CHAPTER 2

Dissection of an Application Framework

WE LOOKED BRIEFLY in the previous chapter at what an application framework is, and we considered its benefits and costs. In this chapter, we will dive more deeply into the details of the framework. We will look specifically at what is in a framework, how we can develop a framework for our application, and what object-oriented techniques we can leverage in developing the framework.

To better understand how we can develop an application framework, we need first to understand what goes in an application framework and its relationship to other parts of the system.

Framework Layers

You learned in chapter one that an application framework is a "semifinished" application that can act as a starting point for a business application. Applications that are built on top of the framework consist of two layers: the application layer and the framework layer. The framework layer may consist of numerous components, which can be again grouped into domain-specific components and cross-domain components. Figure 2-1 illustrates the different participants in an application and their relationship to each other.

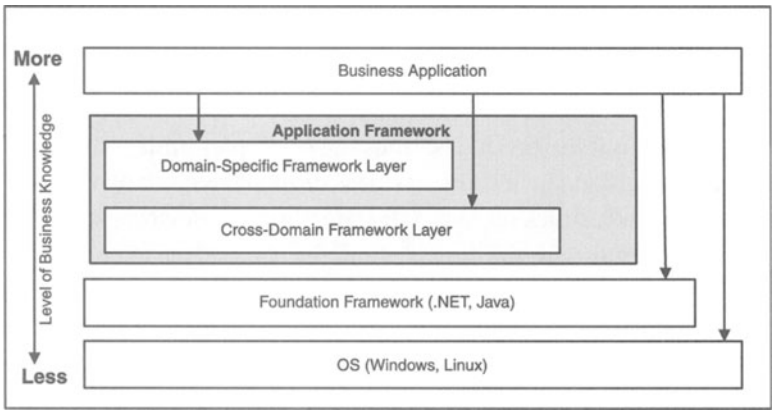

Figure 2-1. Multiple layers within an application

The following is a brief description of what each layer represents and what role it plays in the overall system.

The Business Application

The business application represents the custom application that developers are responsible for. It implements the detailed business knowledge for the specific application under development. Developers build the business application according to the particular scenario described by business analysts. As the business logic and rules change, it is this level at which changes will mostly likely occur, particularly when such changes are minor and isolated.

The Application Framework

The application framework represents the semifinished application that architects have developed as a basis for developers to use to construct their business applications. The application framework can be broken down into two layers: a domain-specific framework layer and a cross-domain framework layer.

The Domain-Specific Framework Layer

The domain-specific framework consists of specialized framework components that target a specific business domain. In comparison to the business application layer, the domain-specific framework layer implements knowledge that is common to all applications of a particular business domain, in contrast to the business application layer, where the business knowledge and logic are targeted to a particular application.

You can think of a domain-specific framework as corresponding to a country's constitution and a business application as analogous to the laws of a particular state or local government. The constitution doesn't describe the specific laws that the state has to implement, but instead it describes the principles under which the system of laws should be framed. Each state may pass its own laws, but all those laws must be based on the principles set out in the constitution. However, as long as the state law is in conformity with the constitution, the state is free to create laws that are best suited to that state. Like the constitution, a domain-specific framework doesn't mandate how each business application should be built; instead, it provides a set of components that encapsulate the core business characteristics and processes of a particular business domain. For example, a shopping cart component describing a customer's selected product items, the quantity