

Lecture 22: Two Examples of OO Design Methods

Object-Oriented Analysis and Design
CSCI 6448 - Spring 2001
Kenneth M. Anderson

Goals for this Lecture

- Cover CRC Cards
- Cover Peter Coad's method of using color in OO Analysis and Design

April 5, 2001

© Kenneth M. Anderson, 2001

2

Source Materials

- CRC Cards
 - UML Distilled by Martin Fowler, Kendall Scott
 - OO Methods by Ian Graham
 - <http://c2.com/doc/oopsla89/paper.html>
 - by Kent Beck and Ward Cunningham
- Color in OO A&D
 - Java Modeling in Color with UML
 - by Peter Coad, Eric Lefebvre, Jeff De Luca

April 5, 2001

© Kenneth M. Anderson, 2001

3

CRC Cards

- CRC Cards stands for
 - Class-Responsibility-Collaboration Cards
- Meant primarily as a brainstorming tool for analysis and design
 - Rather than use diagrams, use index cards
 - Rather than record attributes and methods, record responsibilities

April 5, 2001

© Kenneth M. Anderson, 2001

4

Why index cards?

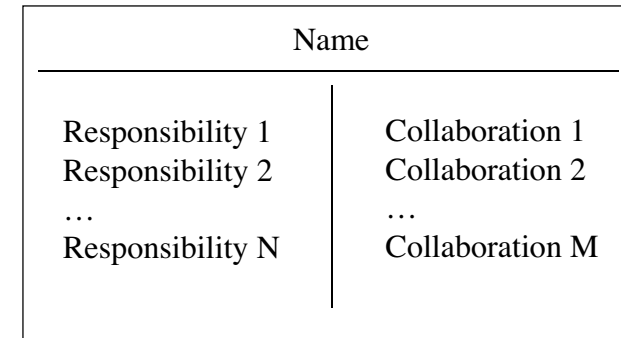
- Forces you to be concise and clear
 - and focus on major responsibilities
 - since you must fit everything onto one index card
- Inherent Advantages
 - cheap, portable, readily available, and familiar
- Affords Spatial Semantics...
 - Close collaborators can be overlapped
 - Vertical dimension can be assigned meanings
 - Abstract classes and specializations can form piles
- ...which provides benefits
 - Beck and Cunningham report that they have seen designers talk about a new card by pointing at where it will be placed

April 5, 2001

© Kenneth M. Anderson, 2001

5

Example CRC Card



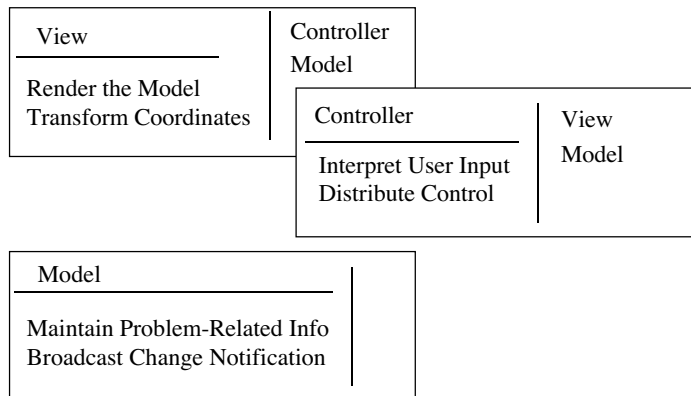
Note: Collaborations are indicated by listing the names of other classes; Responsibilities are typically denoted as short English sentences

April 5, 2001

© Kenneth M. Anderson, 2001

6

Example



April 5, 2001

© Kenneth M. Anderson, 2001

7

Additional Details

- CRC Cards are a technique related to Responsibility-Driven Design
- Some elements of the approach
 - Start with written reqs. spec.
 - Use textual analysis to identify key objects
 - Derive classes and place each class on a card
 - Annotate with super- and sub-class information if desired
 - Continue textual analysis to identify responsibilities
 - Begin to analyze existing cards for collaboration relationships, such as “has-knowledge-of” or “depends-on”
 - Refine until set is stable; continue with detailed design

April 5, 2001

© Kenneth M. Anderson, 2001

8

When to Use

- Problem Domain Analysis
- Use Case Implementation
 - e.g. what classes are needed to carry out the main success scenario (or extensions) of a use case
- Subsystem Design
 - e.g. what objects compose this module?
- Removing “designer’s block”
 - if your team is getting bogged down in details, use CRC Cards to get them thinking about high level responsibilities

Using Color in OO A&D

- Peter Coad (and co-authors) advocates a method of design based on
 - archetypes
 - domain-neutral framework
 - domain-specific components
 - and color!

Why Color?

