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

UNIT III
CASE STUDY

Case study – the Next Gen POS system, Inception -Use case Modeling - Relating Use cases – include, extend and generalization - Elaboration - Domain Models - Finding conceptual classes and description classes – Associations – Attributes – Domain model refinement – Finding conceptual class Hierarchies - Aggregation and Composition

1. What is Inception? APIRL/MAY-2011

Inception is the initial short step to establish a common vision and basic scope for the Project. It will include analysis of perhaps 10% of the use cases, analysis of the critical non- Functional requirement, creation of a business case, and preparation of the development Environment so that programming can start in the elaboration phase. Inception in one Sentence: Envision the product scope, vision, and business case.

2. What Artifacts May Start in Inception?

Some sample artifacts are Vision and Business Case, Use-Case Model, Supplementary Specification, Glossary, Risk List & Risk Management Plan, Prototypes and proof-of-concepts etc.

4. What are the key ideas for Planning the Next Iteration?

Organize requirements and iterations by risk, coverage, and criticality.

5. What is a Domain Model? APIRAL/MAY-2011

A domain model is a visual representation of conceptual classes or real-situation objects in a domain. The term "Domain Model" means a representation of real-situation conceptual classes, not of software objects. The term does not mean a set of diagrams describing software classes, the domain layer of a software architecture, or software objects with responsibilities.

6. **How the domain model is illustrated?**

Applying UML notation, a domain model is illustrated with a set of class diagrams in which no operations (method signatures) are defined. It provides a conceptual perspective. It may show:

- domain objects or conceptual classes
- associations between conceptual classes
- attributes of conceptual classes

7. Why Call a Domain Model a "Visual Dictionary"?

The information it illustrates could alternatively have been expressed in plain text. But it's easy to understand the terms and especially their relationships in a visual language, since our brains are good at understanding visual elements and line connections. Therefore, the domain model is a visual dictionary of the noteworthy abstractions, domain vocabulary, and information content of the domain.

8. What are the elements not suitable in a domain model?

The following elements are not suitable in a domain model

- Software artifacts, such as a window or a database, unless the domain being modeled is of software concepts, such as a model of graphical user interfaces.
- Responsibilities or methods

9. What are Conceptual Classes?

The domain model illustrates conceptual classes or vocabulary in the domain. Informally, a conceptual class is an idea, thing, or object. More formally, a conceptual class may be considered in terms of its symbol, intension, and extension

- Symbol words or images representing a conceptual class.
- Intension the definition of a conceptual class.
- Extension the set of examples to which the conceptual class applies

10. How to Create a Domain Model?

The current iteration requirements under design:
1. Find the conceptual classes (see a following guideline).
2. Draw them as classes in a UML class diagram.
3. Add associations and attributes.

11. How to Find Conceptual Classes?

1. Reuse or modify existing models. This is the first, best, and usually easiest approach, and where I will start if I can. There are published, well-crafted domain models and data models (which can be modified into domain models) for many common domains, such as inventory, finance, health, and so forth. Example books that I'll turn to include Analysis Patterns by Martin Fowler, Data Model Patterns by David Hay, and the Data Model Resource Book (volumes 1 and 2) by Len Silverton.
2. Use a category list.
3. Identify noun phrases.

13. Define Association.

An **association** is a relationship between classes (more precisely, instances of those classes) that indicates some meaningful and interesting connection.

14. Why Should We Avoid Adding Many Associations?

We need to avoid adding too many associations to a domain model. Digging back into our discrete mathematics studies, you may recall that in a graph with n nodes, there can be associations to other nodes a potentially very large number. A domain model with 20 classes could have 190 associations’ lines! Many lines on the diagram will obscure it with "visual noise." Therefore, be parsimonious about adding association lines. Use the criterion guidelines suggested in this chapter, and focus on "need-to-remember" associations.

15**. How to Name an Association in UML?**

Name an association based on a Class Name-Verb Phrase-Class Name format where the verb phrase creates a sequence that is readable and meaningful.

16. What is Aggregation? APRIL/MAY-2011

Aggregation is a vague kind of association in the UML that loosely suggests whole-part relationships (as do many ordinary associations). It has no meaningful distinct semantics in the UML versus a plain association, but the term is defined in the UML.

17. What is composition? APRIL/MAY-2011

Composition, also known as composite aggregation, is a strong kind of whole-part aggregation and is useful to show in some models. A composition relationship implies that 1) an instance of the part (such as a Square) belongs to only one composite instance (such as one Board) at a time, 2) the part must always belong to a composite (no free-floating Fingers), and 3) the composite is responsible for the creation and deletion of its parts either by itself creating/deleting the parts, or by collaborating with other objects.

18. Mention the guidelines that suggest when to show aggregation.

- The lifetime of the part is bound within the lifetime of the composite there is a createdelete dependency of the part on the whole.
- There is an obvious whole-part physical or logical assembly.
- Some properties of the composite propagate to the parts, such as the location.
- Operations applied to the composite propagate to the parts, such as destruction, movement, and recording.

19. What is Elaboration?

Elaboration is the initial series of iterations during which the team does serious investigation, implements (programs and tests) the core architecture, clarifies most requirements, and tackles the high-risk issues. In the UP, "risk" includes business value. Therefore, early work may include implementing scenarios that are deemed important, but are not especially technically risky.

20. What are the tasks performed in elaboration?

- the core, risky software architecture is programmed and tested
- the majority of requirements are discovered and stabilized
- the major risks are mitigated or retired

21. What are the key ideas and best practices that will manifest in elaboration?

- do short time boxed risk-driven iterations
- start programming early
- adaptively design, implement, and test the core and risky parts of the architecture
- test early, often, realistically
- adapt based on feedback from tests, users, developers
- write most of the use cases and other requirements in detail, through a series of workshops, once per elaboration iteration

PART- B

1. Explain about POS generation systems.

-The Next Gen POS System

-Architectural Layers and Case Study Emphasis

-Iterative Development and Iterative Learning

2. Define Inception. Explain about artifacts of Inception
-Inception: An Analogy
-What Artifacts May Start in Inception
- You Didn't Understand Inception When...

3. Write briefly about elaboration and discuss the differences between Elaboration and Inception with examples.
-Iteration 1 Requirements and Emphasis: Core OOA/D Skills
-Inception and Elaboration
-Planning the Next Iteration

4. Illustrate the concept of Domain model with examples. **APRIL/MAY-2011**
-Definitions
-Guidelines for creating domain model
-Examples

5. Explain the guidelines for finding Conceptual Classes with neat diagrams
- Three Strategies
-Find and Draw Conceptual Classes

6. Explain about Aggregations and compositions
-Definitions
-Identify Composition &Aggregations
-Example: the Next Gen Domain Model