# Models and Modeling

Angelo Gargantini

April 30, 2024

Overview
How to model your system
Declarative vs operational notations
Finite State Machines
Logic
Temporal logic
Abstract State Machines

## Why modeling

- From wind-tunnels to Navier-Stokes equations to circuit diagrams to finite-element models of buildings, engineers in all fields of engineering construct and analyze models.
- Several advantages:
  1. analysis and test can be performed before the actual artifact is constructed,
  2. Models permit us to start analysis earlier and repeat it as a design evolves, and allows us to apply analytic methods that cover a much larger class of scenarios than we can explicitly test.
  3. Importantly, many of these analyses may be automated.

## Model characteristics

Compact  A model must be representable and manipulable in a reasonably **compact** form. Models intended for human inspection and reasoning must be small enough to be comprehensible. Models intended solely for automated analysis may be far too large and complex for human comprehension, but must still be sufficiently small or regular for computer processing.

Predictive  A model used in analysis or design must represent some salient characteristics of the modeled artifact well enough to distinguish between "good" and "bad" outcomes of analysis, with respect to those characteristics.

Multiple models  Typically, no single model represents all characteristics well enough to be useful for all kinds of

Overview
How to model your system
Declarative vs operational notations
Finite State Machines
Logic
Temporal logic
Abstract State Machines

## Model characteristics

Semantically meaningful  Beyond distinguishing between predictions
of success and failure, it is usually necessary to
interpret analysis results in a way that permits
diagnosis of the causes of failure. If a finite-element
model of a building predicts collapse in a category five
hurricane, we want to know enough about that
collapse to suggest revisions to the design. Likewise,
if a model of an accounting system predicts a failure
when used concurrently by several clients, we need a
description of that failure sufficient to suggest
possible revisions.

Sufficiently general  Models intended for analysis of some important
characteristic (e.g., withstanding earthquakes or
concurrent operation by many clients) must be

Overview
**How to model your system**
Declarative vs operational notations
Finite State Machines
Logic
Temporal logic
Abstract State Machines

Notations for Modeling
Choosing a Notation

## How to model your system

- Focus primarily on the SUT
- Show only those classes (or subsystems) associated with the SUT and whose values will be needed in the test data
- Include only those operations that you wish to test
- Include only the data fields that are useful for modeling the behavior of the operations that will be tested
- Replace a complex data field, or a class, by a simple enumeration. This allows you to limit the test data to several carefully chosen example values (one for each value of the enumeration).

Overview
How to model your system
Declarative vs operational notations
Finite State Machines
Logic
Temporal logic
Abstract State Machines

Notations for Modeling
Choosing a Notation

# Notations for modeling

- Pre/post (or state-based) notations
- Transition-based notations
- History-based notations
- Functional notation
- Operational notations
- Statistical notations

Overview
How to model your system
Declarative vs operational notations
Finite State Machines
Logic
Temporal logic
Abstract State Machines

Notations for Modeling
Choosing a Notation

## Choosing a Notation

1. Pre/post notations are best for data-oriented systems. Transition-based notations are best for control-oriented systems.

2. Model-based testing requires an accurate model, written in a formal modeling notation that has precise semantics.

3. *two main families*: *operational* (also called "imperative") and *declarative*. An operational modeller asks "how would I make X happen?". A declarative modeller asks "how would I recognize that X has happened?". In many contexts, it is more natural and concise to use a declarative description. Often one need only to describe the rules of the game and what it means to have won, and not even think about what moves will be needed (and which must be avoided) in order to win.

# Finite State Machines

# Propositional Logic

# Temporal Logic

# Abstract State Machines