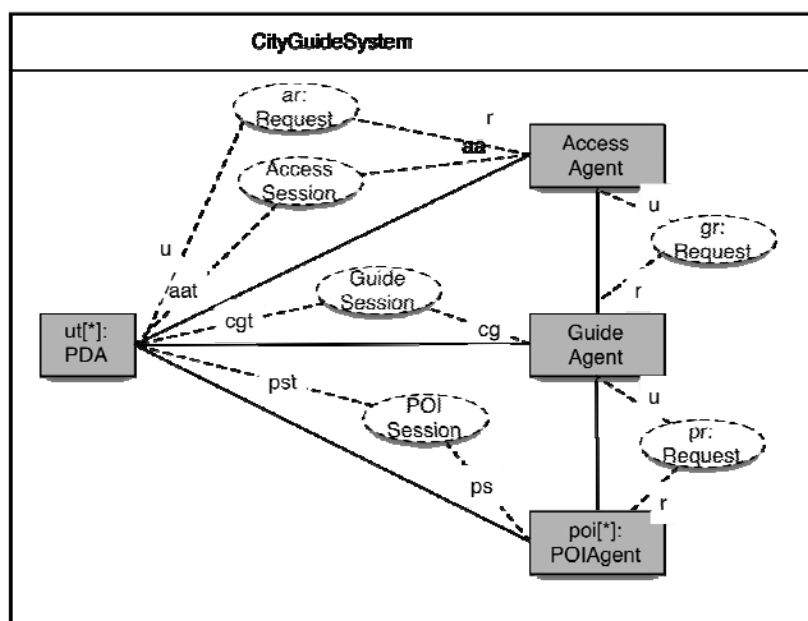
Solving the wrong problem, not session allocation: -5

Technically flawed solution: -2

Not general collab: -1

If they demonstrate here that they now more about sessions and allocators than they showed in Q1.2, give corresponding credit to Q1.2

3. Add collaboration uses in Figure 1 to show where the collaboration you defined under point 2 above is to be used.



It is assumed above that the access session makes the request for the Guide session and that the guide session requests the POI session. Alternativley all requests can be made by the PDA. This is equally correct.

Changing the structure: -2

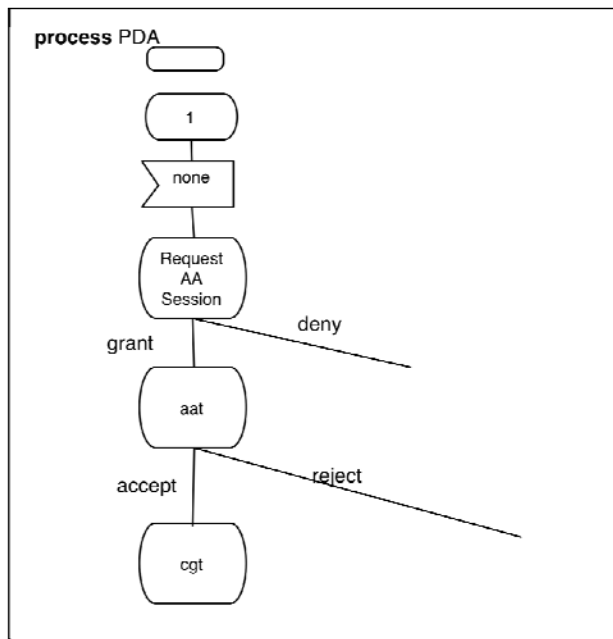
Missing collaboration use: -1

Technically flawed solution:-1-2

Removing existing Collaboration uses:-2

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.)





The first none is needed to initiate the request. The first transition in aat may be a continuation of the “grant” transition in the figure above in stead of being triggered by a “none”. Important here is to use composite states correctly, and to take care of the session request. Here we assume that session request for AA is done by the PDA, while session request for the city guide is done by the aa session. If the student has indicated that the PDA should do this, then a session request for the CG session should be added to the process graph above.

Not used composite states from Q1: -3

Not handled request for AA session: -3

SDL errors: no penalty here if given earlier under Q1. Else as for Q1.

