

Advanced Design and System Patterns (code ADP)



A quick Review of the GOF Patterns

- Pattern categories
- Most important patterns in real-life applications
- Precise documentation of patterns using UML 2.0
- Extensions and adaptations of GOF patterns
- What are the most important patterns?

Pattern Languages using GOF patterns

- Combining inheritance and composition
- Discovering and inventing new pattern combinations
- Combining Creational, Structural and Behavioural patterns
- Reusing pattern languages

Generic Patterns: Fundamentals

- What is genericity, exactly?
- Generic programming versus object-oriented programming
- Subtype and parametric polymorphism
- Creating generic versions of GOF patterns

Generic Patterns: Design and Applications

- Services and interfaces (provides and requires)
- Traits and traits classes
- Policies and their relationship with contracts
- Policy-based design
- Enriched policies

The POSA Patterns: Overview

- What is POSA?
- Architectural patterns in POSA
- Design patterns in POSA
- Relationship with GOF patterns

Structural Decomposition

- Whole-Part pattern
- Examples from application areas
- Comparing Whole-Part with Composite
- Combinations

Management

- Command
- Command Processor
- Generic Command
- Combining Command and Visitor
- View Handler pattern

Communication

- Mediator and its role in large systems
- The 6 Proxy types
- The Propagator pattern
- Forward-Receiver pattern
- Client-Dispatcher-Server pattern
- Why Observer is not always the best solution

Organisation of Work

- The Builder pattern
- Applying Builder to create arbitrary object networks
- Master-Slave pattern
- Master-Slave: threads and parallel versions
- Facet (Role) pattern

Interactive Systems

- Model-View-Controller (MVC)
- Presentation-Abstraction-Control (PAC)
- PAC versus MVC; which one (or both)?
- Creating large applications with PAC

Parallel Decomposition Techniques

- Task and data decomposition
- Grouping and ordering tasks
- Data sharing among tasks

Algorithm Structure

- Task and data parallelism
- Divide and conquer
- Geometric decomposition
- Other decomposition techniques

Multithreading

- An introduction to multithreading
- Threads and thread lifecycle
- Mutexes, lock and condition variables

Designing Threaded Applications

- Thread models
- Boss-Worker
- Crew model
- Assembly line

High-level Structures

- Pipes and Filters
- Broker
- Blackboard
- The steps in designing a Blackboard application

An Introduction to Domain Architectures

- What is a domain architecture?
- The five DA categories
- Using domain architectures to structure POS and GOF patterns
- Embedding POSA and GOF patterns in a DA setting
- Examples and test cases

Your Trainer

Daniel J. Duffy started the company Datasim in 1987 to promote C++ as a new object-oriented language for developing applications in the roles of developer, architect and requirements analyst to help clients design and analyse software systems for Computer Aided Design (CAD), process control and hardware-software systems, logistics, holography (optical technology) and computational finance. He used a combination of top-down functional decomposition and bottom-up object-oriented programming techniques to create stable and extendible applications (for a discussion, see Duffy 2004 where we have grouped applications into domain categories). Previous to Datasim he worked on engineering applications in oil and gas and semiconductor industries using a range of numerical methods (for example, the finite element method (FEM)) on mainframe and mini-computers. Daniel Duffy has BA (Mod), MSc and PhD degrees in pure and applied mathematics and has been active in promoting partial differential equation (PDE) and finite difference methods (FDM) for applications in computational finance. He was responsible for the introduction of the Fractional Step (Soviet Splitting) method and the Alternating Direction Explicit (ADE) method in computational finance. He is also the originator of the exponential fitting method for time-dependent partial differential equations.

He is also the originator of two very popular C++ online courses (both C++98 and C++11/14) on www.quantnet.com in cooperation with Quantnet LLC and Baruch College (CUNY), NYC. He also trains developers and designers around the world. He can be contacted dduffy@datasim.nl for queries, information and course venues, in-company course and course dates