POAD Book: Chapter 7 POAD: The Process

Instructor: Dr. Hany H. Ammar Dept. of Computer Science and Electrical Engineering, WVU

#### Goals

- Pattern Integration: Present two different ways for gluing patterns
- To introduce the process aspects of POAD
- to outline the POAD approach and to illustrate how patterns can be utilized as design building blocks
- We will later see how POAD evolve from their integration

- Pattern Integration: Stringing vs Overlapping
- POAD Process Outline (Nutshell)
  - Analysis
  - Design
  - Design Refinement

# Pattern Integration: Stringing vs Overlapping

- Both techniques (stringing patterns, and overlapping them together) were originally used for Civil engineering purposes, but they do apply to building software systems.
- Stringing -patterns are glued together
  - glue could be UML relationships
  - design is a loose assembly of patterns
  - You end up with too many classes with trivial responsibilities

# Pattern Integration: Stringing vs Overlapping

- Overlapping
  - A class as a participant in one pattern could be
    at the same time a participant of another pattern
    in the same application design
  - One class can play two different roles, in two different patterns
- The Advantage of overlapping is that you have fewer classes
- The disadvantage is that pattern boundary is lost and patterns are hard to trace

## Example

#### Consider the Reactor and Composite Patterns



## Example



## Example

Now, you have two patterns to glue, the *Reactor* and the *Composite* patterns

 Stringing the 2 patterns together gives us the following

## Example: Stringing



## Example: Overlapping



#### What does POAD uses

- Although Stringing is the easier of the 2 approaches, it is often avoided
  - It provides good traceability
- POAD uses both
  - It uses the simplicity and traceability of the stringing-patterns approach
  - the density and profoundness of the overlapping-patterns approach
- POAD integrates both methods in one process

# Hierarchical Integration Techniques

- POAD starts by assembling patterns at a higher level of abstraction using the stringing approach
- It then allows the designer to integrate the lower level classes to produce dense and profound designs

- Pattern Integration: Stringing vs Overlapping
- POAD Process Outline (Nutshell)
  - Analysis
  - Design
  - Design Refinement

### POAD Process Outline (Nutshell)

- We will use the purpose/process/product template to explain the various steps within a development phases
  - Purpose explains why a designer would conduct this step
  - Process describes the activity that the designer conducts in this step
  - Product describes expected output of this step

#### Three phases of POAD

#### analysis

A set of patterns are selected from a domain specific library

#### high-level design phase where

- patterns are glued together using pattern
  composition models to produce an initial class
  diagram
- design refinement phase
  - initial class diagram is processed to produce a more dense and profound class diagram





- Pattern Integration: Stringing vs Overlapping
- POAD Process Outline (Nutshell)
  - Analysis
  - Design
  - Design Refinement

#### Analysis phase Purpose



- Analyze the application requirements and decide what patterns that will be used

#### Process

- UML use case diagrams and sequence diagrams are used to identify required patterns
- Main concern is determining whether or not the pattern can be used and why it is better
- analyst searches pattern catalogues for candidate patterns
- Analyst must be *acquainted* with the catalogue

### Analysis phase



- Retrieval how to select a pattern from a catalogue
- Product

- Patterns chosen by the application analyst
- For example: Recall the
- Feedback control example. Frror Observer and Strategy pattern Controlled Actuatin Feed Referenc Output Signat is selected for the forward e Input Elements feedforward component, Plant Feedbac Feedback k Data And for the *feedback* component Measureme Elements nt

- Pattern Integration: Stringing vs Overlapping
- POAD Process Outline (Nutshell)
  - Analysis
  - Design
  - Design Refinement



 Develop the application design by composing the patterns selected in the analysis phase

#### Process

- Instantiating patterns, and identifying relationships between instances
- Proceed from the *Pattern-Level* diagram to create a *Pattern-Level with Interface* diagram

# Design phase

#### Process (Cont.)



From Pattern-Level with Interface diagram, the designer identifies details of the pattern and Detailed Pattern-Level diagram is produced

#### Product

- The product of this phase is *Detailed Pattern-Level* diagrams



- Pattern Integration: Stringing vs Overlapping
- POAD Process Outline (Nutshell)
  - Analysis
  - Design
  - Design Refinement

#### Design Refinement phase

#### Purpose

- To develop the profound dense class diagram for the application
- Process
  - Starting with *Detailed Pattern-Level* diagram
  - Designer instantiates each pattern in the context of the application

## Design Refinement phase

Process (cont.)

- This produces our initial class diagram
- class diagram is obtained from gluing patterns
  together at the high level design

#### Product

– Optimized class diagram for the application

