习题参考答案

Exercises(Homework): P25

1.3 What are the four important attributes that all professional software should have? Suggest four other attributes that may sometimes be significant.

Answer:

Four important attributes are maintainability, dependability, performance and usability. Other attributes that may be significant could be reusability (can it be reused in other applications), distributability (can it be distributed over a network of processors), portability (can it operate on multiple platforms e.g laptop and mobile platforms) and inter-operability (can it work with a wide range of other software systems).

Decompositions of the 4 key attributes e.g. dependability decomposes to security, safety, availability, etc. is also a valid answer to this question.

2.1

Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:

- A system to control anti-lock braking in a car
- A virtual reality system to support software maintenance
- A university accounting system that replaces an existing system
- An interactive travel planning system that helps users plan journeys with the lowest environmental impact

Answer:

1. Anti-lock braking system This is a safety-critical system so requires a lot of up-front analysis before implementation. It certainly needs a plan-driven approach to development with the requirements carefully analysed. A waterfall model is therefore the most appropriate approach to use, perhaps with formal transformations between the different development stages.

2. Virtual reality system This is a system where the requirements will change and there will be an extensive user interface components. Incremental development with, perhaps, some UI prototyping is the most appropriate model. An agile process may be used.

3. University accounting system This is a system whose requirements are fairly ell-known and which will be used in an environment in conjunction with lots of other stems such as a research grant management system. Therefore, a reuse-based proach is likely to be appropriate for this.

4. Interactive travel planning system System with a complex user interface but which must be stable and reliable. An incremental development approach is the most appropriate as the system requirements will change as real user experience with the system is gained.

2.4

Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process.

Answer:

There is a fundamental difference between the user and the system requirements that mean they should be considered separately.

1. The user requirements are intended to describe the system's functions and features from a user perspective and it is essential that users understand these requirements. They should be expressed in natural language and may not be expressed in great detail, to allow some implementation flexibility. The people involved in the process must be able to understand the user's environment and application domain.

2. The system requirements are much more detailed than the user requirements and are intended to be a precise specification of the system that may be part of a system contract. They may also be used in situations where development is outsourced and the development team need a complete specification of what should be developed. The system requirements are developed after user requirements have been established.

Excercises(Homework): P116

4.2, *4.4

4.2

Discover ambiguities or omissions in the following statement of requirements for part of a ticket-issuing system:

An automated ticket-issuing system sells rail tickets. Users select their destination and input a credit card and a personal identification number.

The rail ticket is issued and their credit card account charged. When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select a destination. Once a destination has been selected, users are requested to input their credit card.

Its validity is checked and the user is then requested to input a personal identifier. When the credit transaction has been validated, the ticket is issued.

Answer:

Ambiguities and omissions include:

- • Can a customer buy several tickets for the same destination together or must they be bought one at a time?
- • Can customers cancel a request if a mistake has been made?
- • How should the system respond if an invalid card is input?
- • What happens if customers try to put their card in before selecting a destination (as they would in ATM machines)?
- • Must the user press the start button again if they wish to buy another ticket to a different destination?
- • Should the system only sell tickets between the station where the machine is situated and direct connections or should it include all possible destinations?

4.4

Write a set of non-functional requirements for the ticket-issuing system, setting out its expected reliability and response time.

Answer:

Possible non-functional requirements for the ticket issuing system include:

1. Between 0600 and 2300 in any one day, the total system down time should not exceed 5 minutes.
2. Between 0600 and 2300 in any one day, the recovery time after a system failure should not exceed 2 minutes.
3. Between 2300 and 0600 in any one day, the total system down time should not exceed 20 minutes.

All these are availability requirements - note that these vary according to the time of day. Failures when most people are traveling are less acceptable than failures when there are few customers.

4. After the customer presses a button on the machine, the display should be updated within 0.5 seconds.
5. The ticket issuing time after credit card validation has been received should not exceed 10 seconds.
6. When validating credit cards, the display should provide a status message for customers indicating that activity is taking place. This tells the customer that the potentially time consuming activity of validation is still in progress and that the system has not simply failed.
7. The maximum acceptable failure rate for ticket issue requests is 1: 10000.

