Exam
UML For Embedded Systems - UMLEmb
Fall 2012

Ludovic Apvrille
ludovic.apvrille@telecom-paristech.fr

November, 30th, 2012

Authorized documents: lecture slides, notes you've taken during lectures, lab sessions results.

Each question is graded. 1 additional point is given as a general appreciation, including written skills and readability.

Also, do not spend more than 15 minutes on question **1** since the modeling exercise is long to complete. At last, consider **making assumptions** on the system so as to reduce the modeling work.

1 Understanding UML diagrams (3 points, ∼15 minutes)

You probably have noticed an important characteristic of UML is that it relies on an object-oriented modeling language.

(a) Why isn't it a good idea to connect ports of classes in class diagrams? Why do we really need to do this in composite structure diagrams? Use examples to illustrate your point.

2 Modeling exercise (16 points, ∼100 minutes)

The goal of this exercise is to model a drone software application in charge of autonomously piloting a drone. This application is made of two subparts: one running in the drone, and one running in a remote computer. The time being limited, you may omit modeling details of your choice, but if you do so, clearly mention which ones you

have decided to omit, and why. At last, do not forget to comment your diagrams, grading takes into account as much diagrams as related comments. The application that you have to model is described with the following simplified specification.

The drone is expected to follow a red line located on the floor. It is assumed that it is stabilized with another embedded application that is not meant to be designed. The drone has two cameras: a front camera, and a bottom camera producing 640×480 *pictures at 30 frames per second. Both video streams are wirelessly sent - with WIFI - to a remote computer on which a video stream analyzer runs. Signs can be placed along the red line to tell the drone to turn right, left or to continue forward at the next crossing. Also, signs can be placed on doors to signal the drone that the door is closed.*

A given path to be followed by the drone is first entered in the remote computer (e.g., turn right at the first crossing, then go forward at the next, etc.). The analyzer running on the computer decodes the two video streams and recognizes the lines and signs to follow the pre-entered path. Flight orders (turn right/left of a given angle, go forward, takeoff, land) are sent back to the drones once the pictures have been analyzed. The application which is onboard of the drone forwards the orders to the stabilizing application. When it reaches its destination, the drone lands. If the PC is loaded, then only a subset of pictures of the two video streams shall be analyzed so as to keep the latency as low as possible. In particular, orders shall be sent back to the drone at least every 10ms.

Each time a problem occurs while the drone is following a path, the drone should simply immediatly land.

2.1 Analysis (∼60 minutes)

(a) Make the use case diagram of this application. (3 points)

(b) Make two scenarios, one for the nominal case, and one for a non-regular case. (2 points)

(c) Propose a collection of classes and objects (or SysML blocks) for this system. (1 point)

(d) Refine the two previously performed scenarios. (3 points)

2.2 Design (∼40 minutes)

(a) Propose a design. It may either be based on a UML class diagram and a composite structure diagram (i.e., UML design), or on a SysML internal block diagram (i.e., a SysML design). (3 points)

(b) Make the state diagram of the most important class (or block) of your system. (4 points)