Høgskolen i Østfold

EXAMINATION

Course code:	Course:
ITI43515	Modelling Cyber-Physical Systems
Date:	**Duration:**
11 May 2017	4 hours
Permitted sources:	**Lecturer:**
All written aiding tools allowed	Professor Øystein Haugen

The examination:
The examination papers consist of 9 pages inclusive this page. Please check that the examination papers are complete before you start answering the questions.

Pages 1-4 describes the exam.
Pages 5-9 are Annexes giving relevant details from the course.

The exam is only in English, but the candidate may answer in Norwegian if he or she so pleases.

Date of announcement of the examination results: 7 June 2017

The examination results are available on the Studentweb no later than two workdays after the announcement of the examination results www.hiof.no/studentweb

Exam Modeling Cyber-Physical Systems Spring 2017

Context

The context is The Room – our experimental context consisting of a controlling computer equipped with a Tellstick Duo which communicates wirelessly on 433.92 MHz to a set of sensors and actuators.

Our starting point is The Room X2D. Version X2D has one thermometer and one on/off switch to turn a heater on or off. Version X2D tries to minimize repeated switching, and covers all possible signals in every state.

Version X2DX enhances X2D in the following way:

1. There are 2 thermometers – one by the window and one on the opposite wall
2. There is a magnetic sensor that sends a signal "opened" when the two magnetic pieces are moved apart, and "closed" when the two magnetic pieces are moved tightly together. The magnetic sensor is attached to the window to monitor when it opens and when it is shut. (We assume that there exists a driver in the PSM for the magnetic sensor such that signals "opened" and "closed" can be sent to PIM.)

We have designed The Room X2DX under the following assumptions:

- The room will get warmer when the heater is on, and eventually go above comfort temperature
- The room will get colder when the heater is off, and eventually go below comfort temperature

The assumptions are not always realistic, but for our purpose, they make the temperature control a little easier.

Exercise 1 Model modifications (75%)

a. Composite structure (UML) or configuration (ThingML)

Modify the composite structure / configuration of *X2D* depicted by Figure 1 in UML, into the configuration in ThingML for *X2DX* where the magnetic sensor should be included in the PSM and connected up to the PIM. You also need a mock interface to simulate the window movements.

