LAB 2: Requirements Analysis

1. OBJECTIVES

1.1 Refine Use Case model
1.2 Build the Conceptual Model
1.3 Build the Dynamic Model

2. INTRODUCTION

2.1 Starting from the Use Case descriptions, more details about the system requirements can be uncovered through requirement analysis. By building a model of the application (problem) domain, the team can validate, correct and clarify the requirements.

2.2 Requirements elicitation and analysis are highly iterative activities. Expect to re-visit your Use Case diagram and User Case descriptions to clarify and add detail, as you work through requirements analysis.

2.3 Using the methodology described in Bruegge Chapter 5, the team will develop the Analysis Model consisting of (a) the Object Model, also known as the Conceptual Model, and (b) the Dynamic Model.

2.4 The conceptual model is usually depicted in class diagram, while the dynamic model is usually depicted in sequence diagram and state machine.

2.5 The analysis model forms the basis for system design and is to be refined in Lab 3.

3. PROCEDURE

3.1 *Refine Use Case Model*

3.1.1 Finalize use case diagram with clear definition of actors and use cases

3.1.2 Keep refining user case descriptions. Describe clearly use case precondition, flow of events, and alternative flows.

3.1.3 Expect to re-visit your Use Case diagram and User Case descriptions to clarify and add details as you build the conceptual model and dynamic model

3.2 *Build the Conceptual Model*

3.2.1 From the Use Case Descriptions and the terms in the Data Dictionary, identify the initial entity, boundary and control objects and their attributes.

3.2.2 From your understanding of the classes' responsibilities, apply the stereotypes for entity, boundary or control classes.

3.2.3 In the Class Diagrams, depict these classes and the relationships between them (association, generalization, aggregation).

3.3 *Build the Dynamic Model*

3.3.1 For each use case, model the interaction between the boundary, control and entity classes from the conceptual model, to enact the flow of events in the Use Case Description. Specify the messages and parameters passed between classes in a Sequence Diagram. Use the stereotyped classes to guide the direction and sequence of messages exchanged.

3.3.2 Model the system behavior in a State Machine of UIs based on the UI mockups developed in Lab#1. This state machine of UIs is referred to as a Dialog Map of the system. Refine and further specify your user interface design. See page 420 of Fox for a detailed description of Dialog Maps.

3.3.3 Expect to move back and forth between the Conceptual Model and the Dynamic Model because studying interactions and behavior will reveal additional attributes, operations and relationships of classes.

4. DELIVERABLES

- Complete Use Case diagram
- Use Case descriptions
- Class diagram of entity classes
- Key boundary classes and control classes
- Sequence diagrams of some use cases
- Initial Dialog map

Please submit the deliverables to your SVN repository (under the folder “lab2”) before the Lab#3 starts. Your Lab Supervisor will wish to see and discuss the deliverables with you during Lab #3.

Use Case descriptions, class diagram, sequence diagrams, and dialog map shall be finalized in the Lab#3.

5. REFERENCES

- Introduction to Software Engineering Design: Processes, Principles and Patterns with UML2 – C. Fox
- Object-Oriented Software Engineering: Using UML, Patterns and Java – B. Bruegge and A. Dutoit