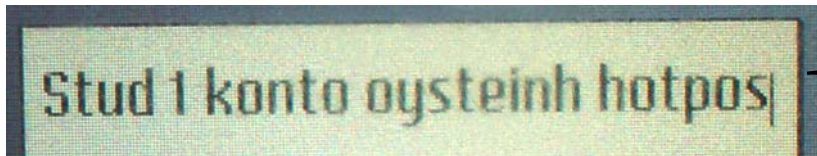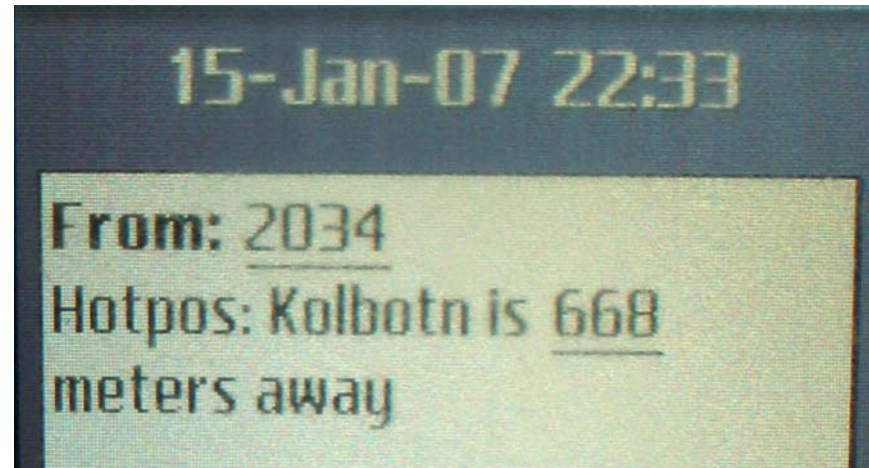# More than one service

Version 081010
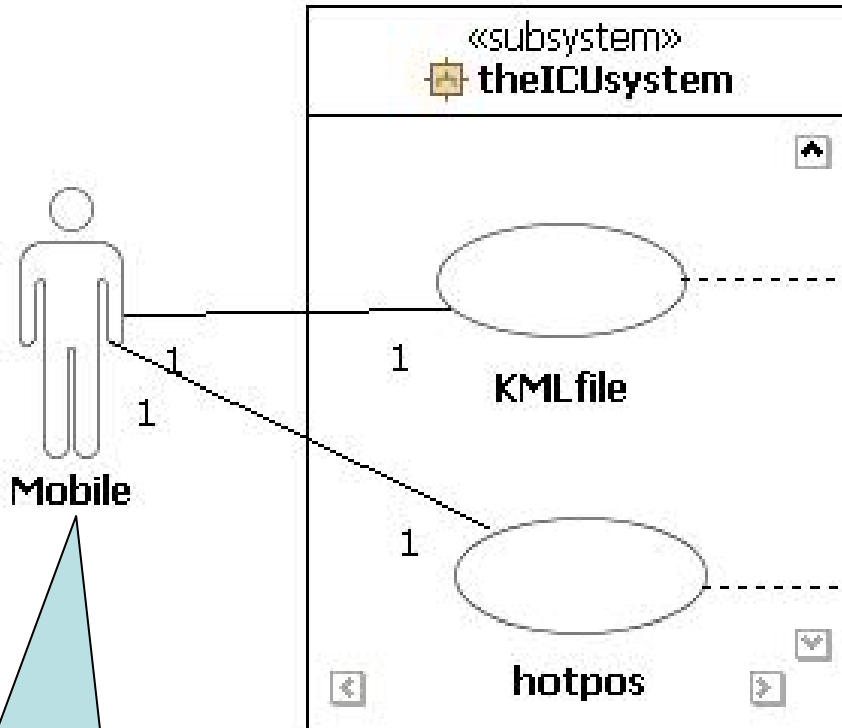
ICU 2-4

INF 5150

Hotpos: finding out where you are

Stud 1 konto oysteinh hotpos

to 2034
(Telenor!!!)

15-Jan-07 22:33

From: 2034
Hotpos: Kolbotn is 668
meters away

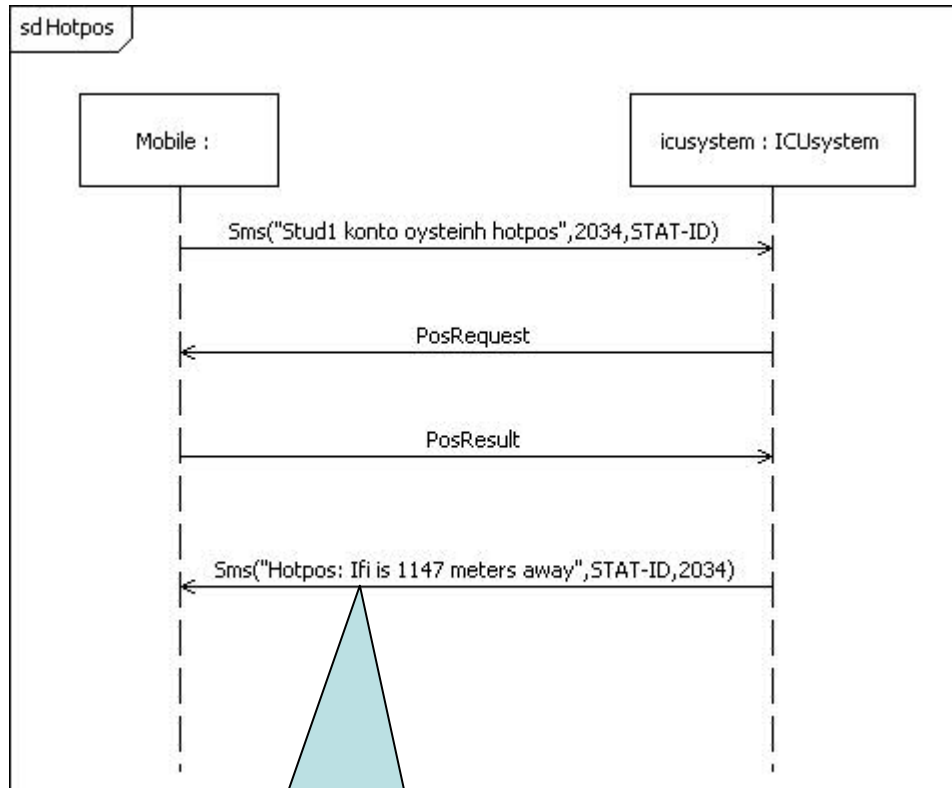# Adding a new service

«subsystem»
theICUsystem

KMLfile

Stud1 konto oysteinh KML
Write out a .kml file to be
read by GoogleEarth to
place Mobile on the map

hotpos

Stud1 konto oysteinh hotpos
Send back an SMS with info
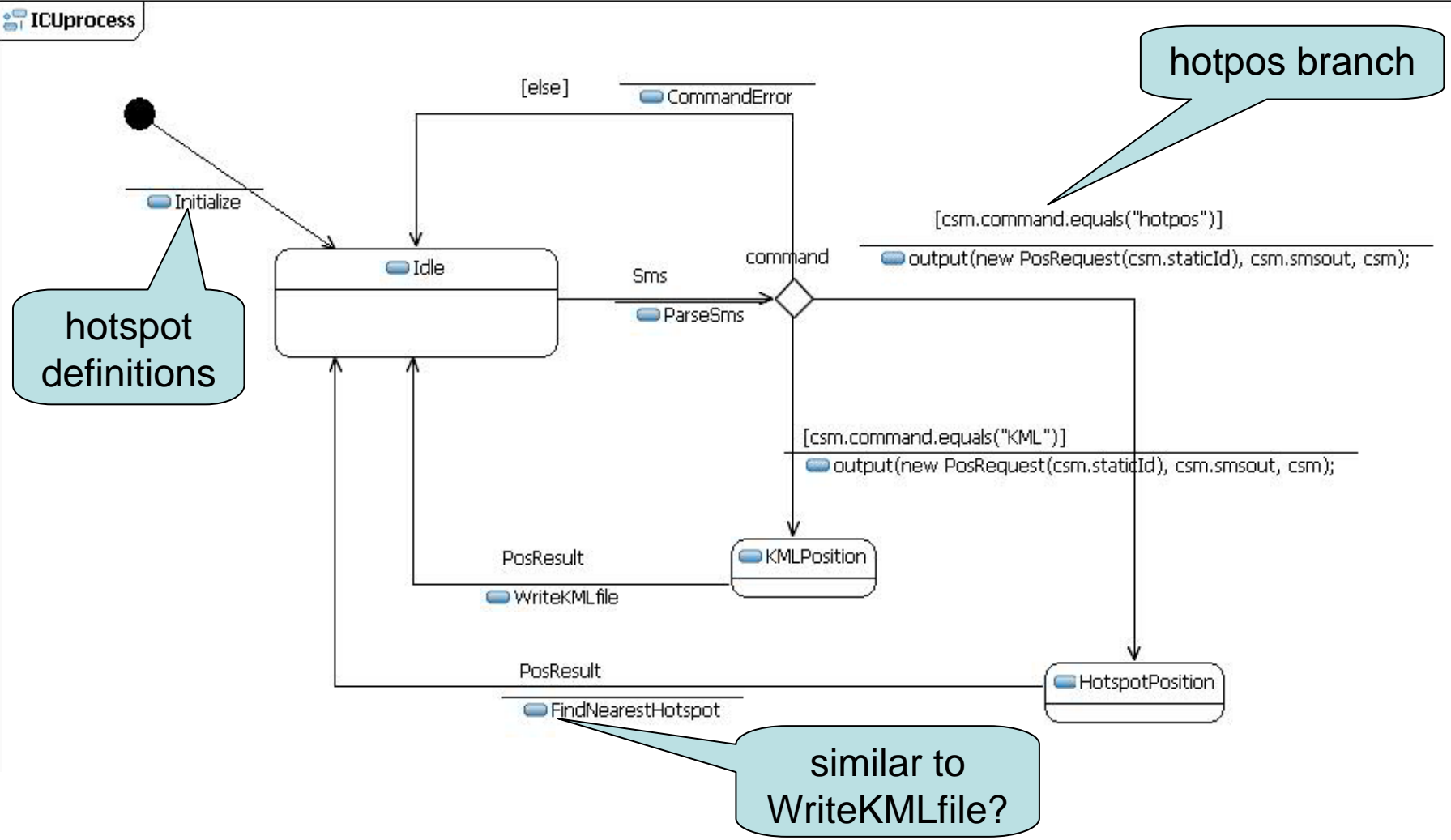on where the user is relative
to some hotspots.

