Modeling Specification (and Verification)

Alcino Cunha

The heavy chair

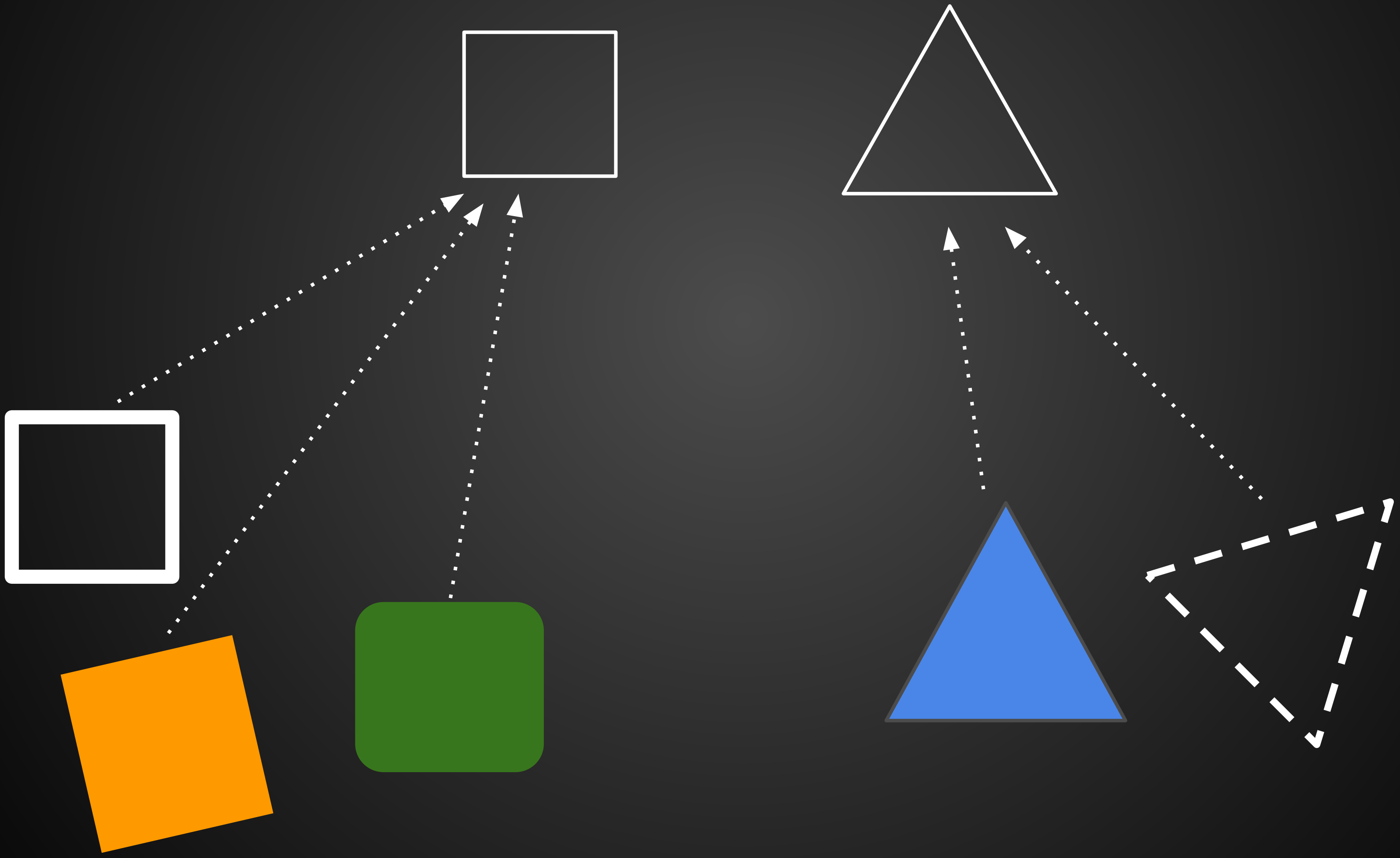
Modeling

What is a model?

“A simplified description, especially a mathematical one, of a system or process, to assist calculations and predictions”

Google Dictionary

What is a model?



What is a software model?

“I use the term ‘model’ for a description of a software abstraction.”

“An abstraction is not a module, or an interface, class, or method; it is a structure, pure and simple - an idea reduced to its essential form.”

“The core of software development, therefore, is the design of abstractions.”