- Black and White Diagrams, while easier to create, fail to take advantage of the visual capabilities of humans
 - Color can be used to convey “at-a-glance” information
 - It can help to provide a “global overview” of a diagram; a user can then examine the diagram to “fill in the details”

Archetypes

- Basic Idea
 - A form or template for one of a small number of class categories. It specifies attributes, links, methods, plug-in points, and interactions that are typical for classes in that category
- Archetype
 - A form from which all things of the same kind *more or less* follow
- Contrast with Stereotype
 - An unvarying model, as though cast from a mold
 - A broad categorization of classes
 - A text tag for annotating a UML diagram element

Key Concept: “More or Less”

- An archetype specifies a generic template; instances of that template can follow it completely or follow it “in spirit”
- In contrast, stereotypes must be rigidly followed, which means we need to get the stereotype “right” from the start
- Additional contrast: inheritance and interfaces are concepts that must be “followed” not “more or less followed”; The latter implies more flexibility

Labeling Archetypes

- We can denote an archetype by using a UML stereotype
 - however this plain label leaves a lot to be desired
 - it is difficult, for instance, for a stereotype to
 - grab your attention to work on that part first
 - help you discover a progression of archetypes
 - guide you in linking other classes to the archetype
 - that’s why Coad recommends adding color to UML diagrams; to address these issues

Main Archetypes

- The moment-interval
- The Role
- The “catalog-entry description”
- The “party, place, or thing”

- These archetypes are combined by Coad into a “domain neutral” framework

Moment-Interval Archetype

- Represents a moment in time or an interval of time
- Examples
 - A sale is made at a moment in time
 - A rental occurs over an interval of time
- Why create such an archetype?
 - Experience has shown that OO Designs always include moment-interval objects; this archetype reminds us to look for such objects when performing domain analysis

Role Archetype

- A role is a means for participation by a person, place, or thing
- Coad prefers “party” to “person”
 - since a “party” can mean “person” or “organization”
 - both of which can participate in similar roles, such as ownership

Description Archetype

- A collection of values that apply again and again
 - it also provides across the collection of all things that correspond to its description
- Example
 - A truck has a serial number, purchase date, color, and odometer; so do other vehicles; so a “vehicle description” is an example of a “catalog description” archetype

Party, Place, or Thing Archetype

- Used to represent objects that can participate in Roles
- We, therefore, model both the “role” and the “role player”

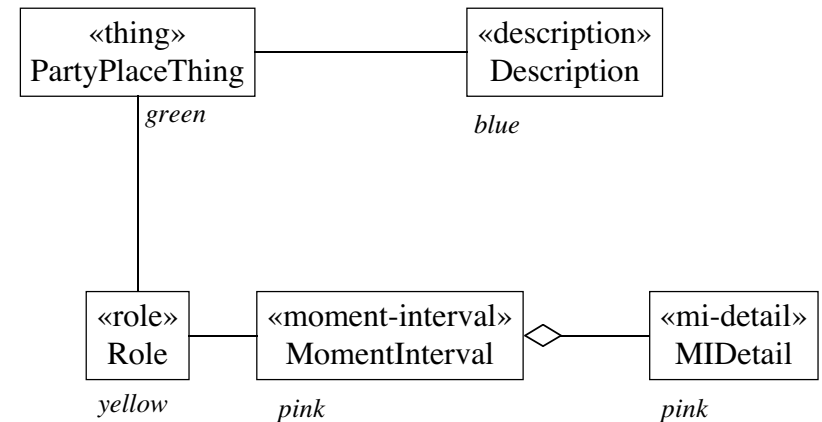
Which Colors?

- Moment Intervals - Pink
- Roles - Yellow
- Descriptions - Blue
- Party, Place, or Thing - Green

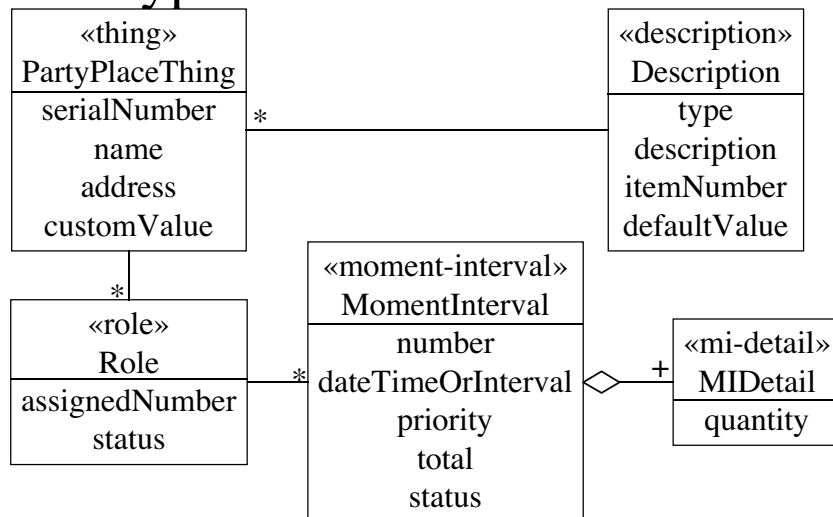
Domain-Neutral Framework

- Coad has developed a domain-neutral framework that captures how a design is to use the defined archetypes
- Lets develop the framework for two steps and then we'll take a look at the end result in Coad's book (page 14)

Core Archetypes and Colors



Archetypes: Attributes and Links



Discussion

- The archetypes are more than just names and colors, they provide suggested attributes, links, methods, interfaces, etc.
- See pages 9-14 of Coad's book
- The rest of Coad's book is a set of domain-specific components, presented in color, using the presented archetypes
 - such as Product-Sale Management
 - or Project-Activity Management