Logical errors in diagram: -1

### Question 3. (25%) Miscellaneous

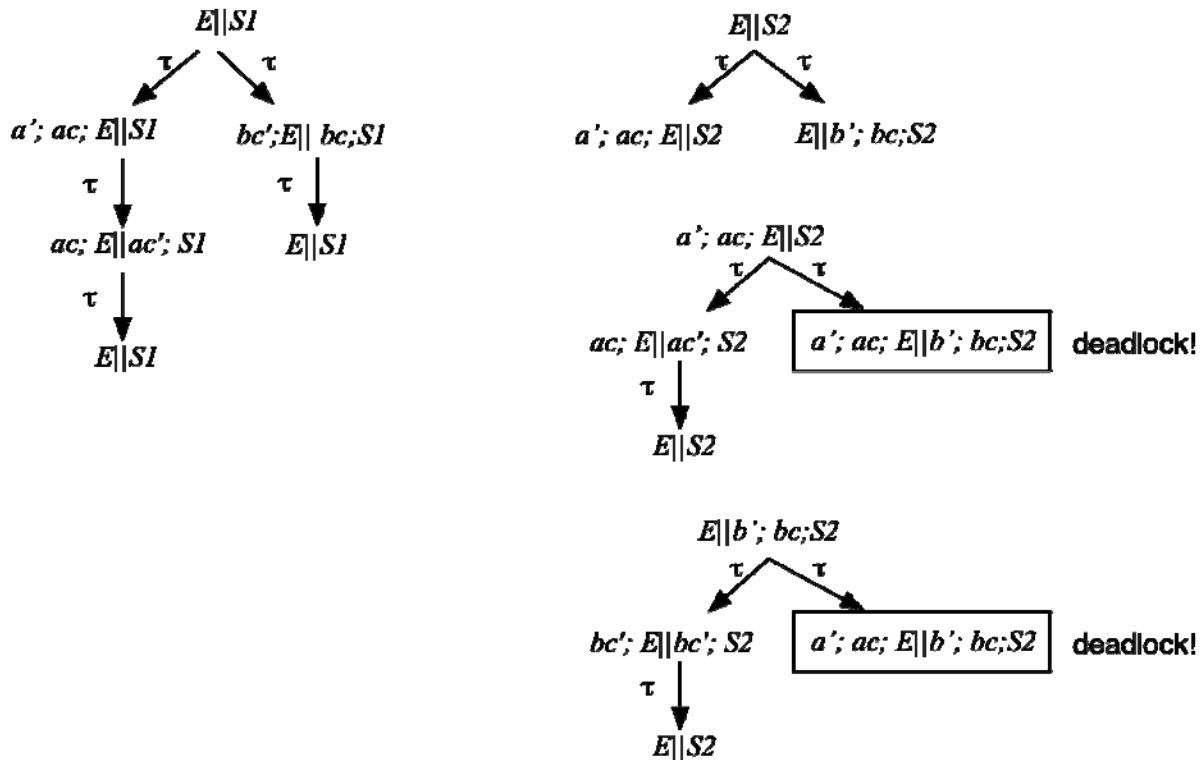
Given an environment  $E$  and two systems  $S1$  and  $S2$  defined using process algebra, CCS:

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

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

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

1. Expand the expressions  $E // S1$  and  $E // S2$ .



The most important here is to demonstrate understanding of the basic expansion rules.

Minor errors in expressions: -1

Not full expansion, but correct start: -1

Omitting/forgetting transitions/adding transitions: -2

Wrongly doing two steps at the time: -2

Wrong transitions: -2

Wrong expressions after transitions: -2

Done only one of them: -4

Completely wrong expansion (e.g just listing all pairs of terms): -10

- Are there deadlocks or other errors in any of the expanded expressions you developed under point 1 above?

We have two deadlocks in  $E||S2$ .

Not finding deadlocks that are present in own answer to 1 above: -10

Correctly not finding deadlocks if not present in own answer to 1 above: -0

Treating only one of the expressions: -2

Claiming deadlock without having found it in Q1: -7

Finding deadlocks that are not deadlocks: -7

- What can be said about equivalence between  $S1$  and  $S2$ ? Can the environment see any difference? Are they observation equivalent?

