Requirements Modeling

15-413: Introduction to Software Engineering
Jonathan Aldrich

The Value of Modeling

• Structure
  • Identify missing information

• Precision
  • Conflict identification
  • Documentation for implementers
  • Often achieved through formal notation

• Form
  • Often graphical
  • Aids in communicating relationships

Why Model?

Student answers
• Catch mistakes early
• Flesh out general design to details
• Find problems or inconsistencies
• Help with estimations
• Experience with problem or design
• Allows you to test against hardware that doesn’t exist

Modeling Goals

• Modeling should be targeted

• When to use models?
  • Aid in communication
  • Increase precision
  • Manage uncertainty

• What models to use?
  • Use models that are easy to understand
  • Use models with semantics
  • Use a model that captures something you don’t understand very well

Analytic and Analogic Models

• This is why object-oriented designers usually do not spend their time in academic discussions of methods to find the objects: in the physical or abstract reality being modeled, the objects are just there for the picking!
  • Bertrand Meyer

Analytic and Analogic Models

• In summer lots of birds will start to sing around sunrise... Does the sun send a message to all the birds individually? If so, in what order? ... These are silly questions, because they are questions about software execution, not the sunrise.
  • Steve Cook and John Daniels
Analytic and Analogic Models

- **Analytic (descriptive) model**
  - A (possibly formal) description of how a system works
  - Economic models with differential equations
  - Finite state machine model showing how software reacts to stimulus
  - Limitations: may not capture all behavior of the target system accurately

- **Analogic (representative) model**
  - A (possibly formal) representation of a system
  - Maps of a battlefield in a war room, with toy planes and tanks positioned
  - Records or Objects representing customers in a corporate database
  - Limitations: the world has properties not captured in the model, and vice versa

  Take home point: Models can be useful, but they are not the same as the thing they describe or represent

Kinds of Requirements Models

- **Goal models**
  - Breaking down complex requirements into simpler ones
  - Understanding the relationship between the machine and parts of the world

- **Scenarios**
  - Use cases
  - Sequence diagrams

- **Information models**
  - Class diagrams
    - Note: although designed to capture OO classes in your program, they may be used to capture more general information domains

Motivation for Goal Modeling

- **Limitations of Scenarios**
  - Inherently partial
  - What should the system do in scenarios not explicitly enumerated?
  - Combinatorial explosion of scenarios
  - Can’t list them all
  - Forces premature commitment to machine/world boundary
  - Scenario picks some boundary
  - May not be the right one
  - Leave required properties implicit
  - Says what happens in this case, but leaves open the question in general

Goal Modeling

- **Goal**
  - An objective the system should achieve through the cooperation of the software and its environment
  - A problem in the world that may not be entirely under software control

- **Requirement**
  - Relations between objects in the environment that are monitored and controlled by the software
  - A problem in the world that is under software control

- **Specification**
  - Relations between input and output of the software
  - The interface of the world and machine

  The purpose of goal modeling is to refine abstract goals into concrete requirements, and design a specification that, together with the properties of the world, will fulfill the requirement

Relationships Among Terms

- **R \&\& As \&\& D = G**
  - The goal G is achieved as a consequence of the requirements R, the assumptions As about actors in the environment, and the properties of the domain D

- **S \&\& Ac \&\& D = R**
  - The requirements R are achieved as a consequence of the specification S, the accuracy of the machine’s knowledge about its environment, and the properties of the domain D

Goal Modeling

- **Goal**
  - Say what should be true of domains in the world
  - Relates two domains: an observed domain and a controlled domain

- **Assumptions**
  - Like a goal, says what should be true of domains in the world
  - Carried out by some actor that is NOT the machine

- **Domains**
  - Machine domains (the machine)
  - Designed domains (interfaces, data formats)
  - Given domains (the world)

- **Goal Refinement**
  - AND-refinement: satisfying all subgoals will satisfy goal
Goal Modeling: Simple Example

- Maintain room temperature according to user preference
- Determine target temperature from user
- Measure current room temperature
- Adjust radiator value
- We use central heating

Legend:
- Client
- Room
- Radiator
- Gate

Extended Example: BART

Extended Example: BART

Building a Domain Model

- Goal Refinement
  - AND-refinement: satisfying all subgoals will satisfy goal
  - OR-refinement: satisfying one subgoal will satisfy goal
- Conflict link
  - Satisfaction of one goal may preclude satisfying the other
- Responsibility link
  - States that an agent can commit to act in such a way that it ensures the satisfaction of the goal

Building a Domain Model

- Goal Maintain[TrainSegment(speedLimit)]
  - InformalDef: A train should stay below the maximum speed it can handle.

FormalDef: ∀ t, s: Train, s: TrackSegment
           On(t, s) = Speed ≤ s.speedLimit

- Goal Maintain[TrainAccess(speedLimit)]
  - InformalDef: A train should stay within the access limit.

FormalDef: ∀ t, s: Train, s: TrackSegment
           On(t, s) = Speed ≤ s.speedLimit

Building a Domain Model

- Goal Maintain[TrainLocation(ofTrain, ofLocation)]: A train should be located at a specific location.
- Goal Maintain[TrainAccess(speedLimit)]: A train should access a specific speed limit.

Abstract to Higher-Level Goals by Asking Why?

- If it enters a closed gate, it could get switched onto the wrong track
- Achieving this requires an additional subgoal!
  - Gate closed when switch is in wrong position
Refine to Concrete Goals by Asking How?

Goal Modeling Takeaways

- Refine abstract goal into precise machine specification. Steps:
  - Refine goals to make them concrete
  - State goals precisely
  - Analyze goals for conflict
  - Develop domain models from goals
  - Assign subgoals to machines
  - Derive machine interfaces from goals
  - Identify operations from interfaces
  - Specify operations to ensure goals

Operationalization of Goals

- Specify operations so that the goal will be achieved
- Operation SendCommandMessage
  - Trigger: No message sent in time window

Formal Analysis of Goal Model

- Automated tool support
  - Detect goal conflicts
  - Prove that subgoals imply goal
- Relies on formal specification of goals
  - Hard to do at top level
  - E.g., safe train operation
  - Expensive
  - Worth it for safety-critical applications

Questions?