CSCI 253

Object Oriented Design:
Frameworks and Design
George Blankenship

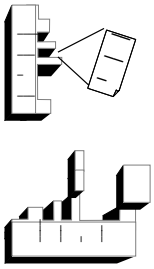
Background

- A class is a mechanisms for encapsulation, it embodies a certain service providing the data and behavior that is useful in the context of some application. A single class is rarely the complete solution to some real problem
- There is a growing body of research in describing the manner in which collections of classes work together in the solution of problems. Application frameworks and design patterns are two ideas that became popular in this context

Application Frameworks

- An application framework is a set of classes that cooperate closely with each other and together embody a reusable design for a general category of problems
- Although one might abstract and discuss the design elements that lie behind the framework, the framework itself is a set of specific classes that are typically implemented only on a specific platform or limited set of platforms
- The framework dictates the overall structure and behavior of the application. It describes how responsibilities are partitioned between various components and how these components interact

A Framework as an Upside-down Library



- In a traditional application the application-specific code defines the overall flow of execution through the program occasionally invoking library-supplied code
- A framework reverts this relation: the flow of control is dictated by the framework and the creator of a new application merely changes some of the methods invoked by the framework
- Inheritance is often used as powerful mechanism for achieving this. Alternatively application-specific components that obey a specified interface are plugged in

Frameworks and Design George Blankenship 4

GUI Application Framework

- GUI application frameworks simplify the creation of graphical user interfaces for software systems
- A GUI application framework implements the behavior expected from a graphical user interface
  - windows, buttons, menu's, text fields
  - move and resize windows
  - handle mouse events on buttons and menus
- A new application is built by specifying and arranging the necessary elements and by redefining certain methods
  - buttons, menu's, text fields
  - responses to the mouse and key events

Frameworks and Design George Blankenship 5

Simulation Application Framework

- Simulation frameworks simplify the creation of simulation style applications
- A simulation framework provides a general-purpose class for managing the types of objects in the simulation
- The heart of a simulation framework is a procedure that cycles through the list of objects introduced in the simulation asking each to update itself
- The framework knows nothing about the particular application it is going to be used for: billiard balls, fish in a tank, rabbits and wolves in an ecological game, etc.

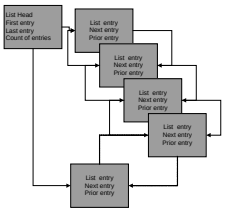
Frameworks and Design George Blankenship 6

GCB Application Framework

- Family
  - Link list
  - GUI
  - Tracing
  - State Machine
  - TCP/IP communications
- Used to build a standard substructure for application

GCB Linked List

- Create linked lists
- Add/remove items
- Traverse a list
- Define an object as a list entry
- Add arbitrary object to a list

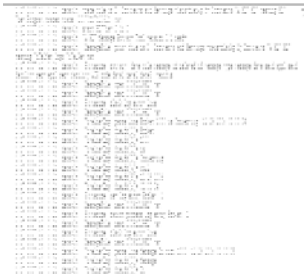


GCB GUI

- Build/maintain main GUI
  - Menu bar
  - Scrolling text area
  - Application specific widgets
  - Input area
  - Time display
- Build/manage Solicitation GUI
- Build application parameter panel
- Build application text validation widget

GCB Trace

- Creation an display of program execution trace entries
- Tagged with time and class that generated entry
- General entry
- Exception entry
- Display to GUI, file, command box
- Controlled by component
- <date time><component><text>



GCB State Machine

- FSM creation and completion
- Transition table
  - Event driven processing
  - Missing event detection
- FSM execution
  - Thread safe execution
  - Get current state
  - Execute state transition
  - Trace FSM execution

GCB TCP/IP Communications

- Create TCP port
  - Passive port (listen)
  - Active port (connection)
- Send message
- Receive message
- Trace program execution

Design Patterns

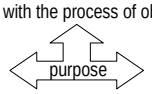
- Design patterns are an attempt to collect and catalog the smallest recurring architectures in object oriented software systems. A design patterns typically captures the solution to a problem that has been observed in many different systems
- Design patterns are more abstract than a framework. Frameworks are partial implementations of (sub)systems, design patterns have no immediate implementation at all
- Design patterns are smaller architectural elements than frameworks, most applications (and frameworks) will use several patterns
- Design patterns are architectural level counterparts of programming idioms

Frameworks and Design George Blankenship 13

Sorts of Design Patterns

Creational Patterns:
are concerned with the process of object creation

Structural Patterns:
are concerned with how classes and objects are composed to form larger structures



Behavioural Patterns:
are concerned with algorithms and the assignment of responsi-bilities between objects

Class Patterns deal with static relationships between classes and subclasses

Object Patterns deal with object relationships which can be changed at run time

Frameworks and Design George Blankenship 14

Creational Design Patterns

- Abstracts the instantiation process:
  - Encapsulates knowledge about which concrete classes to use
  - Hides how the instances of these classes are created and put together
- Gives a lot of flexibility in what gets created, who creates it, how it gets created, and when it it gets created
- A class creational pattern uses inheritance to vary the class that is instantiated
- An object creational pattern delegates instantiation to another object

Frameworks and Design George Blankenship 15

Structural Design Patterns

- Structural design patterns are concerned with how classes and objects are composed to form larger structures
- A class structural pattern uses inheritance to compose interfaces or implementation; compositions are fixed at design time
- An object structural pattern describes ways to compose objects to realise new functionality; the added flexibility of object composition comes from the ability to change the composition at run-time

Behavioral Design Patterns

- Behavioral design patterns are concerned with algorithms and the assignment of responsibilities between objects
- Behavioral class patterns use inheritance to distribute behavior between classes
- Behavioral object patterns use object composition rather than inheritance; some describe how a group of peer objects cooperate to perform a tasks no single object can carry out by itself; others are concerned with encapsulating behavior in an object and delegating request to it

Overview

Creational Patterns
- Singleton
- Abstract factory
- Factory Method
- Prototype
- Builder

Structural Patterns
- Composite
- Façade
- Proxy
- Flyweight
- Adapter
- Bridge
- Decorator

Behavioral Patterns
- Chain of Respons.
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

The Elements of a Design Pattern

- A pattern name
- The problem that the pattern solves
  - Including conditions for the pattern to be applicable
- The solution to the problem brought by the pattern
  - The elements (classes-objects) involved, their roles, responsibilities, relationships and collaborations
  - Not a particular concrete design or implementation
- The consequences of applying the pattern
  - Time and space trade off
  - Language and implementation issues
  - Effects on flexibility, extensibility, portability