1. Introduction to Software Engineering: Solutions

1-1 What is the purpose of modeling?

The purpose of modeling is to reduce complexity by building a simplified representation of reality which ignores irrelevant details. What is relevant or not is defined by the questions the model will be used to answer.

1–2 A programming language is a notation for representing algorithms and data structures. List two advantages and two disadvantages of using a programming language as sole notation throughout the development process.

Advantages:

- Developers need only learn one notation for all development activities.
- Traceability among models and between models and code is made easier since they are written in the same notation.

Disadvantages:

- A programming language is a low level notation which is difficult to use for representing user requirements, for example.
- A programming language enables and encourages developers to represent implementation details too early.

1–3 Consider a task you are not familiar with, such as designing a zero-emissions car. How would you attack the problem?

This is an open ended question whose purpose is for students think about problems they cannot solve without help. Answers should contain two or more of the following points:

- Define the problem precisely by gathering information from potential users.
- Discover the boundaries of the solution space by gathering information from application domain experts.
- Brainstorm ideas with other people, including experts and non experts
- Evaluate ideas using prototypes, simulations, and candidate users.

1–4 What is meant by "knowledge acquisition is not sequential"? Provide a concrete example of knowledge acquisition that illustrates this.

Knowledge acquisition is nonlinear in the sense that the acquisition of a new piece of knowledge may invalidate prior knowledge. In other terms, knowing one more piece of information may lead you to realize that what you thought you knew is invalid. Galileo Galilei invalidated the earth centric model of the universe by observing the moons of Jupiter and the phases of Venus.

1–5 Hypothesize a rationale for the following design decisions:

This exercise tests if the student understands the difference between a decision and its rationale. The exact rationale provided by the student is not important as long as it is rationale (e.g., the answer to the first bullet could have been to allow snow white's seven dwarves to purchase train tickets).

- *"The ticket distributor will be at most one and a half meters tall."*

Enable children and persons in wheelchair to purchase tickets.

- *"The ticket distributor will include two redundant computer systems."*

To achieve a high level of availability such that ticket distribution is not interrupted (and thus, ticket sales not lost in the case of the failure of one computer).

- *"The ticket distributor will include a touch screen for displaying instructions and inputing commands. The only other control will be a cancel button for aborting a transaction."*

Enable substantial modifications to the interface (e.g., increase the number of tariff zones or the number of products) without changes to the hardware.

1–6 Specify which of the following statements are functional requirements and which are nonfunctional requirements:

- *"The ticket distributor must enable a traveler to buy weekly passes."*
- *"The ticket distributor must be written in Java."*
- *"The ticket distributor must be easy to use."*

The first requirement is functional, the third is nonfunctional. Using the definitions in Chapter 1, the second requirement is nonfunctional. In Chapter 4, we will call this requirement a pseudo requirement as it constrains aspects of the system that are not visible to the user.

1–7 Specify which of the following decisions were made during requirements or system design:

- *"The ticket distributor is composed of a user interface subsystem, a subsystem for computing tariff, and a network subsystem managing communication with the central computer."*
- *"The ticket distributor will use PowerPC processor chips."*
- *"The ticket distributor provides the traveler with an on-line help."*

The first decision is a system design decision. The second decision is also a system design decision if made by developers (otherwise, it is a requirements decision). The third decision is a requirements decision.

1–8 In the following description, explain when the term account is used as an application domain concept and when as a solution domain concept:
"Assume you are developing an online system for managing bank accounts for mobile customers. A major design issue is how to provide access to the accounts when the customer cannot establish an online connection. One proposal is that accounts are made available on the mobile computer, even if the server is not up. In this case, the accounts show the amounts from the last connected session."

The first two occurrences of "account" are application domain concepts while the last two occurrences are solution domain concepts. The phrases "accounts are made available on the mobile computer" and "the accounts show the amounts from the last connected session" denote a solution domain concept that gives users the illusion that they are accessing their bank accounts on their mobile computer. However, the actual bank account is not on the mobile computer.

1–9 What is the difference between a task and an activity?

An activity is composed of a number of tasks. Both represent work, but tasks cannot conveniently be decomposed any further.

1–10 A passenger aircraft is composed of several millions of individual parts and requires thousands of persons to assemble. A four-lane highway bridge is another example of complexity. The first version of Word for Windows, a word processor released by Microsoft in November 1989, required 55 person-years, resulted into 249,000 lines of source code, and was delivered 4 years late. Aircraft and highway bridges are usually delivered on time and below budget, whereas software is often not. Discuss what are, in your opinion, the differences between developing an aircraft, a bridge, and a word processor, which would cause this situation.

This is an open question whose purpose is to have students realize that software systems are not the only complex systems out there. Answers should include two or more of the following points:

- To estimate the budget and schedule for a new bridge or aircraft, engineers use actual data from previous bridges and aircraft. Word for Windows was an innovative piece of software with few or no precedents.
- Many bridges and aircraft are simply refinements of other existing artifacts. This reduces the proportion of the overall effort that is dedicated to design (which is the most difficult to estimate).
- Bridges and aircraft are often associated with severe financial penalties when late or over budget.

- Bridges and aircraft have safety requirements associated with them. This leads to a conservative approach to development including the use of mature technologies and well defined processes.
- Bridges and aircraft are sometimes delivered late too.