Mobile

1

1

1

1

only one user at a time

INF 5150

Hotpos described by a sequence diagram

INF 5150

The modified ICUprocess

ICUprocess

hotpos branch

hotspot definitions

[csm.command.equals("hotpos")]

output(new PosRequest(csm.staticId), csm.smsout, csm);

[else]   CommandError

Initialize

Idle   command

Sms   ParseSms

[csm.command.equals("KML")]

output(new PosRequest(csm.staticId), csm.smsout, csm);

PosResult

KMLPosition

WriteKMLfile

PosResult

HotspotPosition

FindNearestHotspot

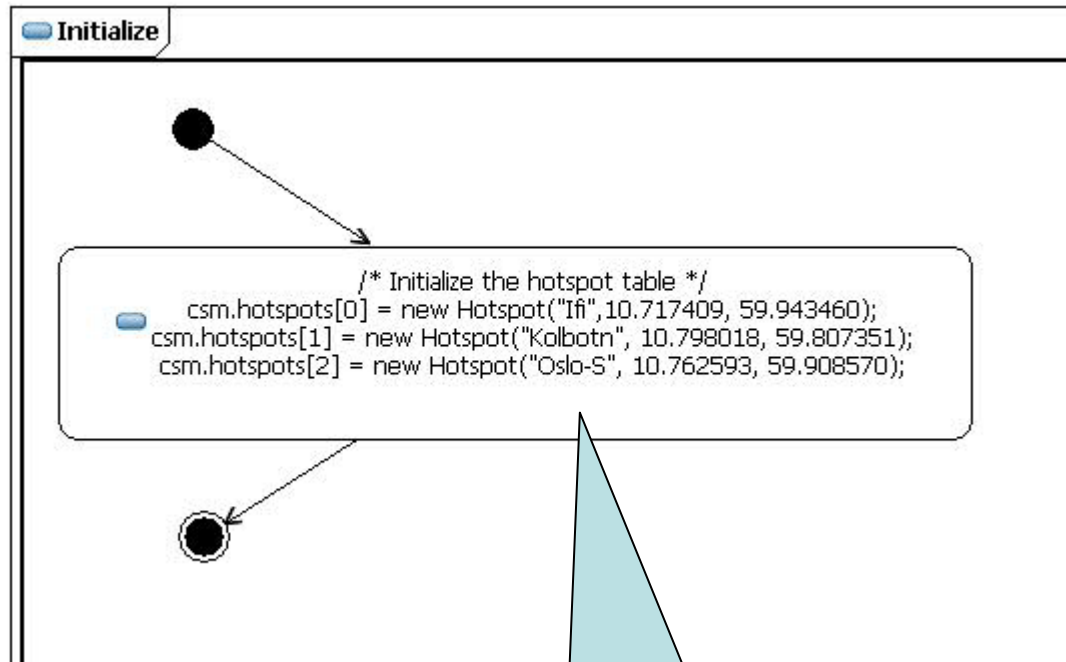similar to WriteKMLfile?

INF 5150

Buzzz 1: Why limiting to one user?

- Make up pairs with one person just beside you
- Discuss for 3 minutes why we have restricted the system to consider only one user at the time

Hardcoding the hotspots

Initialize

/* Initialize the hotspot table */
csm.hotspots[0] = new Hotspot("Ifi",10.717409, 59.943460);
csm.hotspots[1] = new Hotspot("Kolbotn", 10.798018, 59.807351);
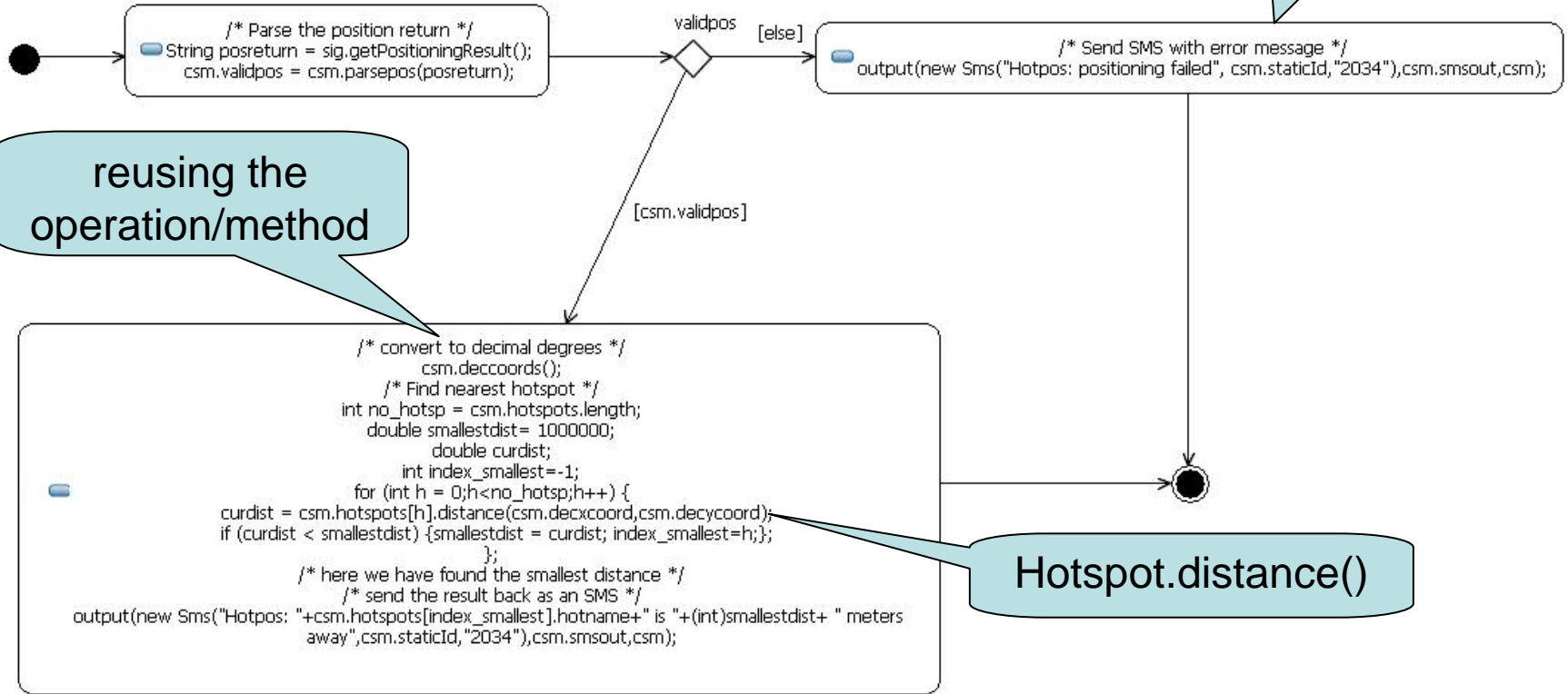csm.hotspots[2] = new Hotspot("Oslo-S", 10.762593, 59.908570);

Feel free to add your own hotspots. Remember to change the size of the array

INF5150 – Unassailable IT-systems

FindNearestHotspot

a little robustness, but it does not cover no return

FindNearestHotspot

FindNearestHotspot

/* Parse the position return */
String posreturn = sig.getPositioningResult();
csm.validpos = csm.parsepos(posreturn);

validpos

[else]

/* Send SMS with error message */
output(new Sms("Hotpos: positioning failed", csm.staticId,"2034"),csm.smsout,csm);

reusing the operation/method

[csm.validpos]

/* convert to decimal degrees */
csm.deccoords();
/* Find nearest hotspot */
int no_hotsp = csm.hotspots.length;
double smallestdist= 1000000;
double curdist;
int index_smallest=-1;
for (int h = 0;h<no_hotsp;h++) {
curdist = csm.hotspots[h].distance(csm.decxcoord,csm.decycoord);
if (curdist < smallestdist) {smallestdist = curdist; index_smallest=h;};
};
/* here we have found the smallest distance */
/* send the result back as an SMS */
output(new Sms("Hotpos: "+csm.hotspots[index_smallest].hotname+" is "+(int)smallestdist+ " meters
away",csm.staticId,"2034"),csm.smsout,csm);

