ORDRE DES INGÉNIEURS DU QUÉBEC

MAY 2014 SESSION

All documentation allowed

Pocket calculator : authorized models only

Duration of examination : 3 hours

14-LO-A3 Software design

Context. The following text describes the main functionalities of the *AutoPilot* software system.

The autopilot software is an application that can be installed in a car and which allows to control automatically the operation of the vehicle. The interface of the autopilot system must be designed to operate using a touch screen. All the options of the system are displayed as buttons which can be selected by the user by pressing the screen. The software enables a user to:

1- Identify himself to the software by using either a smart key, a user name and a password, or a fingerprint or retinal capture.
2- Determine its current geographical position.
3- Save geographical locations and points of interest (petrol station, tourist site, restaurant, etc.) in lists of favorites.
4- Activate the automatic navigation system.
5- Select a destination by using the system location finder or a destination in one of the lists of favorites.
6- Confirm starting the vehicle and begin the journey to a selected destination.
7- Activate the automatic determination of breaks at regular intervals during the voyage, whose positions are selected based on one or more criteria, including:
    a) The touristic interest of the place,
    b) The presence of a province or State rest area,
    c) The presence of a restaurant,
    d) The availability of fuel.
8- Stop the current navigation to a destination and choose from:
    a) Select a new destination,
    b) Park the vehicle as soon as possible,
    c) Navigate towards an automatically determined point of interest.
9- Resume interrupted navigation.
10- Disable the automatic navigation system.
11- Choose automatic driving settings in order to promote certain criteria of driving, such as:
    a) Maximize energy savings,
    b) Minimize distance of travel,
    c) Avoid toll roads,
    d) Avoid traffic jams, or
    e) Maximize the beauty and interest of landscapes.
12- Choose the display mode of the navigation system amongst:
    a) Tourist information (places of interest, information on geography, etc.)
    b) Navigation information (distance traveled and remaining, time from start, estimated time of arrival, etc.),
    c) Information on the state of the vehicle (remaining quantity of fuel, average consumption, etc.)

The *AutoPilot* system also enables a user with administrator privileges to:

13- Create, configure, and delete user accounts for a list of drivers.
14- Configure driving parameters, such as:
    a) The maximum speed at which the vehicle can travel,
    b) The minimum distance that the vehicle can approach the vehicle preceding it,
    c) The amount of fuel remaining in the tank at which a supply alarm is reported to the user,

d) The list of favorite or excluded oil companies.

15- Configure parameters for connection to telecommunication networks over local area networks (WLAN) or phone (3 G).
16- Enable or disable automatic connections to telecommunications networks.
17- Update the *autopilot* system by connecting to a server through a network connection or from a file stored on a USB flash drive connected to the system.

The *AutoPilot* system allows an employee in charge of maintenance of the vehicle to:

18- Run tests on the navigation system.
19- Retrieve the diagnostic records of the navigation system in the form of a file on a USB stick plugged into the system.
20- To fully reset the navigation system.

Question 1 - Design Process and Use-Cases (24 points)

Modern software development processes are based on the elaboration of use-cases of the software system to be conceived.

a) In the context of the use cases, explain what is a generalization relationship between two use-cases, and give an example from the *AutoPilot* system where such a generalization relationship would be appropriate. Justify your answer. (4 points)
b) Draw an UML use-case diagram of the *AutoPilot* software. (12 points)
c) For **a single use-case of your choice**, among those identified in the use-case diagram, write :
   i. The use-case description in a high-level format. (4 points)
   ii. The main scenario of the use-case. (4 points)

Question 2 – Logical architecture (16 points)

Modern software systems are developed in accordance to layered or multi-tier architectural principles.

- Indicate two important advantages of logically decomposing a software into a layered or multi-tier architecture (4 points),
- In the course of logical architectural design, Larman suggests the use of the *Controller* pattern. Explain the role of the *Controller* and state the advantages brought by its use. (2 points),
- Propose a logical multi-tier architectural decomposition of the *AutoPilot* application. Provide your answer in the form of an annotated UML package diagram (10 points).

Question 3 – Object-Oriented Analysis and Design (30 points)

a) In the context of an object-oriented design, define in your own words the following terms :
   i. Operation (1 point),
   ii. Method (1 point),
   iii. Interface (1 point).
b) What distinction do you make between the three concepts defined above? (2 points)
c) Restricting yourself only to the functionality and concepts identified in the description of the *AutoPilot* system above, propose a UML class diagram of the *AutoPilot* application (25 points).

Question 4 - Design Patterns (30 points)

During the analysis and design phase of the *AutoPilot* application, the following three design problems have been identified, which you are asked to solve using software design patterns:

1. When the user configures the driving parameters of the *AutoPilot* application (requirement #11, in the context), there are several algorithms for determining the optimal route based on the choice of parameters expressed by the user. We wish that these algorithms be interchangeable and the choice of a particular algorithm to have no direct impact on the behavior of the object responsible for managing the journey.
2. In the context of the software update function of the *AutoPilot* system through a network connection (requirement #17 in the context), we wish that the *AutoPilot* application communicate with the server through a local object that encapsulates communications with the server and isolates the *AutoPilot* system from communication details.
3. During a journey guided by the automatic navigation system, the user can choose different information display modes (requirement #12 in the context). As the vehicle moves, changes position and consumes fuel, navigation information display must be updated automatically whenever the navigation information changes.

For each of the three design problems, you are asked to:

a) Identify the appropriate design pattern (from *Gamma et al.*) that should be used to solve the design problem and justify your choice (2 points),
b) Apply the design pattern by providing an annotated UML class diagram, where roles present in the pattern are explicitly associated to the appropriate classes (4 points),
c) Discuss advantages, drawbacks and consequences of using the chosen pattern to solve the design problem (4 points).