Information Systems Analysis & Design (M8748)

Tutorial 7 Answer

1. Explain what is meant by ‘use case realization’.
   An activity that moves by stages from a use case to the implementation of a set of software classes that adequately fulfills the requirements of the use case.

2. Why are requirements models in IS development neither entirely graphical nor entirely textual?
   A purely graphical model does not have the precision to cover every aspect of system behavior and structure. A purely textual model would be too large, too complex and extremely difficult to understand and to maintain.

3. Distinguish between attribute and value.
   An attribute is a characteristic of a class (every person has a height). An attribute value is a characteristic of an instance (this author is 1.75m tall).

4. In what sense are classes generally more stable than their instances, and why is this usually the case?
   An element’s stability is the relative infrequency of change in its description. Changes in the description of a class (defined by its attributes, operations and associations) typically occur only due to a change in requirements. Instances may change frequently during software execution, e.g. by being created or destroyed in response to events, or by changes in attribute values or links.

5. Distinguish between link and association.
   A link models a relationship between two specific object instances. An association models all the possible links between instances of the associated classes.

6. What is multiplicity, and why can it be called a constraint?
   Multiplicity denotes the range of permitted cardinalities for a class—i.e. the number of objects that can be linked to a single object of that class by a specific association. It is a constraint because it limits the behavior of a system. If a client can have only one staff contact, it should be impossible to link a second.

7. What is an operation?
   An operation represents something a single object can do. For example, the StaffMember

class has an operation changeGrade()—this represents the ability of the object to change its recorded grade in order to reflect a change in grade for the real member of staff. Each operation represents one part of the overall functionality of a use case.

8. How are operations related to messages?
   Each operation is triggered by one or more messages. An operation cannot run until it is 'called' by a valid message.

9. What is an attribute?
   An attribute represents something an object can know. E.g. the StaffMember class has an attribute staffStartDate—this represents the ability of the object to know the start date of the real member of staff.

10. Explain why Section 7.4.6 makes no mention of updating a link when it is changed, but instead only discusses the creation and destruction of links.
    A link is a connection between two objects. 'Changing' a link (say by substituting another object at one end) is equivalent to destroying the link and creating a new one. (Think about two objects tied with string. In a substitution, there is a moment when neither one nor the other is connected—unless you tie on the second before untying the first, but an object link cannot do this.)

11. What is a collaboration?
    A set of objects (and their relationships) that are collectively capable of interacting to deliver the functionality of a use case. The UML specification gives other definitions that are much more abstract and general. The definition given here expresses the essentials of the concept in this context.

12. How does a collaboration diagram differ from a class diagram?
    A collaboration diagram typically shows instances and links, while a class diagram typically shows only classes and their associations. A collaboration diagram shows only those objects that participate in a specific interaction, while a class diagram typically shows a structure that may be capable of supporting different interactions. A collaboration diagram shows messages between objects, while a class diagram does not.