Hotspot.distance()

INF 5150

Hotspot.distance()

distance

```
/* distance on the great circle of the earth between (decx,decy) and (x,y) given in decimal degrees */
double earth_radius = 6372797; /* meters */
double rad_decx = Math.toRadians(decx);
double rad_decy = Math.toRadians(decy);
double rad_x = Math.toRadians(x);
double rad_y = Math.toRadians(y);

double dist = earth_radius * Math.acos(Math.cos(rad_decy)*Math.cos(rad_y)*Math.cos(rad_decx-rad_x)
+Math.sin(rad_decy)*Math.sin(rad_y));
return dist;
```

Separation of concerns

- We want to separate different concerns of the ICU system through using separate state machines that communicate

- The architecture of the ICUSystem will evolve

- One process controls
  - the handling of SMSes
  - and the production of the KML file

- One process controls the handling of the data
  - which are still going to be hardcoded (for now)

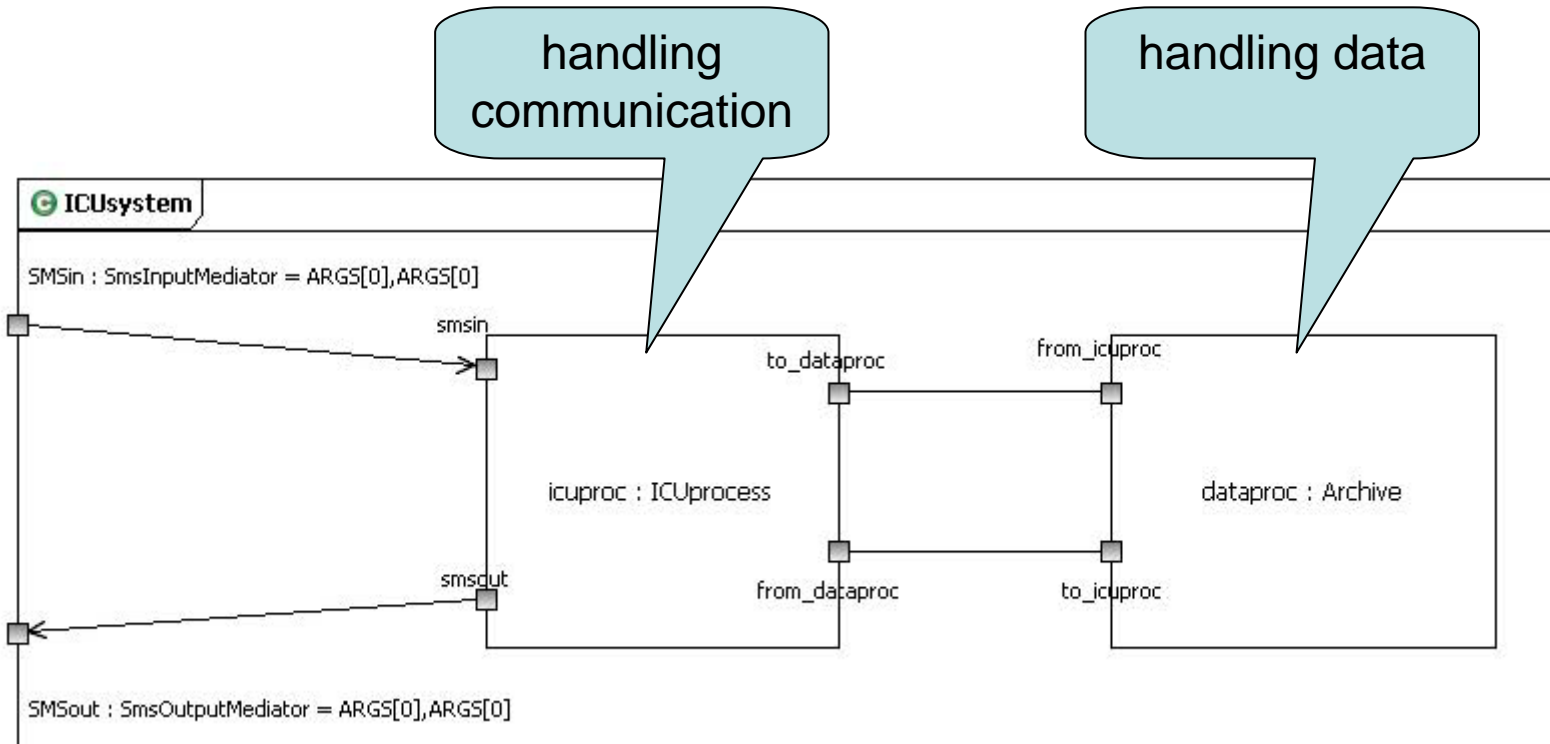- These processes communicate with signals that we define ourselves

INF 5150

Hotpos service – as seen from the context

INF 5150
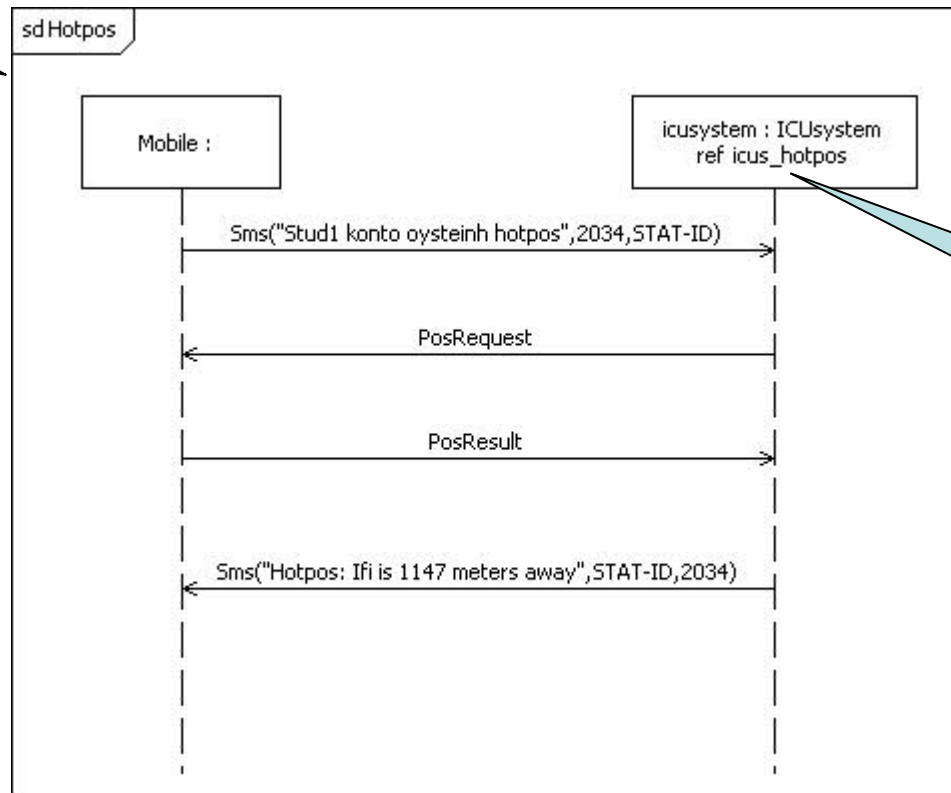
Inside the ICUsystem

INF 5150

Decomposing the ICUsystem

The behavior inside ICUsystem

gate matching the level above

handling communication

handling data

these signals are defined by us

sd icus_hotpos

icuproc : ICUprocess

dataproc : Archive

Sms("Stud1 konto oysteinh hotpos",2034,STAT-ID)

PosRequest

PosResult

GetNearestHotspot(10.73,59.95)

NearestHotspot("Ifi", 1147)

Sms("Hotpos: Ifi is 1147 meters away",STAT-ID,2034)

