Methods, Models, Madness

- Long history of object-oriented methods and models
  - defacto standard modeling exists today: UML
  - Unified Modeling Language (from Rational via others)
    - “three amigos”: Booch, Rumbaugh, Jacobsen
    - emerging standard via OMG: object modeling group
    - notational tool, as distinct from method/process
  - Methods: Booch, Objectory, Fusion, [DOOM!]

- Learning OO
  - programming, design, analysis (OOP, OOD, OOA)
  - from UML Distilled “now that the methods war is over, I think that patterns will be where most of the interesting material about analysis and design will appear.”

http://www.enteract.com/~bradapp/links/oo-links.html
Basics of modeling

- **Modeling**: coherent approach to designing a system
- **Different techniques make up a system**
  - class diagrams: static notation
  - CRC cards: classes, responsibilities, collaboration
  - interaction, sequence diagrams
  - state transition diagrams
  - use cases
- **Syntax and semantics of models**
  - syntax is notational, can be formal or informal
  - semantics depends on rigor, can be formally defined
  - formal should mean “based on logic/mathematics”
CRC card (see SCOOTER assignment)

- **3x5 index card**
  - developed after brainstorming about classes
  - can be moved around, tacked to walls (portable and useful)
  - facilitates group interaction/discussion
  - both in analysis and design

- **Class**
  - name, sub and super

- **Responsibility**
  - scenarios, walkthroughs

- **Collaboration**
  - other classes: Responsibilities
A simple process (based on Booch)

- **Time frames for one-year project:**
  - conceptualization: 1 month
  - analysis: 1-2 months
  - design: 1 month, 2 max
  - evolution: 9 months
- **Conceptualization:**
  - vision, priorities, prototypes
- **Analysis**
  - what, not how; all requirements
- **Design**
  - logical/physical architecture
  - plan releases
- **Evolution**
  - sequence of releases, iterative

Conceptualization, core requirements.

Analysis: Model desired behavior

Design: Create an architecture

Evolution/Implementation

Maintenace: post delivery
Micro (compare previous Macro) process [Booch]

- Used during design phase in specifying architecture
- Classes and objects, can include CRC cards for example
- Semantics include use-cases, class invariants, documentation
- Relationships include “is-a”, “has-a”, “uses-a”, class diagrams updated
- Types, signatures, algorithms for interfaces, implementations
Programming and Development

- **In the real world you have to deal with people**
  - people are not programmers / programmers are not people
  - programs are developed for users
  - abstract factories are great, when I click this button I want something to happen

- **Development**
  - lessons from Scoobi, what’s important in developing a team-constructed software product
  - what can we do with Scooter, Harpoon?
  - what has been good, what has been bad
  - future for OO programming, Java, C++, ..., CPS 108