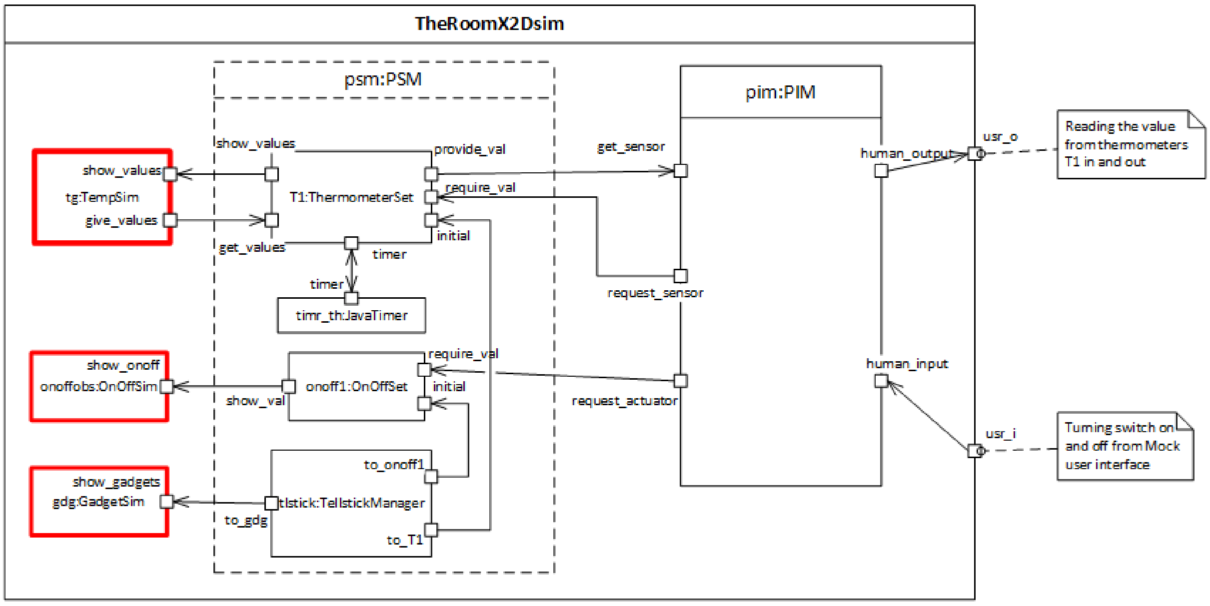

Figure 1 The Room X2D composite structure

b. The Room X2DX behavior specification

This task will ask you to create a sequence diagram for the X2DX system.
The added functionality of The Room X2DX relates to opening and closing of the window. The issue is that when the window is open, the thermometer by the window will quickly show significantly lower temperatures than the one on the opposite wall. This should not immediately cancel out the cooling effects intended by the user when opening the window. We need to make sure that opening the window does not immediately trigger switching the heater on.

Therefore, we decide that when the window is open, the overall temperature of the room should be calculated differently from when the window is closed. The window thermometer will have a much smaller impact on the overall temperature when the window is open.

When the window is closed the overall temperature is calculated by 40% window thermometer and 60% wall thermometer, while when the window is open, the window thermometer is 10% and the wall thermometer 90%.

Make a sequence diagram of the whole simulation system where the following happens. We give here only the external stimuli. You will have to supplement by what the system itself does. To save horizontal space, let the mock interfaces be represented by gates, and disregard the timer connected to the thermometer set such that the temperature is immediately forwarded to the PIM.

0. Assume that the window is closed at the start of the scenario, that comfort temperature is set to 20C with range -1,+1 degrees, and the thermostat is running.
1. Send 19C from the window thermometer
2. Send 23.5 C from the wall thermometer
3. Send 24C from the wall thermometer
4. Open the window
5. Send 15C from the window thermometer
6. Send 22C from the wall thermometer
7. Send 9C from the window thermometer

8. Send 20C from the wall thermometer
9. Close the window

c. The Room X2DX PIM behavior definition

Here we shall **modify the PIM state machine** of X2D to the one for X2DX in ThingML such that the PIM state machine is consistent with the interaction described in the task b above.

You only need to modify the Running state. There are of course changes to the building part as well, but we ignore those in this exam.

d. Guarding against low temperatures close to the window

Furthermore, the window thermometer should guard against excessively low temperatures close by the window. We define that below +4C on the window thermometer alone is excessively low. When such a temperature occurs near the window, a message should be sent to the human user/owner of the Room.

Explain in natural language how you would add the **guard against low temperatures** near the window. Be very explicit about what you would add or modify and exactly where you would do the modifications in your system. Feel free also to use graphics (UML) or ThingML code to explain this precisely.

Exercise 2 Risk analysis of The Room X2DX (25%)

We performed a risk analysis of The Room X3C on behalf of the party room owner/user. How would the **risk analysis differ wrt. The Room X2DX**?

Consider only the following points:

- The assets
- Malicious threats
- Vulnerabilities
- Treatments

----------------------------- End of Exam ---

Annex 1 The Room X2D configuration for simulation in ThingML

```
import "psm_sim.thingml"
import "pim.thingml"
import "io.thingml"
import "javatimer.thingml"

configuration CPS {
	instance tlstick:TellstickManager
	instance T1:ThermometerSet
	instance onoff1:OnOffSet
	instance pim:PIM
	instance myself:Human
	instance timer : TimerJava

	// SIMULATION
	instance tg:TempSim
	instance onoffobs:OnOffSim
	instance gdg:GadgetSim

	// PSM
	connector tlstick.to_T1 => T1.initial
	connector tlstick.to_gdg => gdg.show_gadgets
	connector tlstick.to_onoff1 => onoff1.initial

	connector T1.provide_val => pim.get_sensor
	connector T1.timer => timer.timer
	connector T1.show_values => tg.show_values

	connector onoff1.show_val => onoffobs.show_onoff

	// HMI
	connector myself.send_cmd => pim.human_input

	// PIM outwards
	connector pim.request_sensor => T1.require_val
	connector pim.request_actuator => onoff1.require_val
	connector pim.human_output => myself.get_values

	// SIMULATION
	connector tg.give_values => T1.get_values
}
```