INF 5150
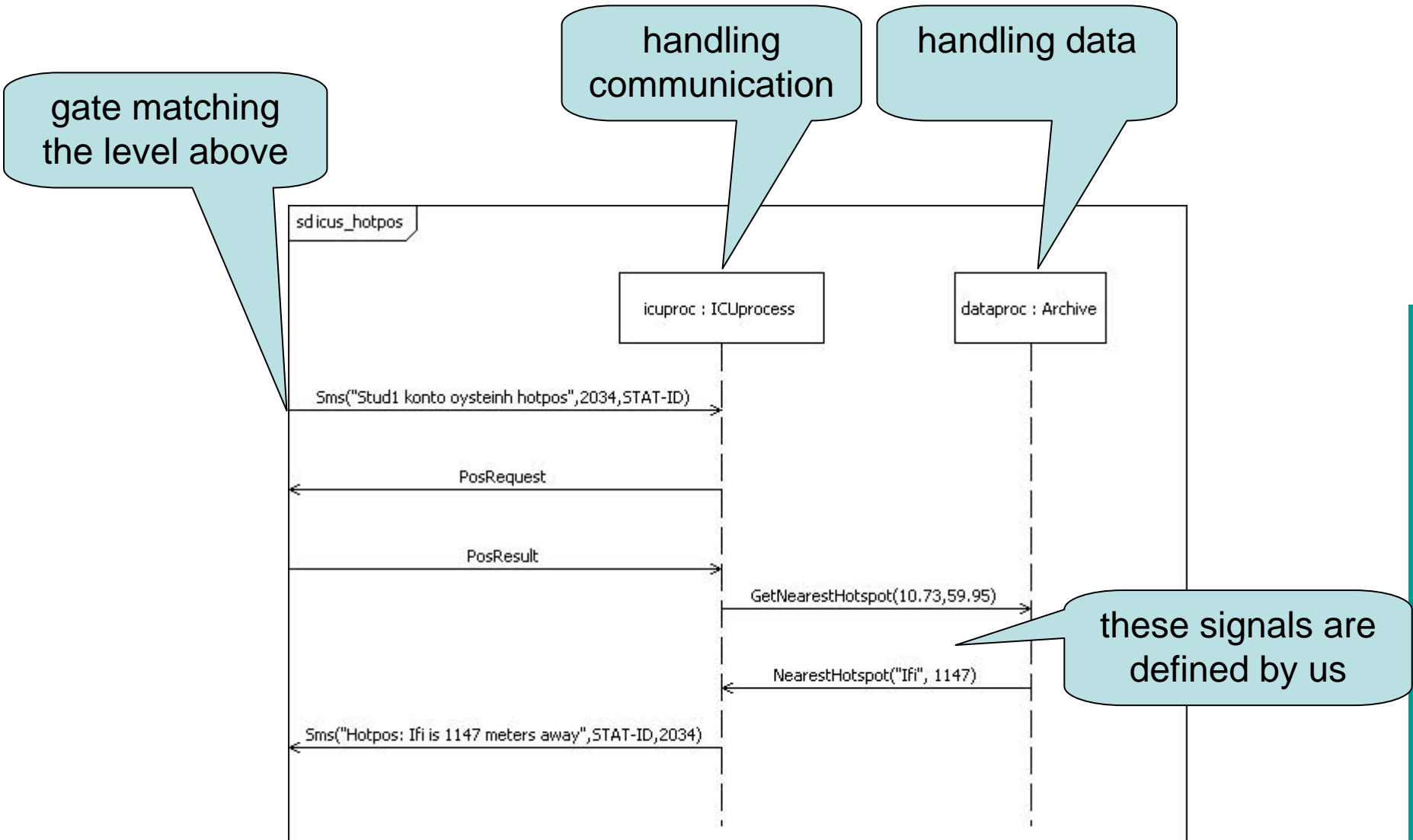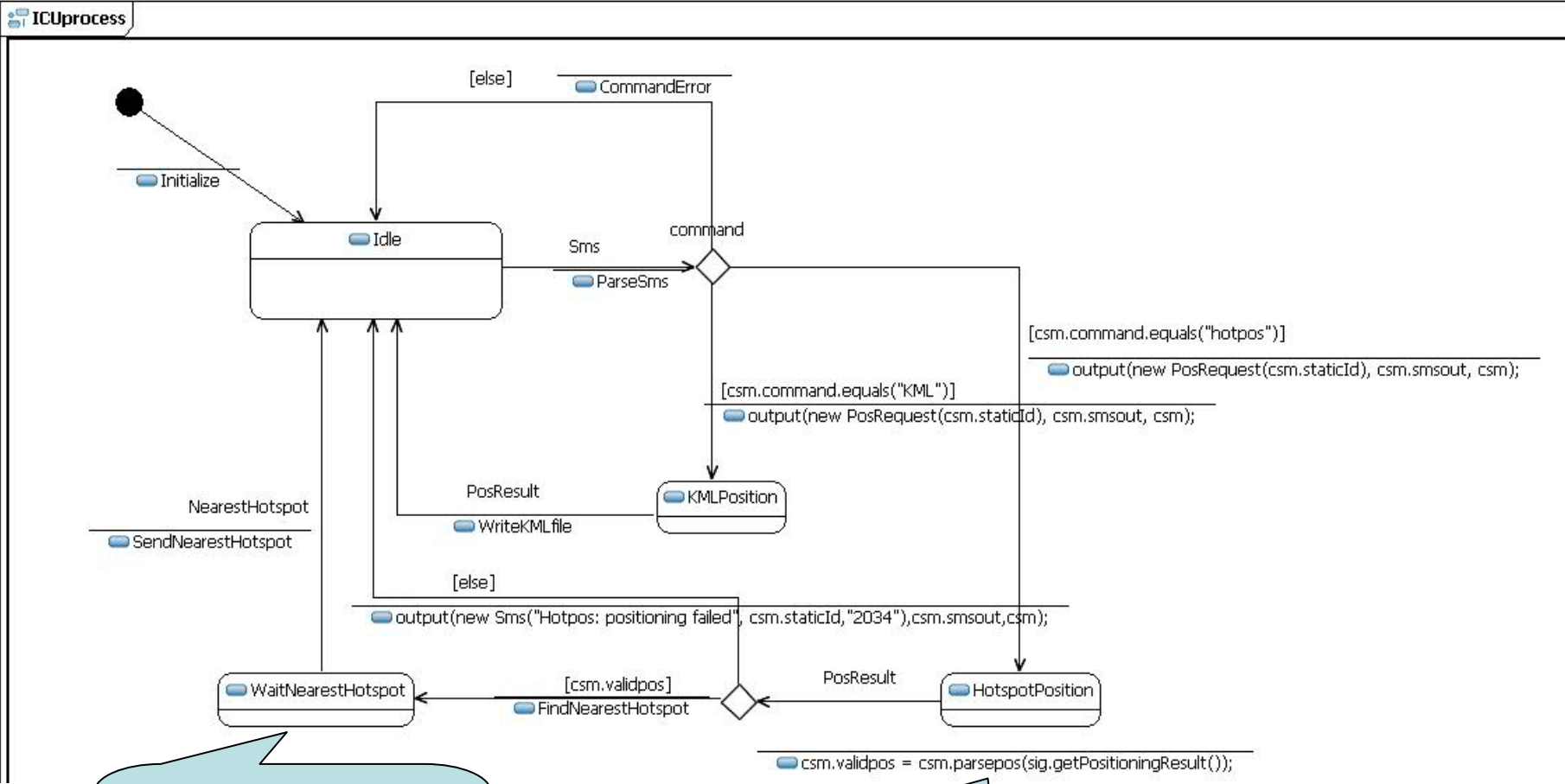
The essence of decomposition
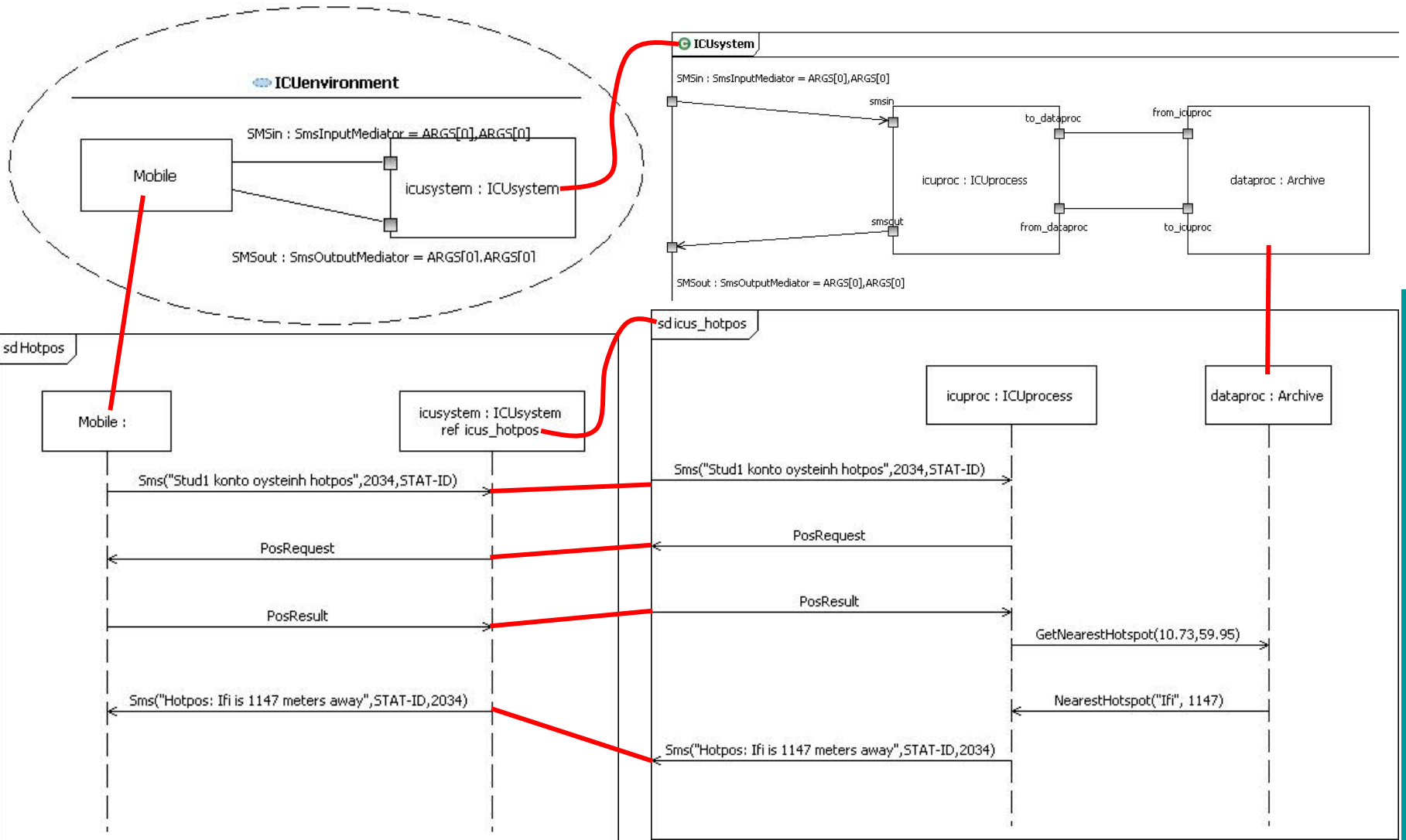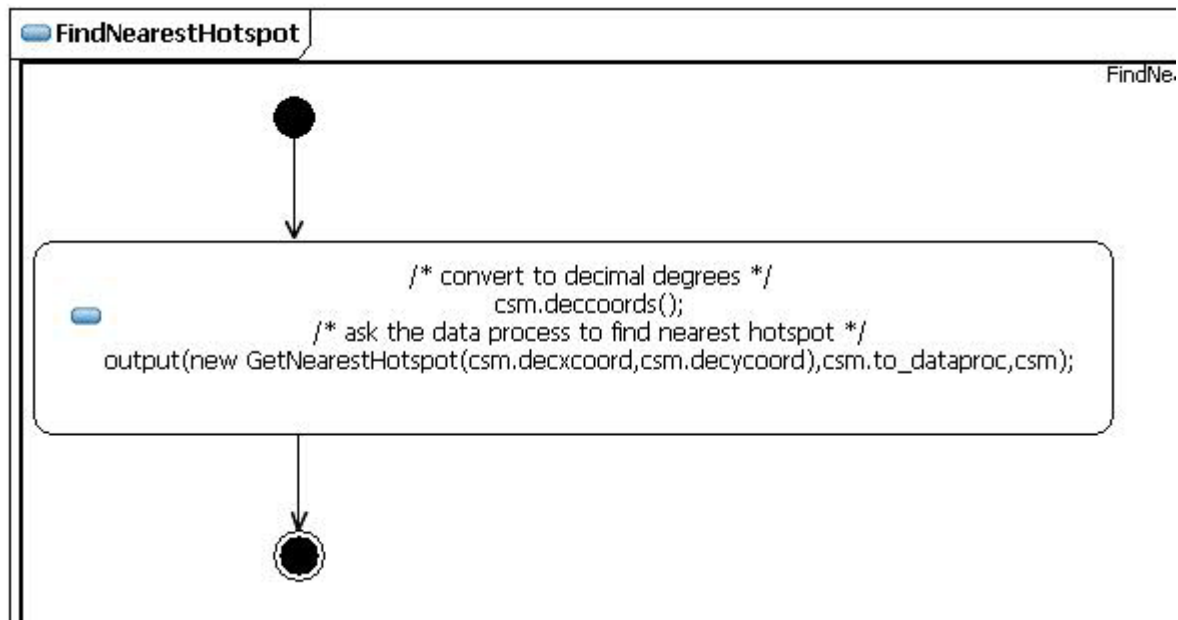
The classes and signals

