<http://cruise.eecs.uottawa.ca/umpleonline/umple.php?model=131114274439>

<http://cruise.eecs.uottawa.ca/umpleonline/umple.php?model=131114468978>

class Decision

{

status {

UnderConstruction {

propose -> OnAgenda;

drop -> Dropped;

}

OnAgenda {

drop -> Dropped;

reject -> Rejected;

elevate -> Approved;

return -> UnderReconsideration;

split -> Split;

merge -> Merged;

}

Approved {

approvedByTop -> Accepted;

putOnAgenda -> OnAgenda;

}

UnderReconsideration{

putOnAgenda -> OnAgenda;

drop -> Dropped;

}

}

}

class AlarmClock

{

state {

Off{

entry / {

System.out.println("Alarm off!");

pause(1000);

power();

}

power -> On;}

On{

entry / {System.out.println("Alarm on!");}

TimeNotSet{

SetTimeHour{

entry / {

System.out.println("Enter settimehour");

pause(1500);

enter();}

enter -> SetTimeMinute;

}

SetTimeMinute{

entry / {

System.out.println("Enter settimemin");

pause(1500);

enter();}

enter -> SetTimeSecond;

}

SetTimeSecond{

entry / {

System.out.println("Enter settimesec");

pause(1500);

enter();

}

enter -> Running;

}

}

Running{

entry / {

System.out.println("Clock running");

}

setAlarm -> SetAlarm;

setTime -> TimeNotSet;

alarmTime -> AlarmRunning;

power -> Off;

}

SetAlarm{

entry / {

System.out.println("Setting Alarm");

pause(1500);

enter();}

SetHour{

entry / {

System.out.println("Setting Alarm Hour");

pause(1500);

enter();

}

enter -> SetMinute;

}

SetMinute{

entry / {

System.out.println("Setting alarm min");

pause(1500);

enter();}

enter -> Running;}

}

AlarmRunning{

entry / {

System.out.println("Alarm running");

pause(1500);

}

AlarmActive{

entry / {

System.out.println("Alarm active");

}

hitSnooze -> Snooze;

afterTimeLimit -> Running;}

Snooze{

entry / {

System.out.println("snooze");

pause(1500);

System.out.println("Quit snooze");

afterTimeLimit();}

afterTimeLimit -> AlarmActive;}

}

}

}

void pause(long ms) {

try {

Thread.currentThread().sleep(ms);//sleep for 1000 ms

}

catch(InterruptedException ie){

}

}

public static void main(String[] args){

System.out.println("Start simulation");

AlarmClock a = new AlarmClock();

pause(1500);

setAlarm();

pause(1500);

setTime();

pause(1500);

alarmTime();

pause(1500);

hitSnooze();

afterTimeLimit();

power();

System.out.println("Stop simulation");

}

}

Start simulation

Alarm off!

Alarm on!

Enter settimehour

Enter settimemin

Enter settimesec

Clock running

Setting Alarm

Setting Alarm Hour

Setting alarm min

Clock running

Enter settimehour

Enter settimemin

Enter settimesec

Clock running

Alarm running

Alarm active

snooze

Quit snooze

Alarm active

Clock running

Alarm off!

Alarm on!

Enter settimehour

Enter settimemin

Enter settimesec

Clock running

Stop simulation