Architecture & Design of Embedded Real-Time Systems (TI-AREM)

Design Pattern Introduction
GoF Design Patterns

GoF (Gang of four) Design Pattern Book:
Design Patterns
Elements of Reusable Object Oriented Software,
By Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides,

Three main parts:

1. Introduction
2. A Case study:
   Designing a Document Editor
3. Design Pattern Catalog:
   with 23 Design Patterns

Also available as
a CD-ROM version
What is a Design Pattern?

A design pattern has four essential elements:

1. The **pattern name**
   - a design vocabulary

2. The **problem**
   - and its context i.e. when to apply the pattern

3. The **solution**
   - the design pattern elements (a solution template)

4. The **consequences**
   - the results and trade-offs (space and time trade-offs)
GoF – Design Pattern Template

- Pattern Name and Classification
- Intent
- Also Known As
- Motivation
- Applicability
- Structure (class diagram)
- Participants
- Collaborations (sequence diagram)
- Consequences (+/-)
- Implementation
- Sample Code (C++ code examples)
- Known Uses
- Related Patterns
GoF Design Pattern Categorization

• Creational Patterns (5 patterns)
  – abstracts the instantiation process

• Structural Patterns (7 patterns)
  – are concerned with how classes and objects are composed to form larger structures

• Behavioral Patterns (11 patterns)
  – are concerned with algorithms and the assignment of responsibilities between objects
GoF Design Pattern – Overview (1)

Creational Patterns:

- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton
GoF Design Pattern – Overview (2)

Structural Patterns:

• Adapter
• Bridge
• Composite
• Decorator
• Façade
• Flyweight
• Proxy
GoF Design Pattern – Overview (3)

Behavioral Patterns:

• Chain of Responsibility
• Command
• Interpreter
• Iterator
• Mediator
• Memento
• Observer
• State
• Strategy
• Template Method
• Visitor
GoF Design Pattern Relationships

Typical relationships from the related patterns sections
Real-Time Design Patterns

Architecture for Real-Time Systems
BPD 1.6: What is a Design Pattern?

"A design pattern is a generalized solution to a commonly occurring problem"

• Design is all about optimization
• The hard part is that there are so many things to optimize and so little time
Design Optimization Criterions

- Performance (worst case, average case)
- Predictability
- Schedulability
- Minimize resource requirements (memory, heat, weight)
- Reusability
- Portability
- Maintainability
- Readability
- Development time / effort
- Safety
- Reliability
- Security

We must rank them in order of importance to the success of the project and the product
Another type of Pattern Categorization

• **Architectural patterns**
  – affect most or all of the system
  – are broadly and strategically applied to the system
  – covered by **BPDs “Real-time design patterns” book**
    • covers patterns particular relevant to real-time and embedded systems

• **Mechanistic design patterns**
  – GoF pattern with a more local scope
  – they define mechanisms for object collaborations
  – have a much more limited scope, but are general applicable
  – covered by **“Design Patterns” book, Gamma et. al. (GoF)**
    • covers patterns relevant to all kind of (OO) systems

• **Idioms**
  – language dependent design Patterns
Pattern Hatching – Locating the Right Patterns

Figure 3-10: Pattern Hatching
Pattern Mining – Rolling Your Own Patterns

Generalization

Figure 3-11: Pattern Mining
Pattern Instantiation – Applying Patterns in Your Design

Figure 3-12: Pattern Instantiation
Domain Model for an Embedded System

- GoF patterns
- BPD patterns
- Statechart
- Sequence Diagram
- Class Diagram
- Collaboration Pattern
- Production System
- Prototyp
- Use Case
- Requirement
- Event
- Interface
- User
- Agent
- Devices
- Class
- Structure
- Behaviour
- 1 Software
- 1 System
- 1 Hardware
- Processor
- Object
- Concurrency Pattern
Summary

• Different types of pattern categorization
• Architectural patterns
  – e.g. Real-Time Design Patterns
• Mechanistic Design Patterns
  – e.g. GoF Patterns with categorization
    • Creational
    • Structural
    • Behavioral
• Idioms (code dependent patterns)