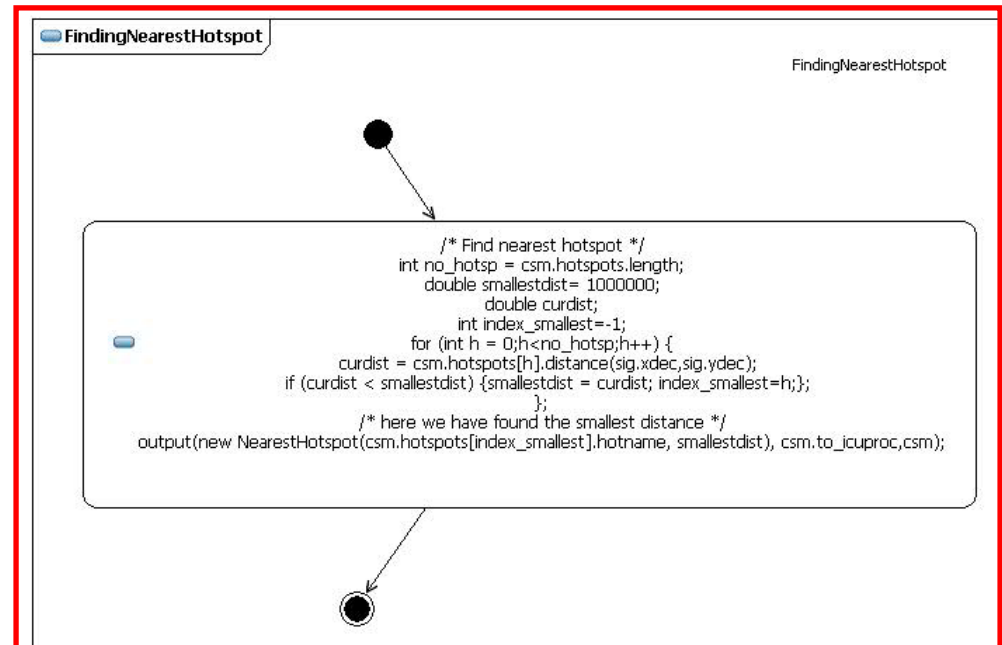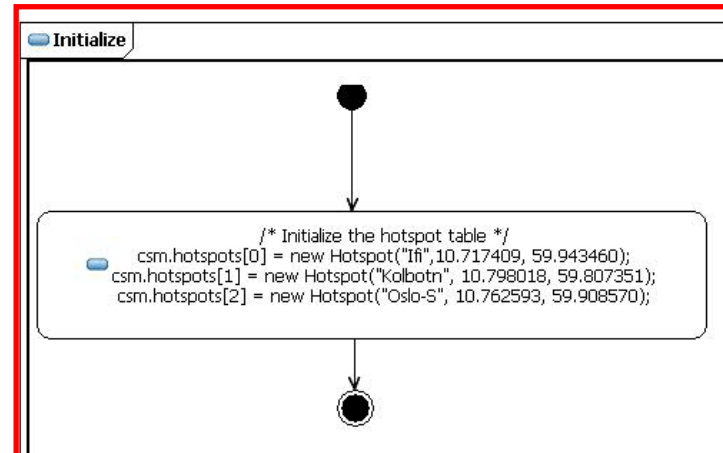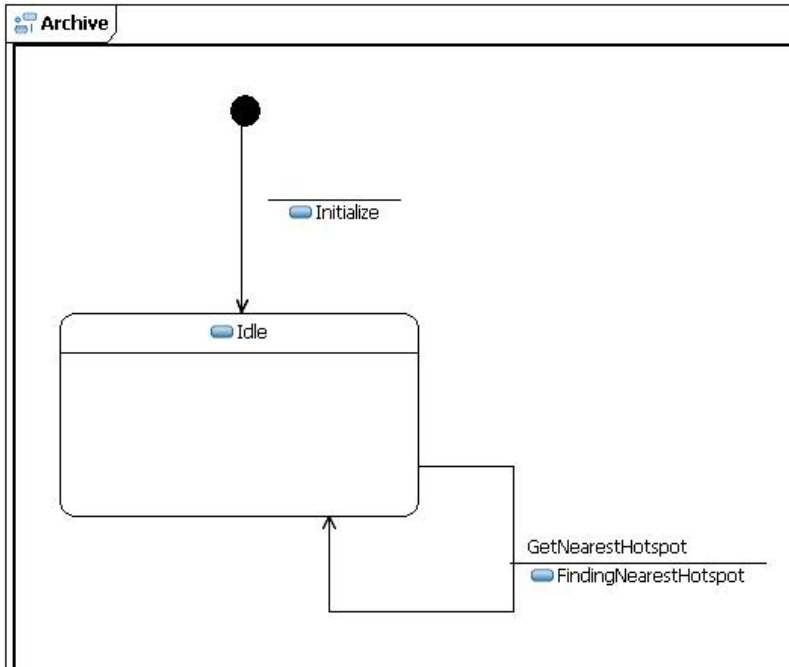INF 5150
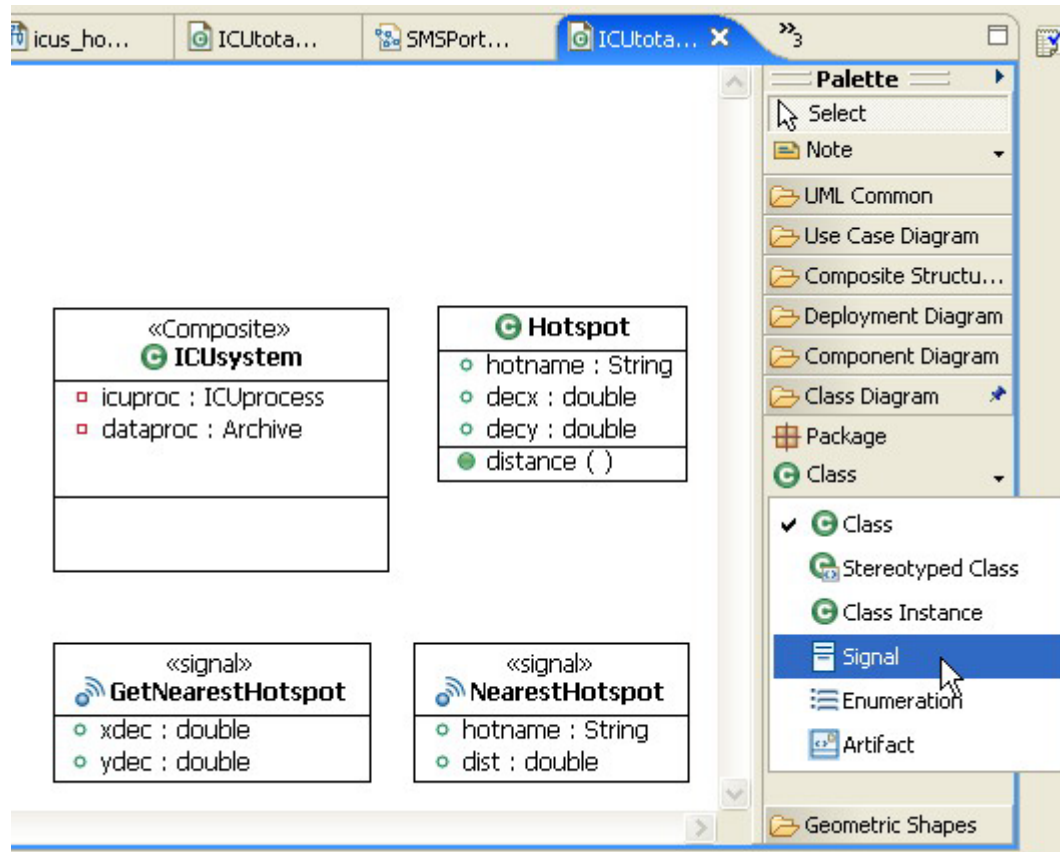
ICUprocess revisited (when intro Archive)