Excercises(Homework): P143-144

5.2, 5.5, 5.6, 5.7

5.2

How might you use a model of a system that already exists? Explain why it is not always necessary for such a system model to be complete and correct. Would the same be true if you were developing a model of a new system?

Answer:

You might create and use a model of a system that already exists for the following

reasons:

1. To understand and document the architecture and operation of the existing

system.

2. To act as the focus of discussion about possible changes to that system.

3. To inform the re-implementation of the system.

You do not need a complete model unless the intention is to completely document the operation of the existing system. The aim of the model in such cases is usually to help you work on parts of the system so only these need to be modelled. Furthermore, if the model is used as a discussion focus, you are unlikely to be interested in details and so can ignore parts of the system in the model.

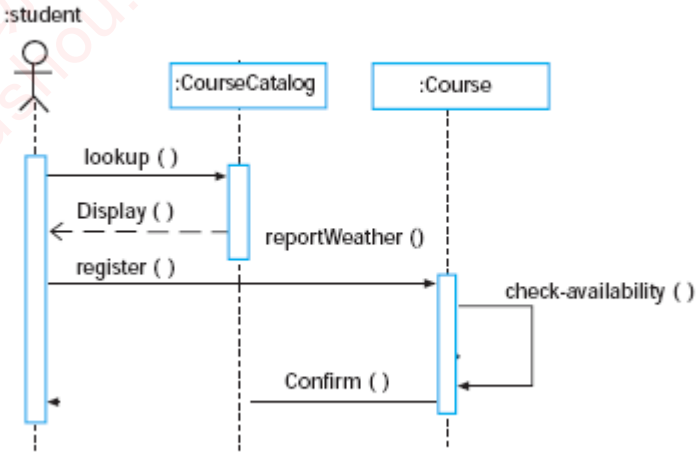
This is true, in general, for models of new systems unless a model-based approach to development is taking place in which case a complete model is required. The other circumstances where you may need a complete model is when there is a contractual requirement for such a model to be produced as part of the system documentation.

5.5

Develop a sequence diagram showing the interactions involved when a student registers for a course in a university. Courses may have limited enrolment, so the registration process must include checks that places are available. Assume that the student accesses an electronic course catalog to find out about available courses.

Answer:

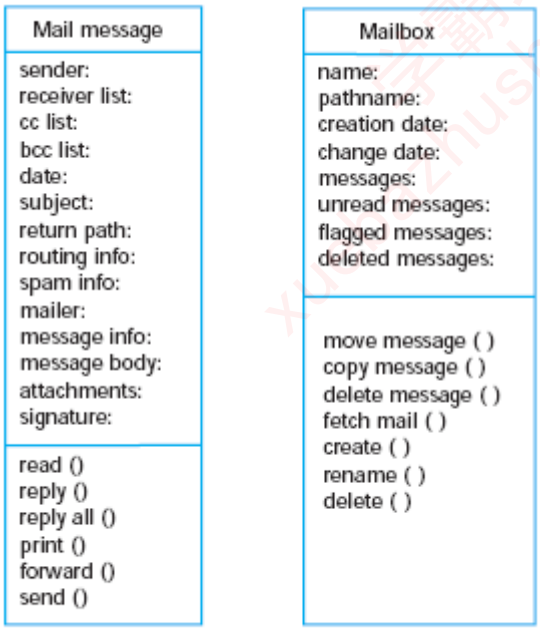
A relatively simple diagram is all that is needed here. It is best not to be too fussy about things like UML arrow styles as hardly anyone can remember the differences between them.



5.6

Look carefully at how messages and mailboxes are represented in the email system that you use. Model the object classes that might be used in the system implementation to represent a mailbox and an e-mail message.

