[ITI41120 Applied Computer Science Project](https://www.hiof.no/english/studies/courses/iio/itk/2022/spring/iti41120.html" \t "_blank)

Spring 2023

**Topic**: Verification tool for UML models of reactive systems

**Research areas involved:** Cyber-Physical systems

**Skills needed:** Programming skills in Java, text parsing based on language grammar ([Xtext](https://www.eclipse.org/Xtext/)), interactive graphic editing, some understanding of sequence diagrams and state machines (this task is suitable for those taking ITI42020)

**Number of students needed:** 2

**Contact:** Øystein Haugen ( oystein.haugen@hiof.no )

**Content:** This is an implementation project that involves:

* Making a supporting tool for students of modeling in the field of IoT systems and other real-time systems with concurrency
* We want a tool for supporting the developer in comparing each lifeline of the sequence diagram with a corresponding state machine.
  + The method has been established and is explained in detail in ITI42020; the tool goal is to be helpful in performing the comparison
  + The tool will be very useful for students of ITI42020

Some references:

1. <https://plantuml.com/> is the starting point, and at the bottom of that page there is a link to an online interpreter of PlantUML.
2. <https://plantuml.com/sequence-diagram> (how to make graphic sequence diagrams by writing text)
3. <https://plantuml.com/state-diagram> (how to make graphic state machines from plantuml text, but this will be generated automatically by ThingML)
4. <https://github.com/TelluIoT/ThingML> ThingML is the software we use in ITI42020
5. Xtext on Eclipse <https://www.eclipse.org/Xtext/>
6. <https://github.com/Systems-Modeling/SysML-v2-Release> (SysML v2 is the upcoming standard for System Modeling by OMG and their pilot implementation also applies PlantUML as middleware for graphics. This may hopefully be a future target for the tool. Østfold University College participates in standardizing SysML v2)

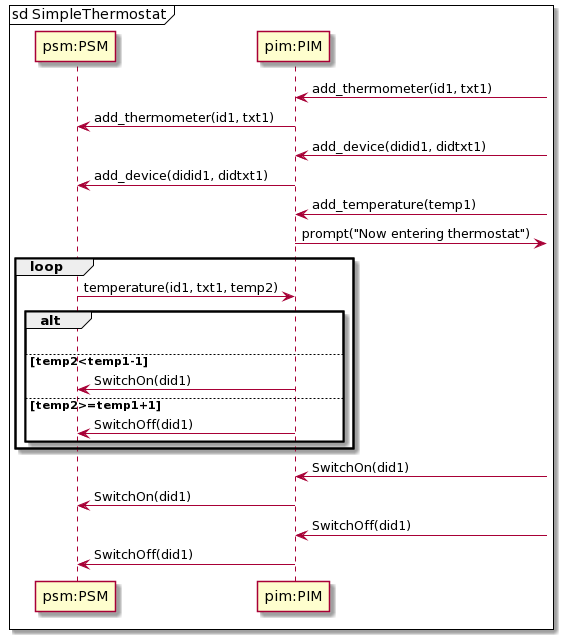
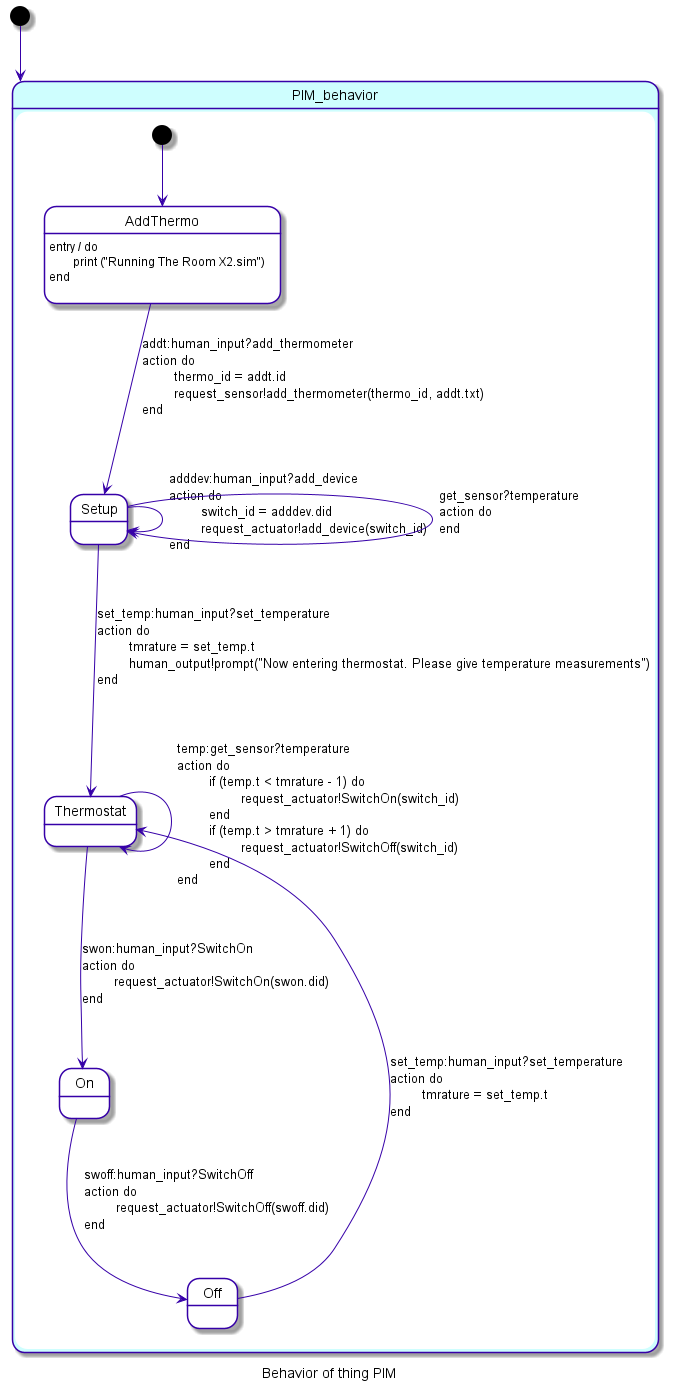
 

Figure 1 A Sequence Diagram generated from PlantUML text and a state machine generated by ThingML

The text that generated the sequence diagram is this:

@startuml

mainframe sd SimpleThermostat

participant "psm:PSM" as psm

participant "pim:PIM" as pim

pim <-] : add\_thermometer(id1, txt1)

pim-> psm: add\_thermometer(id1, txt1)

pim <-] : add\_device(didid1, didtxt1)

pim-> psm: add\_device(didid1, didtxt1)

pim <-] : add\_temperature(temp1)

pim ->] : prompt("Now entering thermostat")

loop

psm -> pim: temperature(id1, txt1, temp2)

alt

|||

else temp2<temp1-1

pim -> psm: SwitchOn(did1)

else temp2>=temp1+1

pim -> psm: SwitchOff(did1)

end

end

pim <-]: SwitchOn(did1)

pim -> psm: SwitchOn(did1)

pim <-]: SwitchOff(did1)

pim -> psm: SwitchOff(did1)

@enduml