Annex 2 Running state in PIM state machine of X2D

```
composite state Running init Thermostat keeps history {

        composite state Thermostat init TemprDecide {
// notice we do not keep history, we do not know whether to turn on or off
            state TemprDecide {
                transition -> TemprDecrease
                event temp2:get_sensor?temperature
                guard temp2.t>=tmrature-1
// We choose switch OFF as much as we can here
                action do
                    request_actuator!SwitchOff(switch_id)
                end

                transition -> TemprIncrease
                event temp2:get_sensor?temperature
                guard temp2.t<tmrature-1
                action do
                    request_actuator!SwitchOn(switch_id)
                end
            }

            state TemprIncrease{
// Invariant: Switch is ON and temperature should increase

                transition -> TemprIncrease
                event temp:get_sensor?temperature
                guard temp.t<=tmrature+1
// it should keep increasing until well above the desired temperature
                action do
                    // Nothing
                end

                transition -> TemprDecrease
                event temp2:get_sensor?temperature
                guard temp2.t>tmrature+1
                action do
                    request_actuator!SwitchOff(switch_id)
                end
            }

            state TemprDecrease{
// Invariant: Switch is OFF and temperature should decrease

                transition-> TemprDecrease
                event temp:get_sensor?temperature
                guard temp.t>=tmrature-1 // it should keep
decreasing until well below the desired temperature
                action do
                    // Nothing
                end

                transition -> TemprIncrease
                event temp2:get_sensor?temperature
                guard temp2.t<tmrature-1
                action do
                    request_actuator!SwitchOn(switch_id)
                end
            }
```

```
        transition -> On
        event swon:human_input?SwitchOn
        action do
            request_actuator!SwitchOn(swon.did)
        end
        transition -> Off
        event swoff:human_input?SwitchOff
        action do
            request_actuator!SwitchOff(swoff.did)
        end
        transition -> Thermostat
        event set_temp:human_input?set_temperature
        action do
            tmrature = set_temp.t
        end
} // end Thermostat

state On {
        transition -> Off
        event swoff:human_input?SwitchOff
        action do
            request_actuator!SwitchOff(swoff.did)
        end
        transition -> On
        event swon:human_input?SwitchOn
        action do
            request_actuator!SwitchOn(swon.did)
        end
        transition -> Thermostat
        event set_temp:human_input?set_temperature
        action do
            tmrature = set_temp.t
        end
}
state Off {
        transition -> Off
        event swoff:human_input?SwitchOff
        action do
            request_actuator!SwitchOff(swoff.did)
        end
        transition -> On
        event swon:human_input?SwitchOn
        action do
            request_actuator!SwitchOn(swon.did)
        end
        transition -> Thermostat
        event set_temp:human_input?set_temperature
        action do
            tmrature = set_temp.t
        end
}

// Transitions of the composite state Running
transition -> Running
event pollint:human_input?set_polling_interval
action do
        // forward the polling interval instructions to PSM
        request_sensor!set_polling_interval(pollint.intrvl)
end
transition -> Running
```

```
                  event temp:get_sensor?temperature
                        // just discard - should only happen in On or Off

                  // Messages that should not occur, but may occur
                  transition -> Running
                  event human_input?add_thermometer
                  event human_input?add_device
                  action do
                        human_output!prompt("Adding gadgets has been done
and then blocked")
                  end
                  // Messages the cannot occur - since they are handled
                  transition -> Running
                  event human_input?SwitchOn
                  event human_input?SwitchOff
                  event human_input?set_temperature
                  action do
                        human_output!prompt("INTERNAL ERROR: Impossible
messages at PIM.Running")
                  end
            } // end Running
```

Annex 3 Excerpts of the risk analysis of X3C

- The assets
- Malicious threats
- Vulnerabilities
- Treatments

Asset	Explanation
Comfort temperature	That the temperature in the room is within the range intended by its user
The Room is not physically jeopardized	There should be no fire due to overheating, or freeze due to no heating

Threat source	Why? (motive)	Where? (attack p.)	How? (Threat)
Script kiddie	Media attention	Internet to User I/F	Override user command
Black hat hacker	Economic gain	433.92 MHz wireless	Controlling heater for blackmail

How? (Threat)	Exploiting What? (Vulnerability)
Override user command	Web i/f to The Room without authentication
Controlling heater for blackmail	433.92 MHz communication is without encryption and can be sniffed and faked

Incident	Risk Level	Treatment
Uses web i/f to set unpleasant temperatures	High	Apply authentication on web i/f
Turns heater off for potential pipe freeze	Medium	Introduce warnings sent to owner at low temperatures and try and turn the heat on
Turns heater on for potential fire	Medium	Introduce warnings sent to owner at high temperatures and try and turn the heat off