The environment will experience deadlock with  $S2$ , but not with  $S1$ . Therefore the environment will observe a difference. The two processes are not observation equivalent.

Claiming the environment will see no difference if deadlock found in above: -7

Mistake about observation equivalence: -3

Correct statement about trace equivalence and/or Strong bisimulation: +3

Not answering the question about the environment seeing any difference: -5

But if the answer about OE is correct, then: -3

*Mention the  $P \approx \tau;P$  as reason for being O.E. is understandable: +2*

*(but not correct because it does not hold under summation:  $P = \tau;P1 + \tau;P2$  is not OE to  $P1+P2$ . However the students may easily have overlooked this. They should however, understand that when there is deadlock in one and not the other then they will behave differently seen from environment.)*

*Answering about  $E//S1$  and  $E//S2$  in stead of  $S1$  and  $S2$ : -2*

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?

*See the textbook page xx. The final state will be 7. The reason is that the "f" is discarded in state 2.*

*Right explanation wrong final state: -5*

*Right final state wrong explanation: -5*

*If the reasoning is right but clearly just did a small mistake (e.g. forgot to consume f in 2) leading to wrong final state: -3*

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.

*One need to have some user objects that are persistent and can store the information. This could be a persistent user agent or it can be a record in a database that is fetched when the user logs on and stored when the user logs off. A persistent user agent would also need a database or similar to store the profile as backup. The agent is persistent if it remains in the system as long as the user is defined in the system, i.e. not created dynamically when the user logs on. The PDA might do this, but then there would be no backup.*

*The main thing here is to identify the need for either persistent data or persistent processes, missing to do so: -5*

