UML for Embedded Systems

Exam FALL 2016
Lunch Seeker

Ludovic Apvrille

ludovic.apvrille@telecom-paristech.fr

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

During an exam, you are not supposed to talk with someone else, by any means (including mobile phones, chat, etc.). Access to Internet is restricted to the website of the UMLEmb course only. You may consult your own UML/SysML models made in the scope of the labs, but not other models. Electronic devices are not allowed at all, apart from your computer ;-).

A grade is provided for each question. 1 bonus point is given for the writing part.

1 Objective

Your objective is to model the **software of the mobile applications for a restaurant system**. The specification of the system is available online[1], however you are asked to only model what is provided in this document.

You have exactly 3 hours to model this system, and answer various questions: the time is very short. This means that **you have to take 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**. Also, **the report must be printed** and **given to Alexia Cepero right after the end of the exam session**. 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

2.1 Description

2.1.1 Overall description

This system will consist of two parts: one mobile application and one web portal. The mobile application will be used to find restaurants and view information about them while the web portal will be used for managing the information about the restaurants and the system as a whole. The mobile application contains a GPS sub-application, which communicates with a physical GPS device to find the location of the user.
The GPS will provide the mobile application with locations of both the user and the restaurants and the distance between them, but it will also provide maps and the functionality to display the application's data on the map. The functionality provided by the GPS will be

[1] http://www.cse.chalmers.se/ feldt/courses/reqeng/examples/srs_example_2010_group2.pdf

embedded into the application in order for the user to be able to use the functions in the application in a seamlessly manner.
Since this is a data centric product it will need somewhere to store the data. For that, a database will be used. Both the mobile application and web portal will communicate with the database, however in slightly different ways. The mobile application will only use the database to get data while the web portal will also add and modify data. All of the database communication will go over the Internet. The mobile application has some restrictions about the resource allocation. To avoid problems with overloading the operating system the application is only allowed to use 20 megabytes of memory while running the application. The maximum amount of hard drive space is also 20 megabytes

2.1.2 Product functions

With the mobile application, the users will be able to search for restaurants with different criteria (price, type of food, etc.). The result of the search will be viewed either in a list view or in a map view, depending on what criteria included in the search. The list view will have one list item for each restaurant matching the search criteria and show a small part of the restaurant information so the user can identify the restaurant. The map view will show each restaurant location as a pin on the map as well as the user's own location. In both views, the users will be able to either select a restaurant as the target destination, or get information on how to get there. The web portal will provide functionality to manage the system and the restaurant information. It will also provide information about the system, for example show when there is a new update.

2.1.3 User characteristics

There are three types of users that interact with the system: users of the mobile application, restaurant owners and administrators. Each of these three types of users has different use of the system so each of them has their own requirements. The mobile application users can only use the application to find a restaurant. This means that the user have to be able to search for restaurants, choose a restaurant from that search and then navigate to it. In order for the users to get a relevant search result, there are multiple criteria that the users can specify, and results must match all of them.
The restaurant owners will not use the mobile application but the web portal instead. There they will manage the information about their restaurant, for example a description of the restaurant, contact information and their menu. The administrators also only interact with the web portal. They are managing the overall system so there is no incorrect information within it. The administrator can manage the information for each restaurant as well as the options for both the mobile application users and the restaurant owners.

2.1.4 Assumptions

One assumption about the product is that it will always be used on mobile phones that have enough memory.

3 Assignments

I. Assumptions

1. Your assumptions should be clear. Do list them in the report: that list might evolve according to the models you will make afterwards. [2 points]

II. Requirements

1. Create a requirement diagram. [3 points]

III. 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]

IV. Design and validation

1. Make a block diagram. Put the emphasis on which blocks are used to model the system to design, and which ones are used either to model the environment, or to prove properties (observers). [3 points]
2. Draw state machines, and provide a nominal simulation trace, as well as an error trace. [3 points]
3. Prove that whenever a search is performed, either a result is provided, or an error is returned to the user. Also, from requirements, pick up 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, have fun!