UML for Embedded Systems

Exam FALL 2024

Software of a Smart Greenhouse Automation System

Ludovic Apvrille

ludovic.apvrille@telecom-paris.fr

http://perso.telecom-paris.fr/apvrille/UMLEmb/

During an exam, you are not supposed to talk with anyone else, by any means (including mobile phones, chat, etc.), see assignment A below.

A grade is provided for each question. 1 bonus point is awarded for writing quality (report and models).

1 Objective

Your objective is to model the **software** of the main controller of a Smart Greenhouse Automation System.

You have exactly 3 hours to model this system and answer various questions: the time is very short. This means that **you have to make modeling assumptions**. **Keep your diagrams simple and readable**, in particular the analysis diagrams.

Your grade takes into account your report and your models. At the end of the exam, **reports** (in **pdf** format) and one **model** file (in **TTool XML** format) **must be sent to me by email**, with Alexia Cepero and Romain Duplan in cc. The report is optional if you decide to put all explanations in your TTool model. If you were to send me a report, the report should contain explanations concerning your models, as well as relevant screen captures of models (e.g., interesting simulation traces, formal verification results).

2 System specification

Again, the system to model is the software of the main controller of a Smart Greenhouse Automation System.

2.1 Description

2.1.1 Overall description

The Smart Greenhouse Automation System is designed to monitor and regulate the environment inside a greenhouse, ensuring optimal conditions for plant growth. It continuously collects data from sensors measuring temperature, humidity, soil moisture, and CO2 levels. Based on this data, it automatically controls actuators such as ventilation fans, heaters, humidifiers, and an irrigation system. If the temperature exceeds a predefined threshold, cooling fans must activate within 500 milliseconds to prevent overheating, while heaters should respond within one second to maintain a stable climate. The irrigation system waters

plants when soil moisture falls below a critical level, requiring a maximum delay of 2 seconds to initiate water flow.
Users can interact with the system through a touchscreen control panel inside the greenhouse or remotely via a mobile app, which provides real-time monitoring with updates every 5 seconds and historical data visualization. Alerts are sent via SMS or email within 3 seconds of detecting a critical deviation. The system is powered by a microcontroller with optional solar panel integration, ensuring energy efficiency. Additionally, an AI-based module may analyze trends and optimize resource usage, requiring processing times under 500 milliseconds per analysis cycle to avoid delays in decision-making.
Manual overrides allow users to adjust settings or activate devices as needed, with response times under 1 second for direct control commands. The system operates in real-time, demanding low-latency communication between sensors and actuators, and a 99.9% reliability rate to prevent failures. Security measures, including encrypted data transmission and access authentication, ensure only authorized users can modify system settings. Designed to be scalable, it can accommodate additional sensors and features, adapting to different greenhouse sizes and crop types.

3 Assignments

A. Personal work

Mandatory: Recopy the following text at the beginning of your report or as a note in your TTool model.

```
I pledge on my honor that I will not
receive any unauthorized help on this
exam, that I will not help others in any
way on this exam, and that all my
answers will be my own personal work.
```

B. Assumptions

1. Your assumptions should be clear. Do list them in the report: that list might evolve according to the models you make afterwards. Make a clear separation between environment and system assumptions. [2 points]

C. Requirements

1. Create a requirement diagram. [3 points]

D. Analysis

1. Make a use case diagram. [3 points]
2. Continue the analysis in the form you want: activity diagrams, nominal scenario, error scenarios, . . . : you are free to use the diagrams you want. Of course, the idea here is to show important points of the specification. [3 points]

E. Design and validation

1. Make a block diagram. Put the emphasis on which blocks are used to model the system being designed, and which ones are used either to model the environment, or to prove properties (observers). [2 points]
2. Draw state machines, and provide a nominal simulation trace, as well as an error trace. [3 points]
3. Select one timing constraint of your choice and prove that is is always satisfied (or not always satisfied). Also, from requirements, define a property of your choice, and prove whether it is satisfied (or not!). And obviously, explain how you have modeled these two properties [3 points]

Good luck!