Daniel Jackson

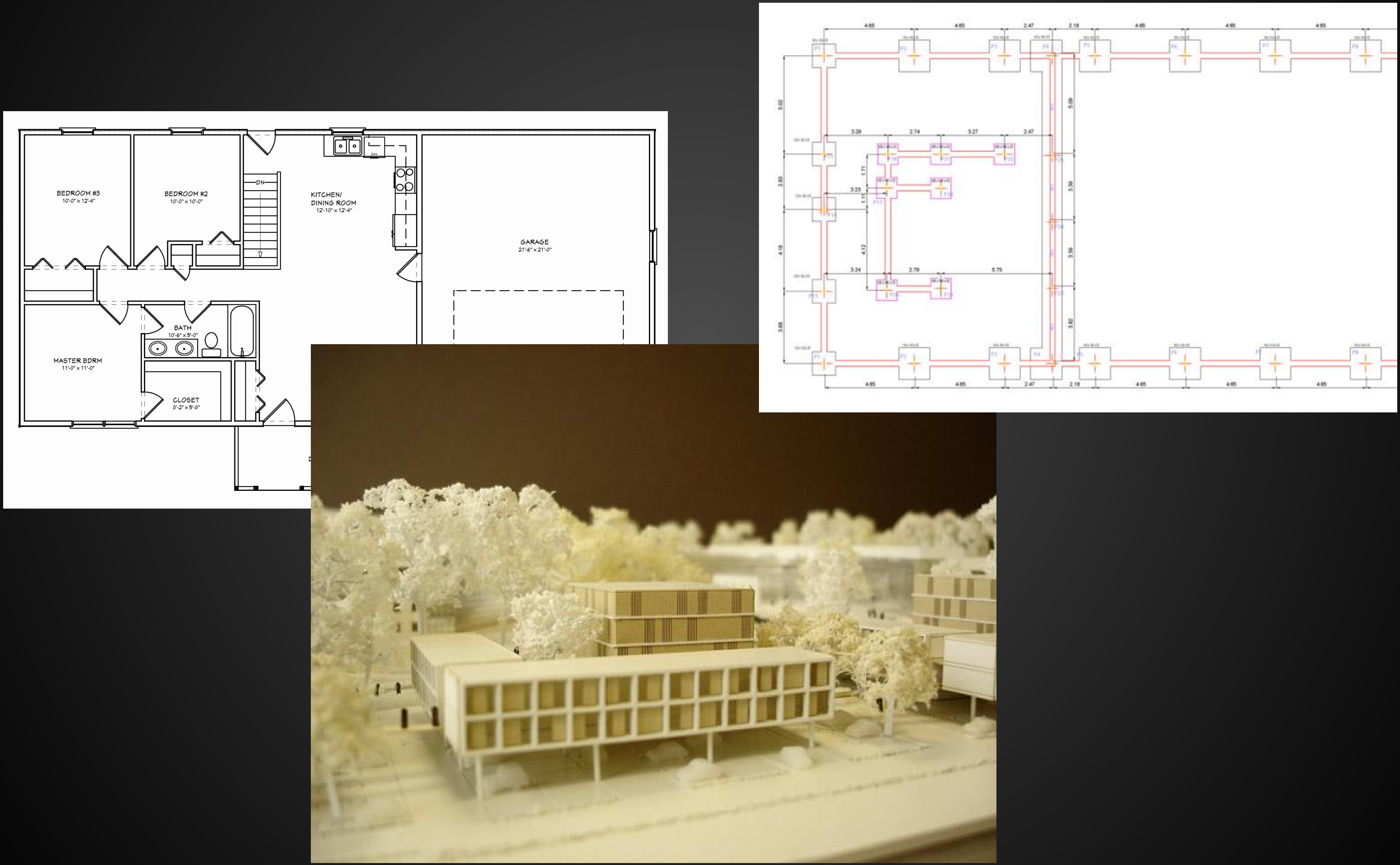
Why model?