Here the data process does the calculations

FindNearestHotspot has been split up

INF 5150

FindNearestHotspot has become pure sending

INF 5150

Archive – the data process

Archive

Initialize

/* Initialize the hotspot table */
csm.hotspots[0] = new Hotspot("Ifi",10.717409, 59.943460);
csm.hotspots[1] = new Hotspot("Kolbotn", 10.798018, 59.807351);
csm.hotspots[2] = new Hotspot("Oslo-S", 10.762593, 59.908570);

FindingNearestHotspot

FindingNearestHotspot

/* Find nearest hotspot */
int no_hotsp = csm.hotspots.length;
double smallestdist= 1000000;
double curdist;
int index_smallest=-1;
for (int h = 0;h<no_hotsp;h++) {
curdist = csm.hotspots[h].distance(sig.xdec,sig.ydec);
if (curdist < smallestdist) {smallestdist = curdist; index_smallest=h;};
};
/* here we have found the smallest distance */
output(new NearestHotspot(csm.hotspots[index_smallest].hotname, smallestdist), csm.to_icuproc,csm);

Idle

GetNearestHotspot
FindingNearestHotspot

INF 5150

Buzzz 2: Why the Archive process?

- Pair up with another student
- Discuss 3 minutes what benefits there are with introducing the Archive process

INF 5150

Why the separate data process?

- Isolate the work on the (semi-)persistent data
  - we shall later show how the handling of data can change without changing its interfaces

- Provide a simple critical region
  - this will be clearer later when we interface to a database system that works concurrently with our system

- The Archive process and the ICUprocess can be designed by different persons

How to make the protocol with the Archive?

- Signals close to the application
  - this is what we have chosen
  - we want to branch on signals rather than on data
- Signals close to data
  - such as e.g. SQL
  - most important information will be in the parameters and branching will be on decision-nodes
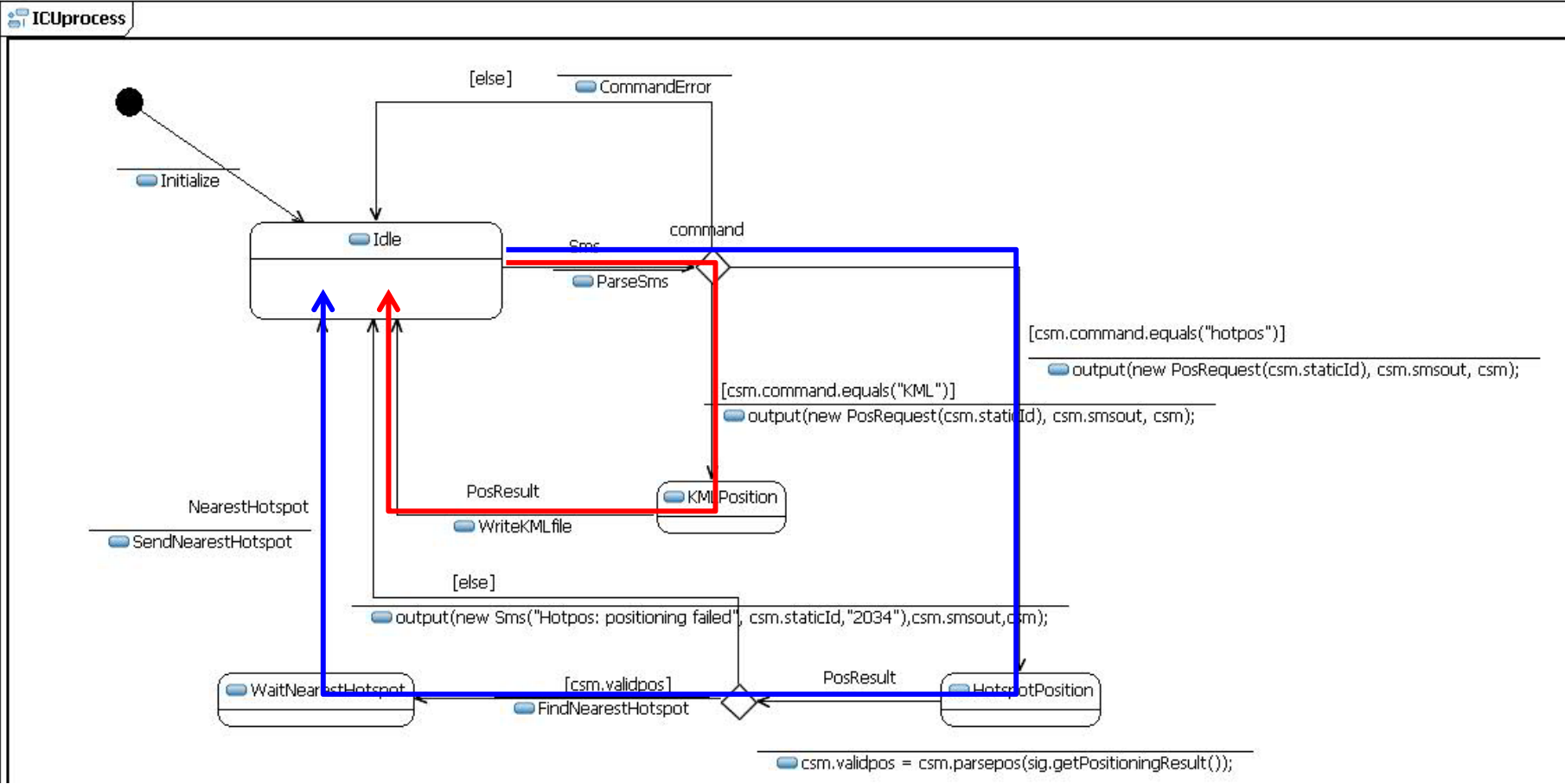- Do not worry about many signal types!
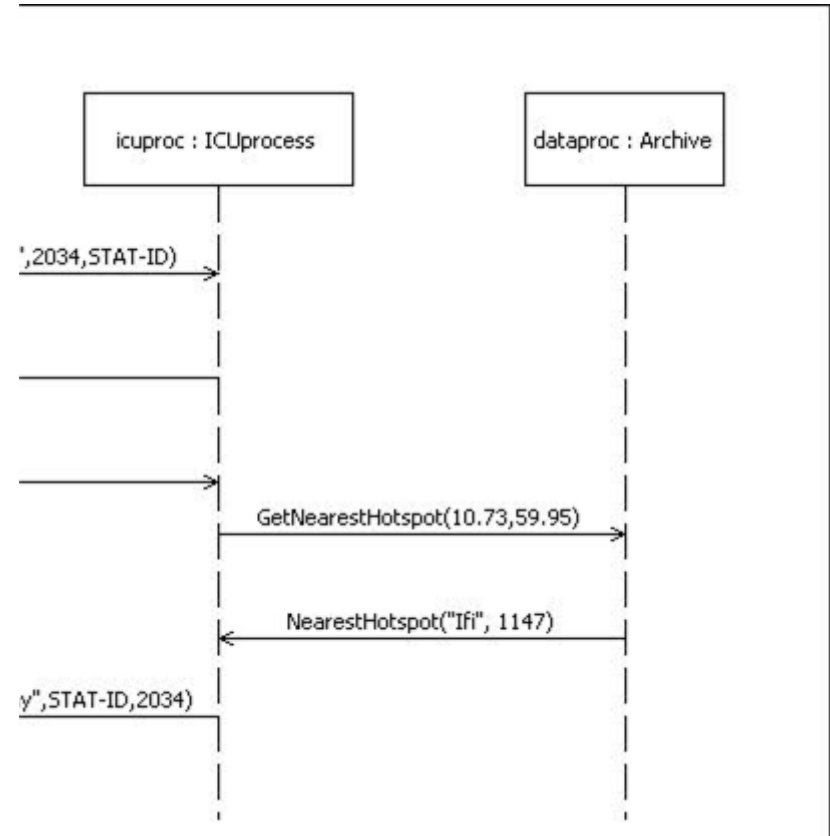
Services as Submachine States

... but we have only one sequential process

INF 5150

ICUprocess serving 2 services

Separation of Concerns

- Isolate reusable functions
  - through operation/method: *parsepos* and *deccoords*

- Separate independent concurrent tasks
  - through parts in composite structures: *icuproc* and *dataproc*

- Separate different alternating services
  - through submachinestates of internal state machines
  - *KML* and *Hotpos*
  - We have introduced the following invariant:
    - One user (defined by one mobile telephone) can only be involved in one (top level) service at one instant

ICUprocess with 2 submachine states

These state machines are not concurrent!

