Models and The Unified Modeling Language (UML)

What is a Model: Maps

Models are *abstractions* of the real situation to give a clearer view of some aspect.

Maps are models of some region terrain and geography.

Suppress details (buildings, etc.) in interest of clarity.



What is a Model: Maps

Models are *abstractions* of the real situation to give a clearer view of some aspect.

Maps are models of some region terrain and geography.

Suppress details (buildings, etc.) in interest of clarity.

BUT

I have a map of the United States... Actual size. It says, 'Scale: 1 mile = 1 mile.' I spent last summer folding it. I hardly ever unroll it. People ask me where I live, and I say, 'E6'.

Steven Wright



Other Models

- · Airplane models
- Blueprints

•

- · Wiring diagrams
- Building maps
- Weather maps
 - Key point: Abstraction from details to get a better overall view.





Software Models

- JavaDoc documents
- · UI Mockups
- . Entity-Relationship Diagrams
- X UML (Uni.. Modeling Language)
- . → UML (Unified Modeling Language)

Quick UML History

- OO Technology in general "took off" in the late 80s and early 90s.
- Diagrams for previous technologies inadequate.
- Competing OO notations arose to address this problem.
- Mid-90s: UML integrated three of the most prominent notations:
- Grady Booch

٠

- James Rumbaugh
- Ivar Jacobsen

UML Diagram Types

Structural: How do pieces fit together?

- Class (static components)
- Object diagrams (run-time examples)
- Component (packaging)
- Deployment (distribution)

Behavioral / dynamic: How to pieces interact?

- Use cases (interaction with external entities)
- Sequence (internal object interactions over time)
- Collaboration (isomorphic to sequence)
- Statecharts (finite state machines)
- Activity (concurrency)
- Timing (primarily for real-time)

UML Diagram Types

Structural: How do pieces fit together?

- Class (static components)
- Object diagrams (run-time examples)
- Component (packaging)
- Deployment (distribution)

Behavioral / dynamic: How to pieces interact?

- Use cases (interaction with external entities)
- Sequence (internal object interactions over time)
- Collaboration (isomorphic to sequence)
- Statecharts (finite state machines)
- Activity (concurrency)
- Timing (primarily for real-time)

UML Diagram Types

Class Diagrams (isa Static model)

- Classes & interfaces optionally with state & methods.
- Generalization (inheritance): extends / implements.
- Interobject associations.

Object Diagrams - run-time examples conforming to the class diagram.

Sequence Diagrams (isa Dynamic model)

- Objects & lifelines
- Calls and returns

ON TO <u>LUCIDCHART</u>!