UML for Embedded Systems

Exam FALL 2022
Software of a Digital Augmented Hand Cart

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 question I.

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 a Digital Augmented Hand Cart.

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 **models** (in **TTool XML** format) **must be sent to me by email**, with Alexia Cepero in cc. 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 a Digital Augmented Hand Cart, as described below. This specification was automatically generated using *ChatGPT3*.

2.1 Description

2.1.1 Overall description

The role of the software in a Digital Augmented Hand Cart specifically designed to react in real-time to protect the user in case of dangerous misuse and provide information about the battery level and obstacles would likely include the following specifications:

- Obstacle Detection and Avoidance: The software should be able to detect and classify obstacles in the cart's path using data from various sensors such as cameras and lidar. The software should then plan and execute a safe path around the obstacles in real-time.
- Dangerous Misuse Detection: The software should be able to detect dangerous misuse of the cart, such as attempting to use it in an unauthorized area or attempting to overload it, and respond in real-time by triggering an emergency stop or alarm.

- Real-time Information: The software should provide the user with real-time information about the battery level and obstacles to be avoided, via the user interface at least every 30 seconds.
- Human-Machine Interface: The software should provide an intuitive interface for the user to control the cart and monitor its status.
- Emergency Stop: The software should be able to trigger an emergency stop of the cart in response to inputs from the sensors, dangerous misuse detection or user input.
- Security: The software should be designed to protect the cart from unauthorized access or hacking.
- Privacy: The software should not collect or store any personal information without the user's consent.
- Data Management: The software should be able to manage and store the data generated by the sensors and the cart's operations.
- Compliance: The software should comply with relevant standards for software development and testing.
- Maintenance: The software should have a system for monitoring and updating the software to ensure that it continues to function correctly and securely over time.

3 Assignments

I. Personal work

1. Recopy the following text at the beginning of your report (this is mandatory)

```
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.
```

II. 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]

III. Requirements

1. Create a requirement diagram. [3 points]

IV. 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]

V. 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. Prove that an emergency situation always results in a warning to the user. 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!