Requirements Modeling

15-413: Introduction to Software Engineering

Jonathan Aldrich

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

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

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 scenarioes
    - 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

A. v. Lamsweerde, KAOS; M. Jackson, Problem Frames

- 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

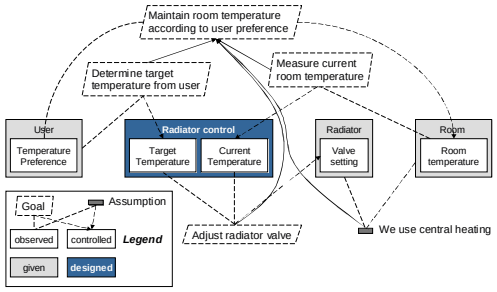
Relationships Among Terms

- $R \wedge As \wedge D = G$
  - 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
- $S \wedge Ac \wedge D = R$
  - 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

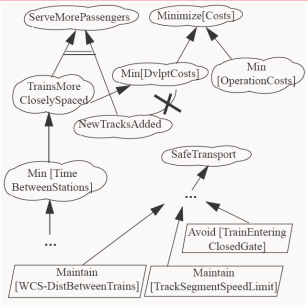
Goal Modeling: Simple Example



Goal Modeling

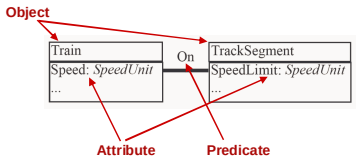
- Goal Refinement
  - AND-refinement: satisfying all subgoals will satisfy goal
  - OR-refinement: satisfying one subgoal will satisfy goal
- Conflict link
  - Satisfaction of one goal may preclude satisfying the other
- Responsibility link
  - States that an agent can commit to act in such a way that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

Extended Example: BART



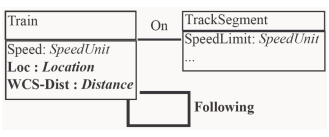
Building a Domain Model

Goal Maintain[TrackSegmentSpeedLimit]
InformalDef *A train should stay below the maximum speed the track segment can handle.*
FormalDef ∀ tr: Train, s: TrackSegment : On(tr, s) ⇒ tr.Speed ≤ s.SpeedLimit

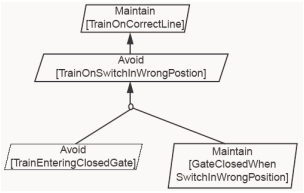


Building a Domain Model

Goal Maintain[WCS-DistBetweenTrains]
InformalDef *A train should never get so close to a train in front so that if the train in front stops suddenly (e.g., derailment) the next train would hit it.*
FormalDef ∀ tr1, tr2: Train : Following(tr1, tr2) ⇒ tr1.Loc - tr2.Loc > tr1.WCS-Dist

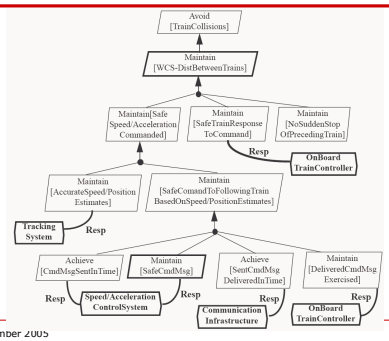


Abstract to Higher-Level Goals by Asking Why?

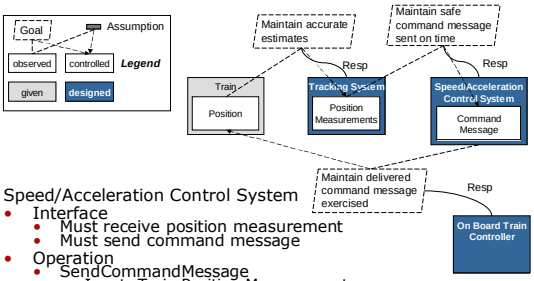


- If it enters a closed gate, it could get switched onto the wrong track
- Achieving this requires an additional subgoal!
  - Gate closed when switch is in wrong position

Refine to Concrete Goals by Asking How?



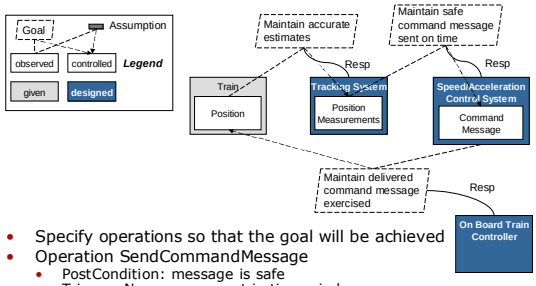
Derive Machine Interfaces and Operations



Speed/Acceleration Control System

- Interface
  - Must receive position measurement
  - Must send command message
- Operation
  - SendCommandMessage
    - Input: Train Position Measurement
    - Output: Command message

Operationalization of Goals



- Specify operations so that the goal will be achieved
- Operation SendCommandMessage
  - PostCondition: message is safe
  - 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

Goal Modeling Takeaways

Refine abstract goal into precise machine specification. Steps:

- Refine goals to make them concrete
  - But also abstract them to find higher-level objectives
- 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

Questions?