DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
CS6502-OBJECT ORIENTED ANALYSIS AND DESIGN

UNIT IV
APPLYING DESIGN PATTERNS

System sequence diagrams - Relationship between sequence diagrams and use cases Logical architecture and UML package diagram – Logical architecture refinement - UML class diagrams – UML interaction diagrams - Applying GoF design patterns.

1. What is meant by System Sequence Diagrams? APRIL/MAY-2011

A system sequence diagram (SSD) is a picture that shows, for a particular scenario of a use case, the events that external actors generate their order, and inter-system events. All systems are treated as a black box; the emphasis of the diagram is events that cross the system boundary from actors to systems.

2. What is meant by System Behavior?

System behavior is a description *of what* a system does, without explaining how it does it. One Part of that description is a system sequence diagram. Other parts include the Use cases, and system contracts (to be discussed later).

3. What is meant by Inter-System SSDs?

SSDs can also be used to illustrate collaborations between systems, such as between the Next Gen POS and the external credit payment authorizer. However, this is deferred until a later iteration in the case study, since this iteration does not include remote systems collaboration.

4. Define System Events and the System Boundary.

To identify system events, it is necessary to be clear on the choice of system boundary, as discussed in the prior chapter on use cases. For the purposes of software development, the system boundary is usually chosen to be the software system itself; in this context, a system event is an external event that directly stimulates the software.

5. How to Naming System Events and Operations?

System events (and their associated system operations) should be expressed at the level of intent rather than in terms of the physical input medium or interface widget level. It also improves clarity to start the name of a system event with a verb Thus "enter item" is better than "scan" (that is, laser scan) because it captures the intent of the operation while remaining abstract and noncommittal with respect to design choices about what interface is used to capture the system event.

6. What is meant by interaction diagram?

The term *interaction diagram* is a generalization of two more specialized UML diagram types; both can be used to express similar message interactions:
. Collaboration diagrams
. Sequence diagrams

7. What is meant by link?

A link is a connection path between two objects; it indicates some form of navigation And visibility between the objects is possible . More formally, a link is an instance of an association. For example, there is a link or path of navigation from a *Register* to a *Sale,* along which messages may flow, such as the *make 2 Payment* message.

8. What is meant by Messages?

Each message between objects is represented with a message expression and small arrow indicating the direction of the message. Many messages may flow along this link. A sequence number is added to show the sequential order of messages in the current thread of control.

9. How to create an instance?

Any message can be used to create an instance, but there is a convention in the UML to use a message named *create* for this purpose. If another (perhaps less obvious) message name is used, the message may be annotated with a special feature called a UML stereotype, like so: *«create».* The *create* message may include parameters, indicating the passing of initial values. This indicates, for example, a constructor call with parameters in Java.

10. What is meant by Low Coupling?

Coupling is a measure of how strongly one element is connected to, has knowledge of, or relies on other elements. An element with low (or weak) coupling is not dependent on too many other elements; "too many" is context-dependent, but will be examined. These elements include classes, subsystems, systems, and so on.

11. What is meant by High cohesion?

Cohesion (or more specifically, functional cohesion) is a measure of how strongly related and focused the responsibilities of an element are. An element with highly related responsibilities, and which does not do a tremendous amount of work, has high cohesion. These elements include classes, subsystems, and so on.

12. Define Controller.

Assign the responsibility for receiving or handling a system event message to a class representing one of the following choices:

- Represents the overall system, device, or subsystem *(facade controller).*
- Represents a use case scenario within which the system event occurs, often named <UseCaseName>Handler, <UseCaseName>Coordinator, or <Use-CaseName>Session *(use-case or session controller).*
- Use the same controller class for all system events in the same use case scenario.

-Informally, a session is an instance of a conversation with an actor.

-Sessions can be of any length, but are often organized in terms of use cases (use case sessions).

13. What is meant by CRC card?

CRC cards are index cards, one for each class, upon which the responsibilities of the class are briefly written, and a list of collaborator objects to fulfill those responsibilities. They are usually developed in a small group session. The GRASP patterns may be applied when considering the design while using CRC cards.

14. What is meant by Pure Fabrication?

This is another GRASP pattern. A Pure Fabrication is an arbitrary creation of the designer, not a software class whose name is inspired by the Domain Model. A use-case controller is a kind of Pure Fabrication.

15. List the relationships used in class diagram? APRIL/MAY-2011

Generalization(class to class)
Association (object to object)
Aggregation (object to object)
Composition (object to object)

PART- B

1. How to Adding New SSDs and Contracts?
-New System Sequence Diagrams
-New System Operations
-New System Operation Contracts

2. Explain about Interaction Diagram Notation? **APRIL/MAY-2011**
-Sequence and Collaboration Diagrams
-Collaboration Diagram
-Sequence Diagram
-Common Interaction Diagram Notation
-Basic Collaboration Diagram
-Notation
-Basic Sequence Diagram Notation

3. Design the Model and Creating Design Class Diagrams.
-When to Create DCDs
-Example DCD
-DCD and UP
-Domain Model vs. Design Model Classes
-DCDs, Drawing, and CASE Tools
-DCDs within the UP

4. What are concepts involved in domain refinement?
-Generalization
-Defining Conceptual Super classes and Subclasses
-Class Hierarchies and Inheritance
-Aggregation and Composition
-Examples