OO Analysis and Design Practice Lab

Week 4

## Purpose of Lab : To Think Critically

1. **The following description identifies the business need for a simple university library system. Based on the description, develop the following diagrams:**

**a. Develop a domain class diagram**

**b. Develop a use case diagram**

**c. Do a CRUD analysis to ensure that the identified classes and use cases are consistent. Update both your class diagram and use case diagram as necessary.**

**This case is a simplified (initial draft) of a new system for the University Library. Of course, the library system must keep track of books. Information is maintained about both book titles and the individual book copies. Book titles maintain information about title, author, publisher, and catalog number. Individual copies maintain copy number, edition, publication year, ISBN, book status (whether it is on the shelf or loaned out), and date due back in.**

**The library also keeps track of patrons to the library. Since it is a university library, there are several types of patrons, each with different privileges. There are faculty patrons, graduate student patrons, and undergraduate student patrons. Basic information about all patrons is name, address, and telephone number. For faculty patrons, additional information is office address and telephone number. For graduate students, information such as graduate program and advisor information is maintained. For undergraduate students program and total credit hours are maintained.**

**The library also keeps information about library loans. A library loan is a somewhat abstract object. A loan occurs when a patron approaches the circulation desk with a stack of books to check out. Over time a patron can have many loans. A loan can have many physical books associated with it. (And a physical book can be on many loans over a period of time. Information about past loans is kept in the database.) So, in this case, it is recommended that an association class be created for loaned books.**

**If a book is checked out that a patron wants, he/she can put that title on reserve. This is another class that does not represent a concrete object. Each reservation is for only one title and one patron. Information such as date reserved, priority, and date fulfilled is maintained. When it is fulfilled, the system associates it with the loan on which it was checked out.**

**Patrons have access to the library information to search for book titles and to see whether a book is available. A patron can also reserve a title if all copies are checked out. When patrons bring books to the circulation desk, a clerk checks out the books on a loan. Clerks also check books in. When books are dropped in the return slot, the clerks check them in. Stocking clerks keep track of the arrival of new books.**

**The managers in the library have their own activities. They will print out reports of book titles by category. They also like to see (online) all overdue books. When books get damaged or destroyed, they will delete information about book copies. Managers also like to see what books are on reserve.**

# CRUD Analysis

ListUse cases and domain classes in a Use case-domain Class Matrix.

The matrix shows which use case requires access to each domain class.

This information will be needed when designing the object interactions for each use case.

The cells of the matrix show what the use case does to the domain class. The letter C means the use case creates new data, R means the use case reads data, U means the use case updates data , and the D means the use case might delete data.

Another verification technique is to verify that there are use cases that are responsible for creating updating, and deleting each domain class.

**Example Matrix**

The following diagram is a matrix for the Use Cases and Domain Model **for a College Management System**



This CRUD matrix has highlighted some worrying omissions in our analysis;

1. Several of the classes have no “Create” operation, meaning that something is missing. As an example, there is no way of creating a new tutor in our system! This suggests a Use Case along the lines of “Register Tutor” is required (and the same applies for Course Event, Course, Location and Room).
2. Also of concern is that many of the classes cannot have their instances deleted. This isn’t a showstopper, but it may not be what is required. Perhaps we don’t care that Rooms cannot be deleted (it may be a very minor requirement), but the Data Protection Act would probably be violated by us neglecting to provide a mechanism to delete a Delegate.

Although many of the omissions are because we have reduced the number of

Use Cases to keep our examples simple,