*Unclear ideas: -1-3*

*Using the PDA without backup: -3*

*How to store and retrieve not mentioned: -3*

## 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 grunnjev 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.

**Oppgave 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.)

**Oppgave 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.

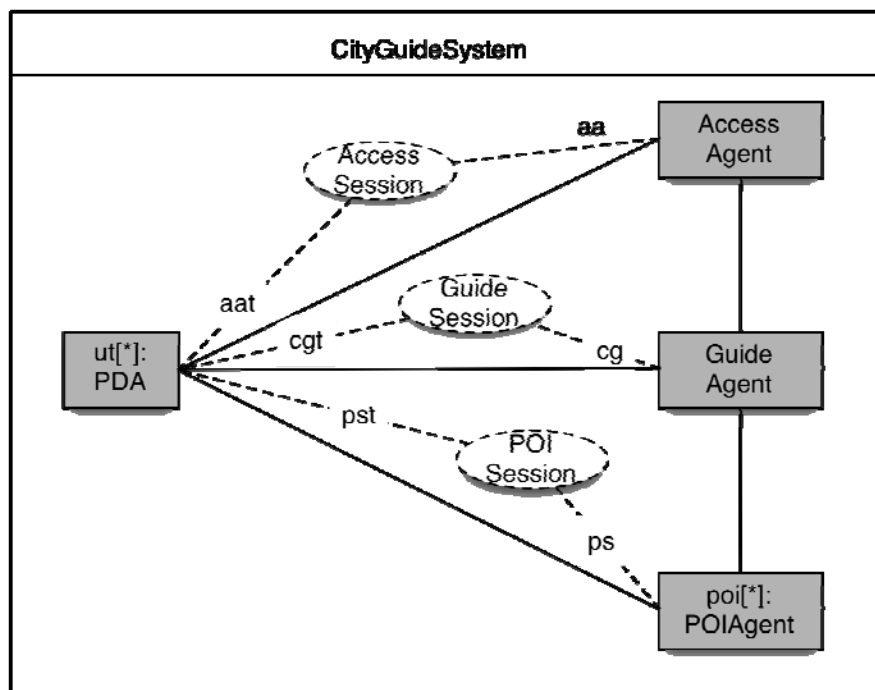


Figure 1 The CityGuide system

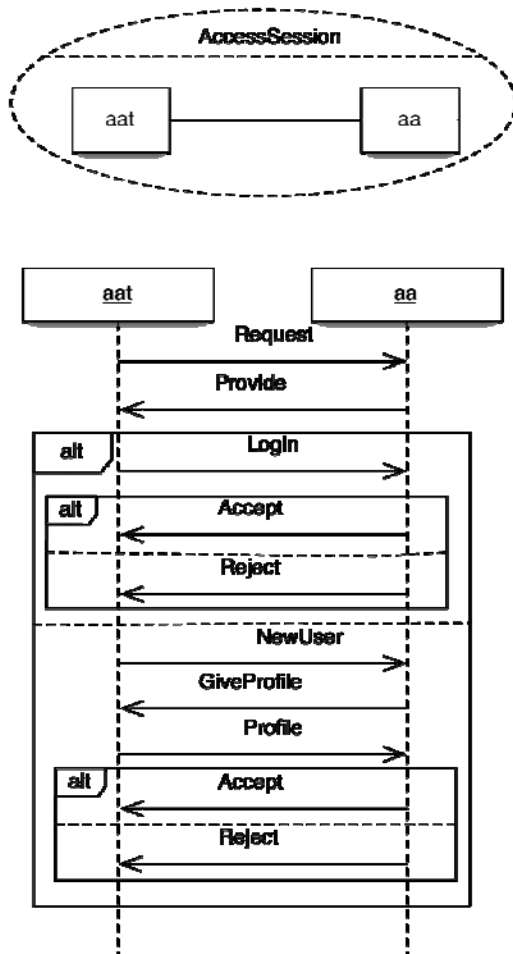


Figure 2 Access Session

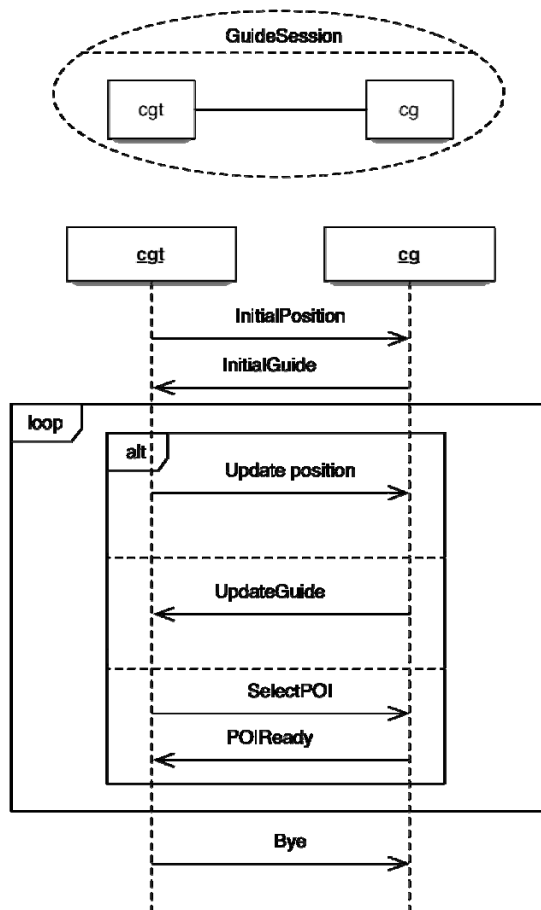


Figure 3 GuideSession

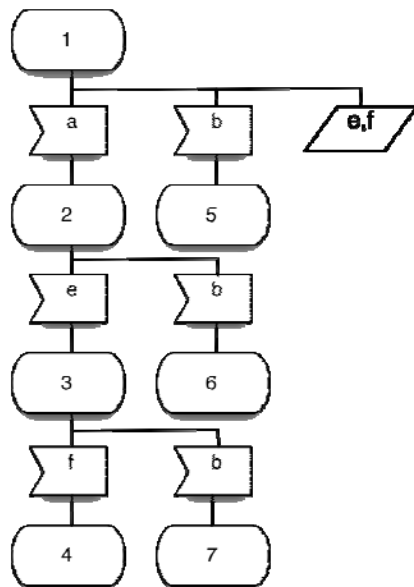
**process P**

Figure 4 Process P