Answer:

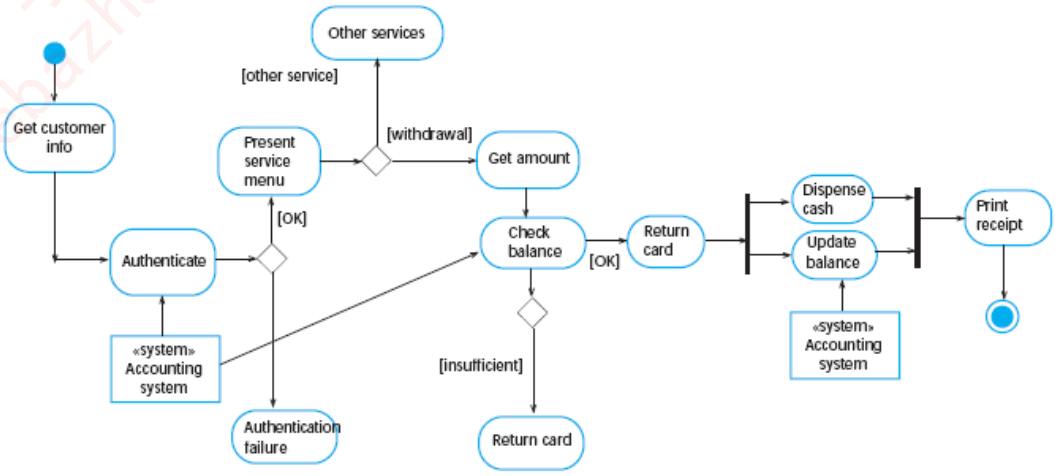


5.7

Based on your experience with a bank ATM, draw an activity diagram that models the data processing involved when a customer withdraws cash from the machine.

Answer:

Notice that I have not developed the activities representing other services or failed authentication.



Excercises(Homework): P173-174

6.1, 6.3, 6.9

6.1

When describing a system, explain why you may have to design the system architecture before the requirements specification is complete.

Answer:

The architecture may have to be designed before specifications are written to provide a means of structuring the specification and developing different subsystem specifications concurrently, to allow manufacture of hardware by subcontractors and to provide a model for system costing.

6.3

Explain why design conflicts might arise when designing an architecture for which both availability and security requirements are the most important non-functional requirements.

Answer:

Fundamentally, to provide availability, you need to have (a) replicated components in the architecture so that in the event of one component failing, you can switch immediately to a backup component. You also need to have several copies of the data that is being processed. Security requires minimizing the number of copies of the data and, wherever possible, adopting an architecture where each component only knows as

much as it needs to, to do its job. This reduces the chance of intruders accessing the data.

Therefore, there is a fundamental architectural conflict between availability (replication, several copies) and security (specialization, minimal copies). The system architect has to find the best compromise between these fundamentally opposing requirements.

6.9

Using the basic model of an information system as presented in Figure 6.16, suggest the components that might be part of an information system that

allows users to view information about flights arriving and departing from a particular airport.

Answer:

Students should consider the levels in the information system and should identify components that might be included at each level. Examples of these components

might be:

Level 1 (Database level)

Flight database; Flight status database; Airport information;

Level 2: (Information retrieval level)

Status management; Flight management; Search;

Level 3: (User interaction level)

Authentication; session management; forms processing ()

Level 4 (User interface)

Input checking (Javascript), browser

Excercises(Homework): P202-203

7.1, 7.3

7.1

7.1 Using the structured notation shown in Figure 7.3, specify the weather station use cases for Report status and Reconfigure. You should make reasonable assumptions about the functionality that is required here.

Answer:

System: Weather station
Use case: Report status
Actors: Weather information system, weather station
Data: The weather station sends a status update to the weather information system giving information about the status of its instruments, computers and power supply.
Stimulus: The weather information system establishes a satellite link with the weather station and requests status information.
Response: A status summary is uploaded to the weather information system
Comments: System status is usually requested at the same time as the weather report.

System: Weather station
Use case: Reconfigure
Actors: Weather information system, weather station
Data: The weather information station sends a reconfiguration command to the weather station. This places it into remote control mode where further commands may be sent from the remote system to update the weather station software.
Stimulus: A command from the weather information system.
Response: Confirmation that the system is in remote control mode
Comments: Used occasionally when software updates have to be installed.

7.3

7.3 Using the UML graphical notation for object classes, design the following object classes, identifying attributes and operations. Use your own experience to decide on the attributes and operations that should be associated with these objects.

- a telephone
- a printer for a personal computer
- a personal stereo system
- a bank account
- a library catalogue

Answer:

There are many possible designs here and a great deal of complexity can be added to the objects. However, I am only really looking for simple objects which encapsulate the principal requirements of these artefacts. Possible designs are shown in the above diagram.