Objectives

In this practical you will gain experience with the analysis and modelling activities of identifying objects and classes, associations between classes, and decomposing a system into subsystems. You should use UML notation, but that is not the main focus of this practical. Whilst working in groups you will have the opportunity to argue for and against and to evaluate your choice of models from the many different solutions possible.

Task A

This part is based on the reading *No Silver Bullet: Essence and Accidents of Software Engineering*, by F. P. Brooks. Read this article and discuss the following questions:

1. In Brooks’ view, why is building software hard?
2. Explain, in your own words, each of the following concepts: complexity, conformity, changeability, invisibility.
3. What claims does Brooks make in favour of object-oriented programming?
4. Why is OO nonetheless, not a silver bullet?
5. Why are good designers important for software development companies?
6. Can good designers be trained? Give arguments for and against.
7. This article, and its original 1975 version in Brooks’ book *The Mythical Man Month* (FIZ 005.1.1975 MYT), have prompted much debate in the SE community. Search the library or the web for some of the arguments made against Brooks’ claims.
**Task B**

This part is to be undertaken by students in groups of 5 or 6.

1. Study the description of the required behaviour of a University Library system given below.

2. Draw a **UML use case diagram** for the system to reflect the following analysis: the actors in this system are the BookBorrower, JournalBorrower, Browser and Librarian; the required functionality of the system is captured by 8 use cases: BorrowCopyofBook, ReturnCopyofBook, BorrowJournal, ReturnJournal, ExtendLoan, BrowseLibrary, ReserveBook, UpdateCatalog.

3. Divide your group into pairs and use the 3 object identification methods studied in lectures (grammatical analysis, entity-boundary-control, and CRC method) to **identify potential objects and classes** to be modelled in this system. Initially, focus on just the first 4 use cases for borrowing and returning a book or a journal. If you have time you can return to identify the objects for the full system.

4. Each pair should present their identified objects, and the group should select objects for the model from the potential objects suggested. Identify the relationships in this system between classes and label with role and multiplicity as necessary. Remember that class A is *associated* with class B if A *has to know about* B.

5. Use a **UML sequence diagram** to document the activities and interactions between objects which occur in the BorrowCopyofBook use case.

6. Select an object with non-trivial dynamic behaviour (for example the journal class), and describe the behaviour of that object using a **UML statechart diagram**.

**University Library Case Study**

You have been contracted to develop a computer system for a **university library**. After some careful investigation, the following facts emerge about the required system.

**Books and Journals** The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for 3 weeks. Only members of staff may borrow journals. Members of the library can normally borrow up to 6 items at a time but members of staff may borrow up to 12 items at one time. New books and journals arrive regularly, and old ones are sometimes disposed of. The current year’s journals are sent away to be bound into volumes at the end of each year.

**Borrowing** It is essential that the system keeps track of when books and journals are borrowed and returned, since the current system already does that. The new system should produce reminders when a book is overdue. There may in future be a requirement for users to be able to extend the loan of a book if it is not reserved.

**Browsing** The system should allow users to search for a book on a particular topic by a particular author etc., to check whether a copy of the book is available for loan and, if not, to reserve the book. Anybody can browse in the library.

**Case Study Reference:** Stevens & Pooley, *Using UML*, Addison Wesley 2000