Overview of Frameworks

Objectives

Upon completion of this module, you should be able to:

- Define a framework
- Describe the advantages and disadvantages of using frameworks
- Identify several common frameworks
- Understand the concept of creating your own business domain frameworks

Additional Resources

Additional resources – The following references provide additional information on the topics described in this module:

- Fowler, Martin. *Analysis Patterns*. Addison Wesley Longman, Inc., 1997.
- Larman, Craig. *Applying UML and Patterns (3rd ed)*. Upper Saddle River: Prentice Hall, 2005.

Description of Frameworks

A software framework is a re-usable software infrastructure that can be extended and configured to provide a specific software solution. The infrastructure can include components, APIs, scripts, support applications, configuration files.

A software framework provides extension points in the framework where the application programmers may make additions or modifications for specific functional requirements.

A framework can provide the infrastructure for:

- One or more tiers such as web presentation, business services, entities, and integration tiers

  These infrastructure files will often facilitate the non-business oriented tasks required by a software system such as transactions, security, or object life-cycle management.
- A specific business domain, for example, insurance, banking or oil exploration

  These infrastructure files are created and packaged but are usually used internally within an organization
- A shared business domain requirement such as resource allocation, event management or billing

  These infrastructure files can be packaged and reused within an organization or sold to third parties

Customization of a framework is done by:

- Extending framework classes or implementing framework interfaces. You implement the framework interfaces or extend the framework abstract classes and provide the specific method implementation required by the framework. In this customization process:
  - Your classes may be less coherent due to their mixed responsibilities
  - Your classes may be more difficult to test outside of the framework
- Informing the framework of the Plain Old Java (POJO) classes it must manage by using configuration files or annotations. POJOs do not extend or implement the framework classes or interfaces, but are

configured using configuration files, or annotations added to your source POJO files. POJOs often allow the framework to be replaced with minimal effort. In this customization process:

- POJOs tend to be more coherent
- POJOs can be easier to test outside of the framework environment

Using Existing Frameworks

List of common frameworks:

- Ruby on Rails – A free software application framework for the Ruby Platform using model-view-controller (MVC) architecture, for rapid web development.
- Spring Framework – An open source application framework for the Java platform and.NET Framework, primarily used for the business tiers, but it has extensions for building the web presentation tier. It is often used in conjunction with hibernate and as an alternative to a JEE application server.
- Java Server Faces (JSF) – An open source web application framework for the Java platform, which uses component based and event-driven approach for rapid application development.
- Hibernate – An open source Java-based framework to perform Object/Relational mapping and queries. This framework fits into the Integration tier.
- Struts – A free open source framework for creating web applications, based on JavaServer Pages (JSP). It helps developers achieve separation of concerns of the model, view and controller, by utilizing the MVC architecture.
- Microsoft .NET – A software framework, designed to run on Microsoft Windows operating systems. It provides a framework for developing in multiple tiers including the web, business, integration tiers.
- Struts 2 – An open source framework for web application development, with improved features over Struts. The improvements include POJO forms and action classes.

Advantages and Disadvantages of Using Frameworks

Following is the list of advantages for using frameworks during application development:

- Developers can focus on the new business problem and need not worry about the infrastructure problems, or common business aspects of the overall application.
- Frameworks usually include good OO practices and patterns.

- Once you have gained experience with a framework, code is easier to write and support

Some of the disadvantages for using frameworks are:

- Your code may become bloated due to the one-size-fits-all approach of the framework.
- Frameworks may be difficult to learn.
- You are restricted by the infrastructure code and cannot usually modify the infrastructure files.
- Changing to an alternative framework may be difficult.

Building Frameworks

A generic framework can be built for a specific business domain. Generic attributes and methods could be specified in the framework. The generic classes could have associations with other generic classes and link attribute methods could be specified in the framework classes.

An insurance company could create a generic insurance framework that would support any insurance domain product. For example:

- Pet insurance
- Car insurance
- Life insurance
- Property insurance
- Public liability insurance

The following examples show two alternative approaches to build a generic framework for an insurance domain with a specialization of pet insurance:

- Figure 15-1 shows a framework based on abstract classes
- Figure 15-2 shows a framework based on interfaces and abstract classes

There are other possible approaches to build frameworks.