Each submachine state refers to a state machine

INF 5150

Submachine states

- Submachine states are states
- Submachine states have a state machine definition
  - but at the level of the submachine state, they are perceived only as states
- Submachine states are compiled into JavaFrame composite states
  - which must not be confused with composite structures!!!
  - UML also has something called "composite states" but they are not as powerful as submachine states. The JavaFrame compiler does not recognize UML composite states.
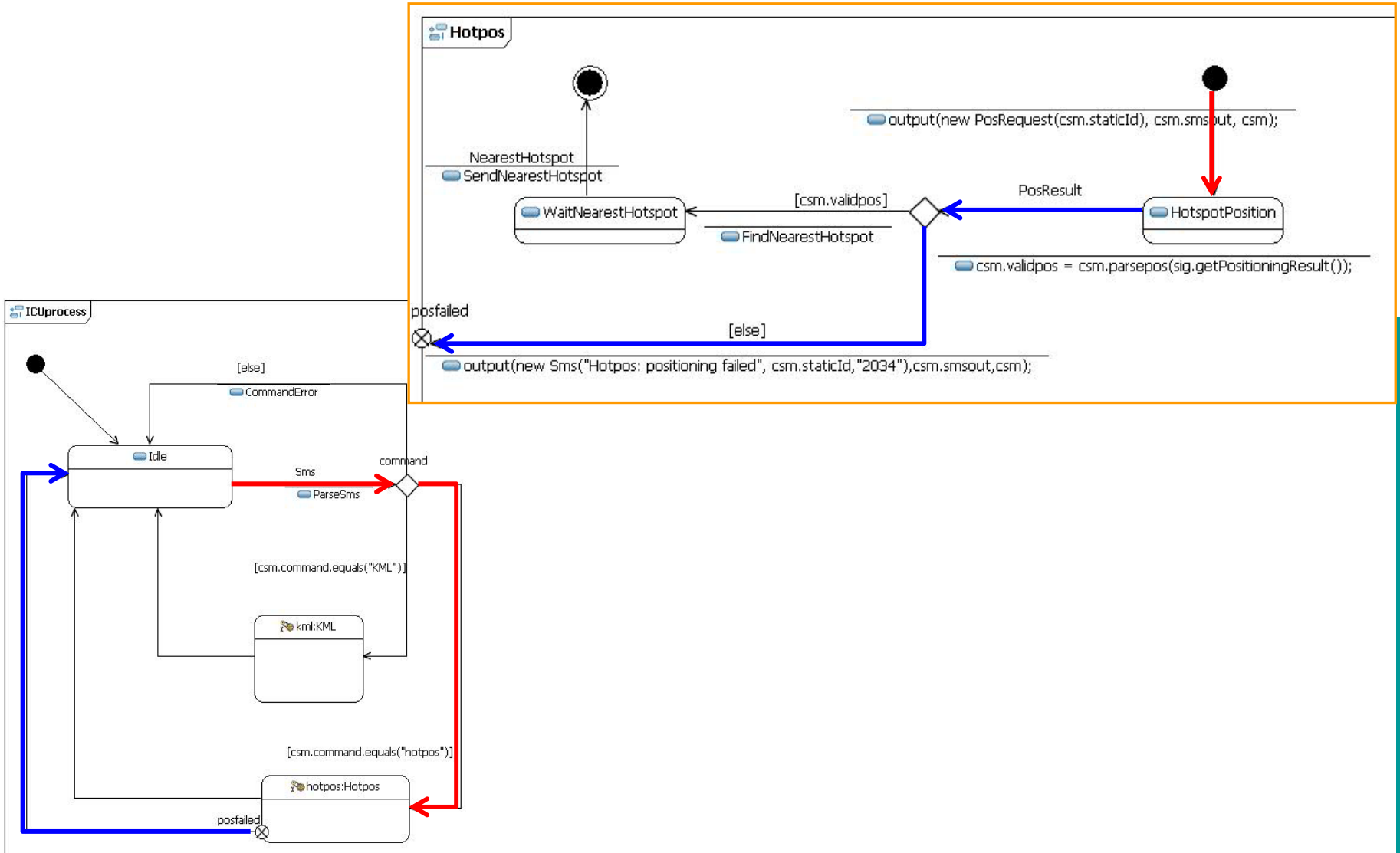
INF 5150

KML process inside ICUprocess

INF 5150

INF5150 – Unassailable IT-systems
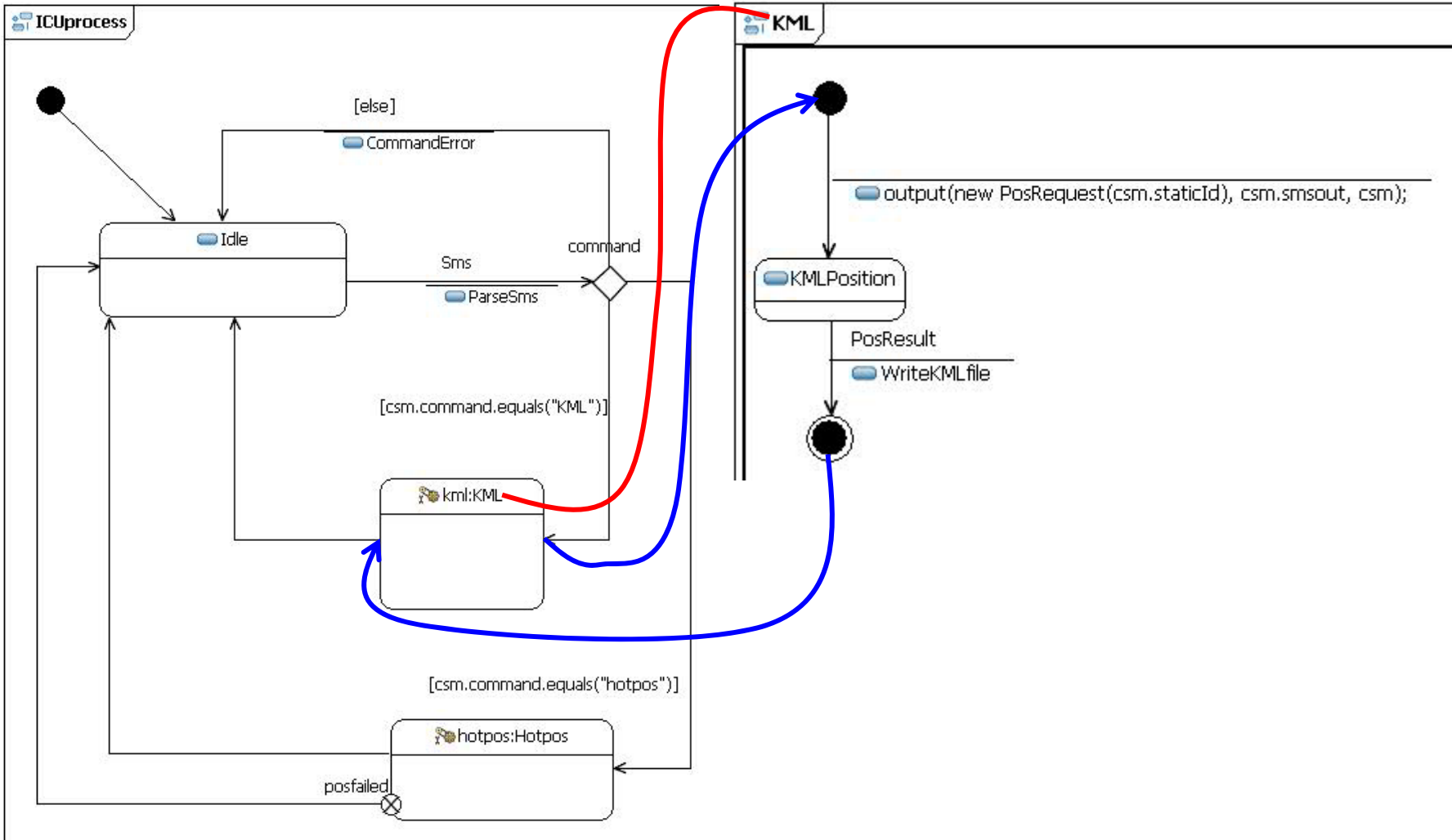
Hotpos process inside ICUprocess

