UML for Embedded Systems

Exam FALL 2020
Software of a platooning system

Ludovic Apvrille

ludovic.apvrille@telecom-paris.fr

http://soc.eurecom.fr/UMLEmb/

During an exam, you are not supposed to talk with anyone else, by any means (including mobile phones, chat, etc.), see question I. Your microphone and your camera must be on at any time.

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 vehicle of a platooning system**[1].

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 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 part of automotive embedded software that handles platooning systems.

2.1 Description

2.1.1 Overall description

Platooning is a transportation technique that consists in grouping trucks or vehicles together to reduce CO2 emissions. A platoon consists of one or several vehicles, the first one in the platoon playing the role of the platoon leader, the other ones playing the role of followers.

1. A vehicle can create a platoon: this vehicle is then the leader of this platoon. This vehicle informs neighbour cars about this platoon by sending a platoon information message (position, speed, acceleration) every second. Once followers have joined, it

[1] https://en.wikipedia.org/wiki/Platoon_(automobile)

regularly informs —every half second — the followers of its current situation (speed, acceleration, direction, selected lane). Whenever there is an important modification of speed / acceleration / direction / lane, the leader immediately informs the followers.

2. A follower can join a platoon only at the last position, i.e. behind all other vehicles of the platoon. When it joins the platoon, it informs the leader about this. When a follower wishes to leave the platoon, it informs all other vehicles of the platoon (with a "leave" message) and then brakes or changes of lane.

3. Leaders and followers use front and back cameras to detect the lanes and the distance to other vehicles. The distance between vehicles within a platoon is considered to be between a min and a max distance. If there is less than the min distance between two vehicles, then the first vehicle detecting this situation broadcasts the information to all others and all vehicles of the platoon must perform an emergency braking. If the distance between two vehicles v1 and v2 —with v1 before v2— gets over max, then v2 and all the vehicles behind v2 have to leave the platoon. v2 is assumed to send the "leave" message. Obviously, one important goal of the platooning software is to keep the inter-vehicle distance between min and max so as to ensure that the platoon works for a long time.

In a more advanced version of the platooning system, the platoon can split i.e. a given follower can decide to become the leader of all the followers behind it.

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 a distance below *min* always result in an emergency braking in all vehicles of the platoon. Last, from requirements, define a property of your choice, and prove whether it is satisfied (or not!). And obviously, explain how you have modeled those properties [3 points]

Good luck!