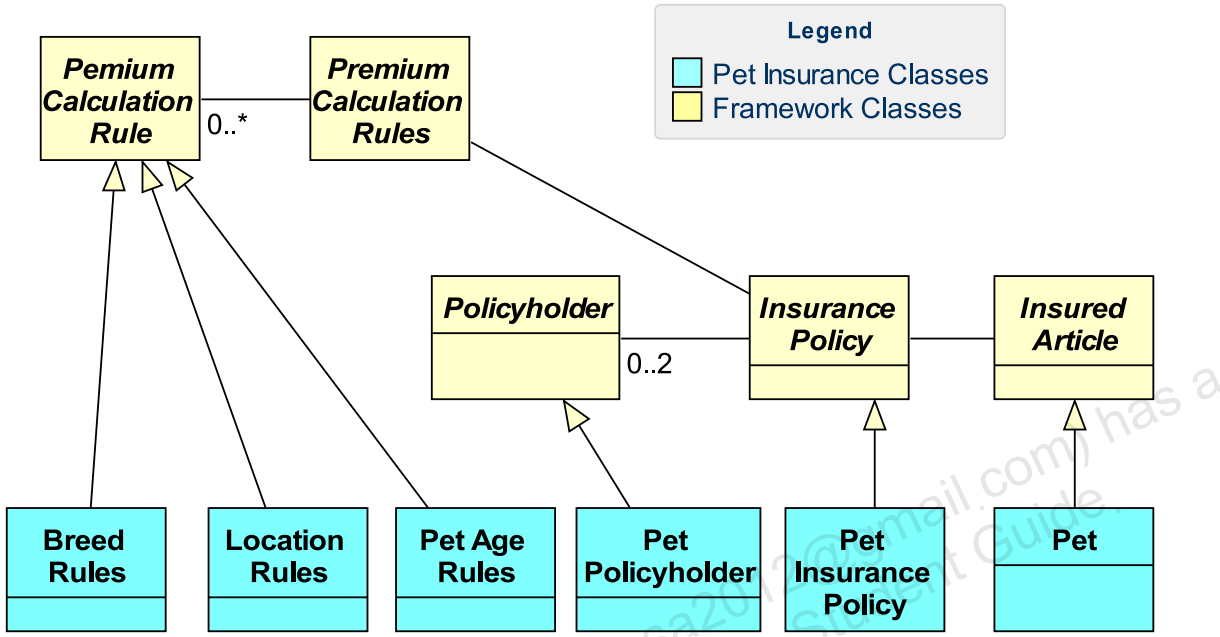

Figure 15-1 Example showing framework of abstract classes

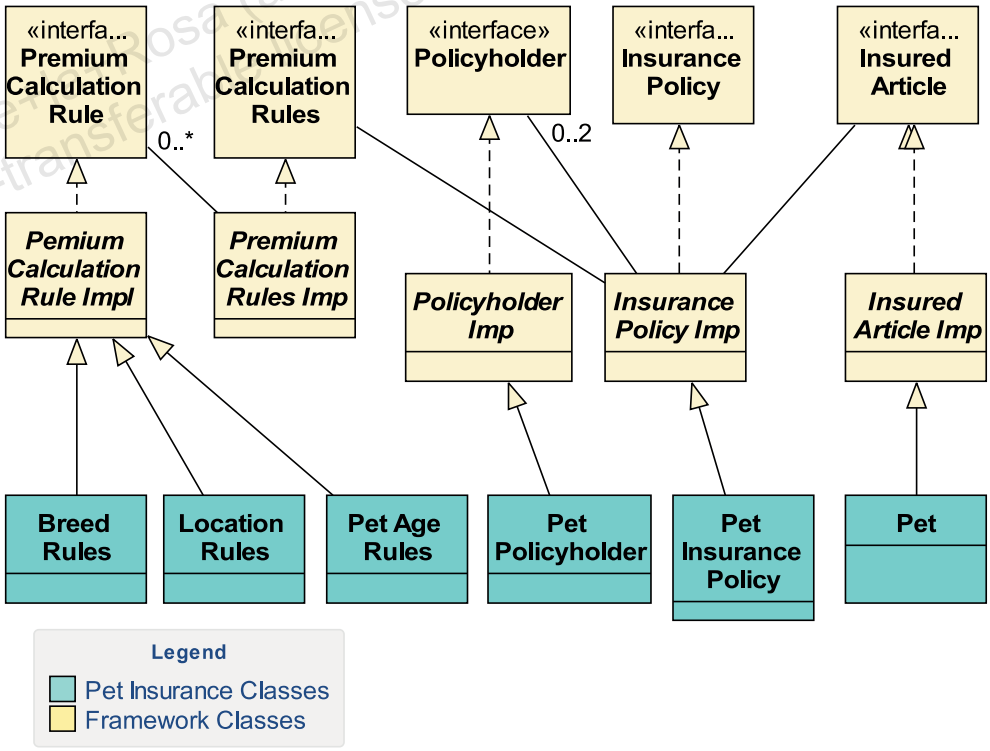

Figure 15-2 Example showing a framework of interfaces and abstract classes

Domain Neutral Frameworks

Domain neutral frameworks are often used for subsystems. These frameworks contain common features required by a variety of different domains. For example:

- A used car sales system could use a trading framework and a billing framework
- A human resource system could use a resource allocation framework

The generic patterns are often called analysis patterns. For example one such pattern is Party (Person or Company), Place, Thing, Event. This forms the basis of event planning or resource scheduling. However these may be far too abstract and generic to be used in an application.

Advantages and Disadvantages of Building Frameworks

The advantages of building a framework include:

- A reduction in cost and development time for each specific domain version using the framework

  Many of the time consuming development aspects, for example, the generic requirements gathering, requirement analysis, design, and testing have been completed and are contained in the framework. Therefore the cost and time to develop the specific version may be significantly reduced.

- May result in a competitive advantage

  If your time to market and cost are reduced, you can achieve a significant advantage over your competitors' offerings.

- Developers can focus on the differences between the specific domain and the framework
- Frameworks usually include good OO practices and patterns

The disadvantages of building a framework include:

- Can be expensive to build

  The cost to build a generic framework will be high because the developer has to consider all the possible use cases in which the framework will be used and provide a flexible framework to accommodate the specific domain versions

- Requires an excellent knowledge of all the domains that the framework will be used for

  You need access to domain experts in all of the specific domains that the framework will be used for.

- The code may become bloated due to the one size fits all approach of a framework
- The framework code and specific version code combined will always be greater than a version written from scratch
- Frameworks may be difficult to learn
- You are restricted by the infrastructure and cannot usually modify the infrastructure files

  If the developers of the framework did not consider all the possible use cases, then the framework may not fit. For example, consider a vehicle domain framework envisaged to be used for vehicles of land, water, or air. If the framework used an attribute of weight this would cause a problem in a case where the framework gets used for Space vehicles, where an attribute of mass would be more appropriate.

Summary

In this module, you were introduced to the basic concepts of:

- Frameworks
- Using existing frameworks
- Creating domain neutral and domain specific frameworks