INF 5150

exit point
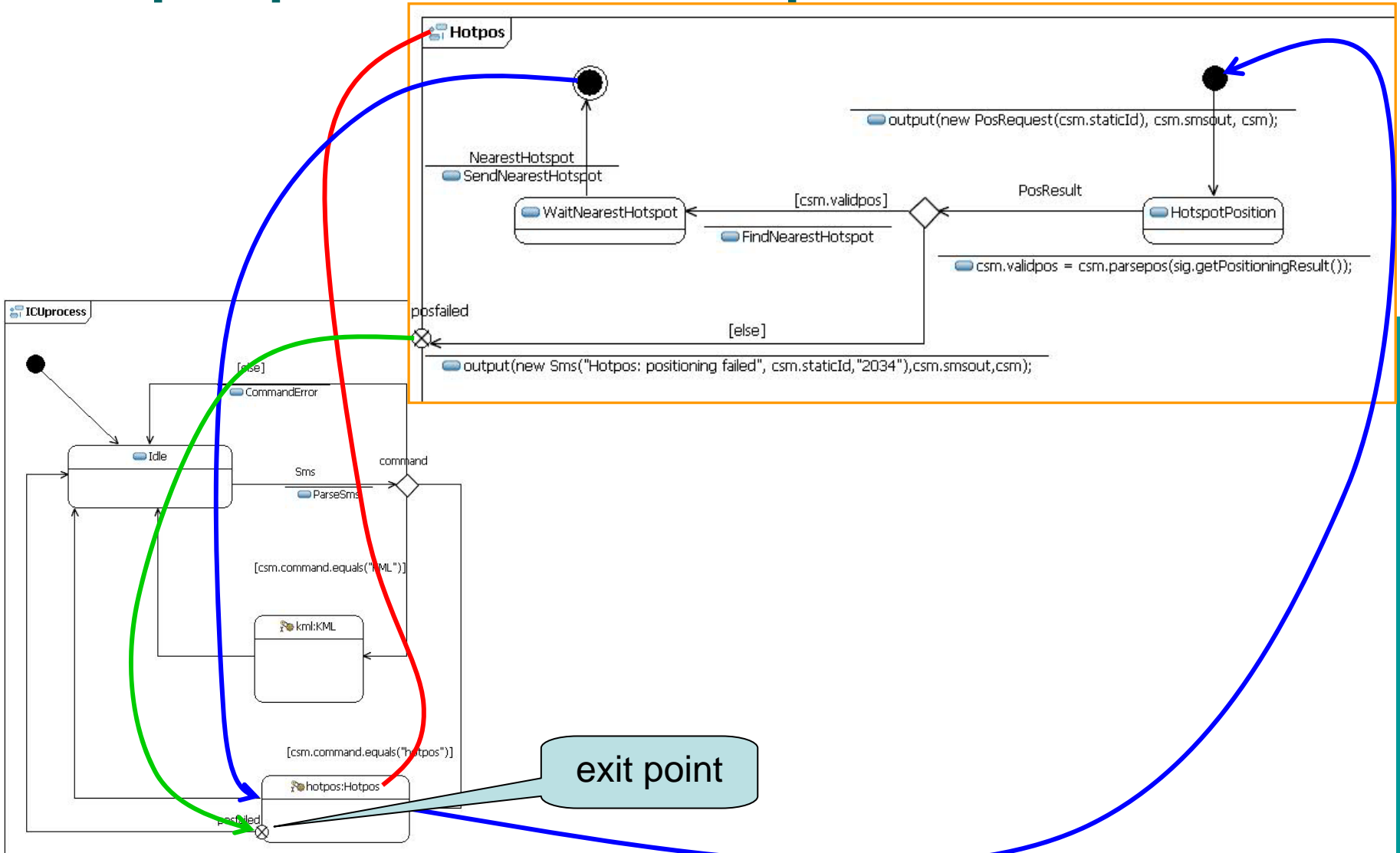
Two assembled transitions

Execution as seen from JFTrace

Stack of states

2 processes

Filtered Trace from /127.0.0.1:54321 at 2007-02-18 14:19:21.497

Table View

Time	State Machine	Current State	Input	Transition Behaviour	Next State
0	New ICUsystem_ICUprocess@3f3aac99				
0	New ICUsystem_Archive@3226ac99				
1803	ICUsystem_ICUprocess@3f3aac99	null	StartMessage@3e27ec99		Idle
1803	ICUsystem_Archive@3226ac99	null	StartMessage@325bac99		Idle
45065	ICUsystem_ICUprocess@3f3aac99	Idle	Sms@55062c99 (Stud1 konto oysteinh hotpos,2034,A-HAUGEN)	Output PosRequest@6c19ec99	HotspotPosition^hotpos
47508	ICUsystem_ICUprocess@3f3aac99	HotspotPosition^hotpos	PosResult@57836c99	Output GetNearestHotspot@11c06c99 (10.7441666666667, 59.93138888888889)	WaitNearestHotspot^hotpos
47759	ICUsystem_Archive@3226ac99	Idle	GetNearestHotspot@11c06c99 (10.7441666666667, 59.93138888888889)	Output NearestHotspot@2b6eac99 (Ifi, 2006.3401083482877)	Idle
47809	ICUsystem_ICUprocess@3f3aac99	WaitNearestHotspot^hotpos	NearestHotspot@2b6eac99 (Ifi, 2006.3401083482877)	Output Sms@2e5a6c99 (Hotpos: Ifi is 2006 meters away,A-HAUGEN,2034)	Idle

INF 5150

Write down the names of these elements
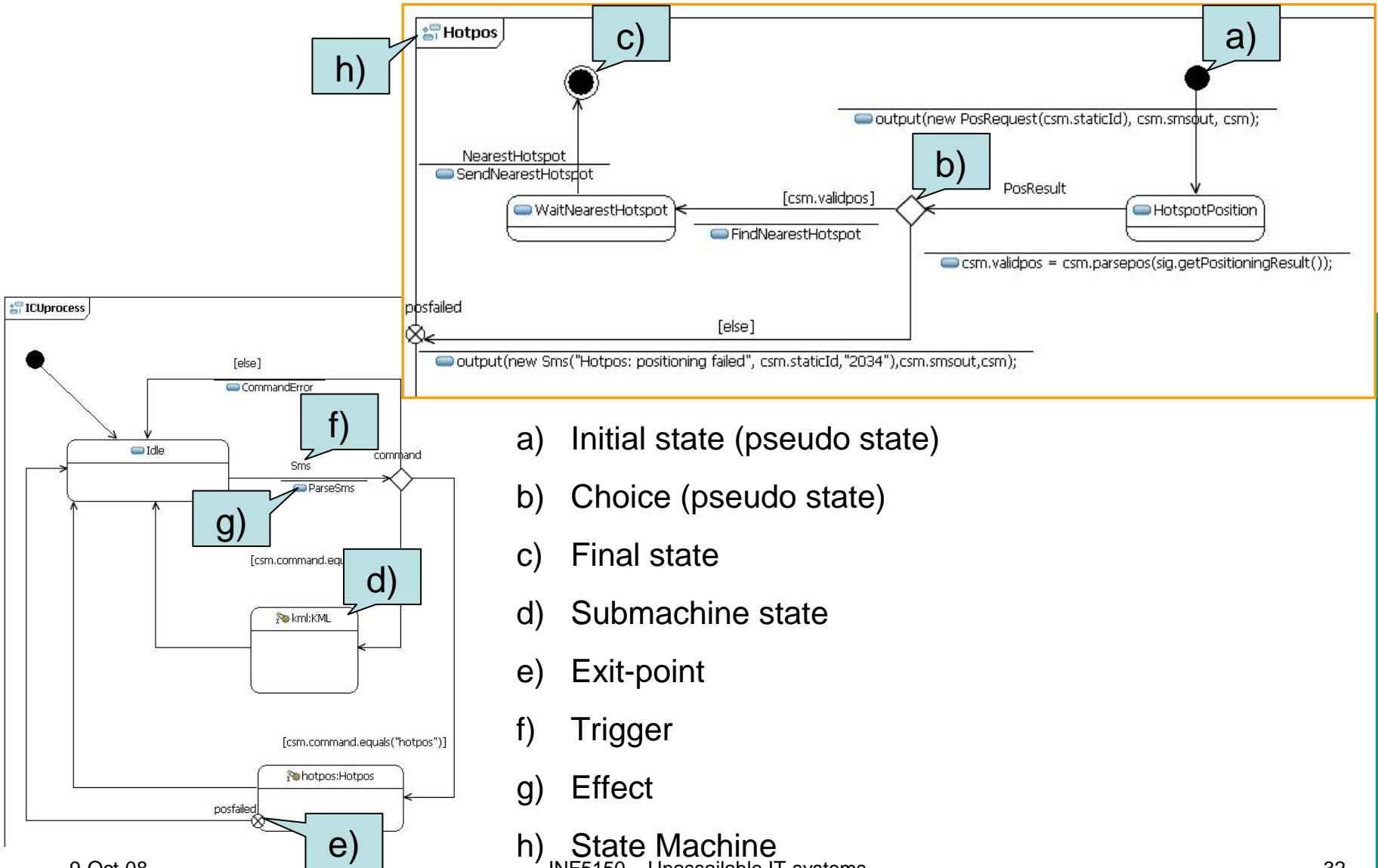
a) Initial state (pseudo state)

b) Choice (pseudo state)

c) Final state

d) Submachine state

e) Exit-point

f) Trigger

g) Effect

h) State Machine

INF 5150