Chapter X

Transactions and Their Top-Level Design

This chapter starts with an overview of the design phase according to the functional and object oriented methodology (FOOM) and presents the subphases and their products. Then, the chapter focuses on the design of transactions. First it describes what transactions are and how they can be identified and extracted from the object oriented data flow diagrams (OO-DFD). Afterwards, it explains how to map transaction diagrams to top-level descriptions, which detail their components and process logic. The transactions’ top-level descriptions will eventually become detailed descriptions of respective class methods. The chapter ends with additions to the data dictionary (DD) the class diagram due to the definition of the transactions.

Overview of the Design Phase According to FOOM

In the design phase we use the products of the analysis phase in order to design the components of the system. At the end of the design phase, the following products will be created: (1) a complete class diagram which will include, in addition to all the data classes, the interface, input (Forms), and output (Reports) classes; every class will include, in addition to its attributes, a list of its functions; (2) a detailed description of the methods, the functions of the classes.
methods will be described in pseudo code or with message charts; (3) the user interface—in the form of a menus tree; and (4) the input and output screens/reports. The products of the design phase will enable the creation of the information system (IS) in an object oriented environment.

In order to create these products, the design phase is carried out in several subphases, which are:

1. identification of the transactions of the system and creation of their top-level descriptions;
2. design of the user interface—a menus tree—and addition of the Menus class;
3. design of the input and output screen/reports and addition of the Forms and Reports classes; and
4. detailed design of the transactions and their decomposition into class methods, described in pseudo code or message charts.

This chapter deals with the first subphase, while the other subphases will be dealt with in the following chapters.

### Identifying the Transactions

#### What is a Transaction?

A transaction is an independent computer process which performs a task for a user of the system in order to assist in the completion of a business process. A transaction may include a series of activities performed by a computer program which support in achieving the user’s task. For example, a transaction may enable the user to input new data, update the database, retrieve data from the database, perform various computations (e.g., summarizing, sorting, comparing), produce reports, and so on. The activities included in a transaction are performed in a certain order or process logical, according to the user’s task.

Hence, an IS is made up of transactions, each assisting in performing a certain business task. All the transactions taken together express the functionality of the system. In a business/organizational IS, most of the transactions will be activated by users who interact with the system; that is, a certain user may interact with a certain transaction and together carry out the user’s task. But some of the transactions may not be activated by users but rather “automatically” by the