- To explain and predict phenomena
- To predict other similar (new) phenomena
- To build and design systems

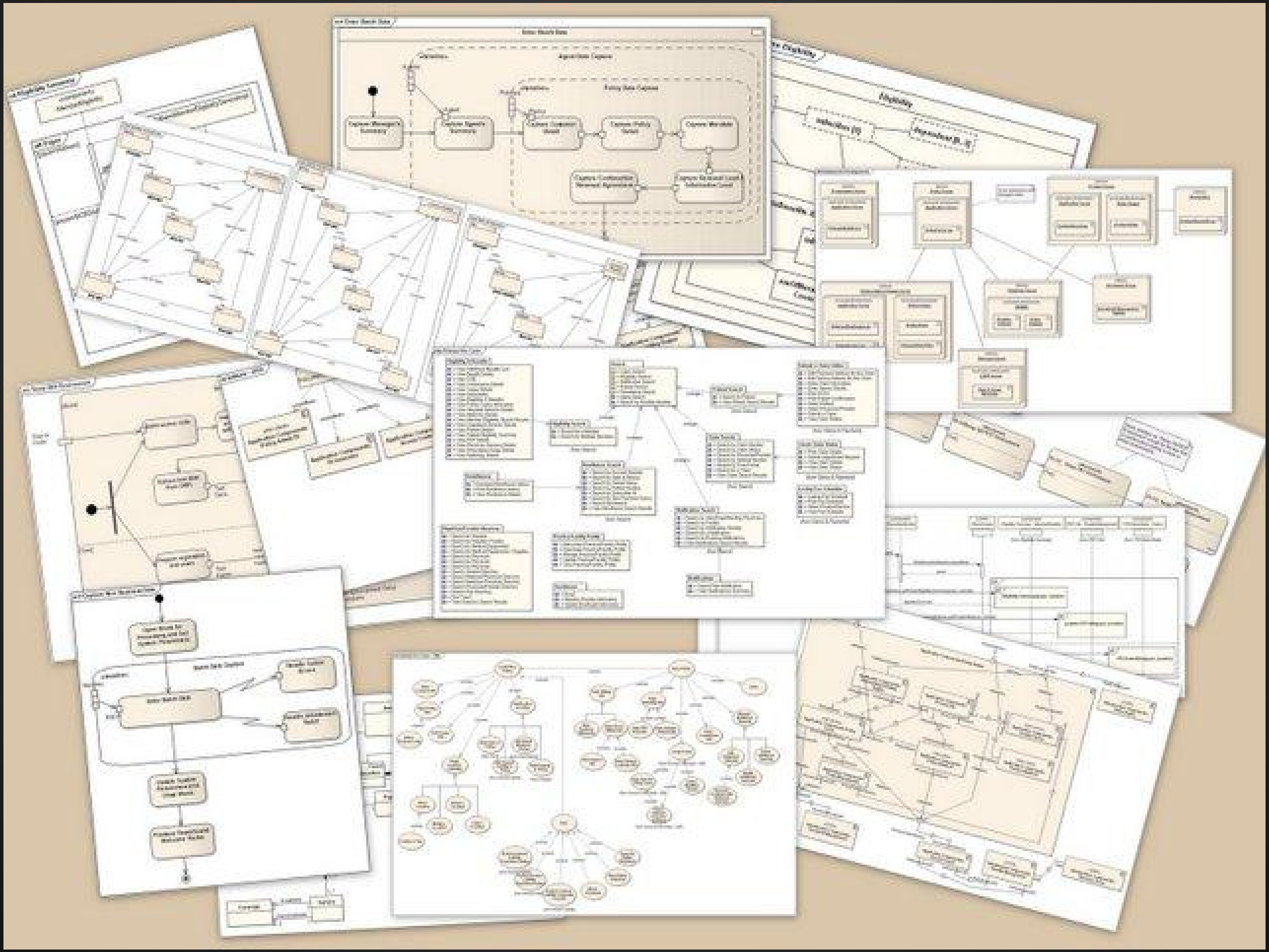
Why model software?

- Clarify and validate requirements
- Verify assumptions and desired properties
- Explore alternative designs
- Documentation
- ...

How to model?


BEDROOM #3
10'-0" × 12'-4"
BEDROOM #2
10'-0" × 10'-0"
KITCHEN/
DINING ROOM
12'-10" × 12'-4"
GARAGE
21'-6" × 21'-0"
DN
BATH
10'-6" × 5'-0"
MASTER BDRM
11'-0" × 11'-0"
CLOSET
8'-2" × 5'-8"
4.65
2.47
2.18
5.02
5.09
3.29
2.74
3.27
3.23
2.83
3.39
4.12
4.18
3.24
2.79
5.75
3.68
3.82

How to model?



How to model?

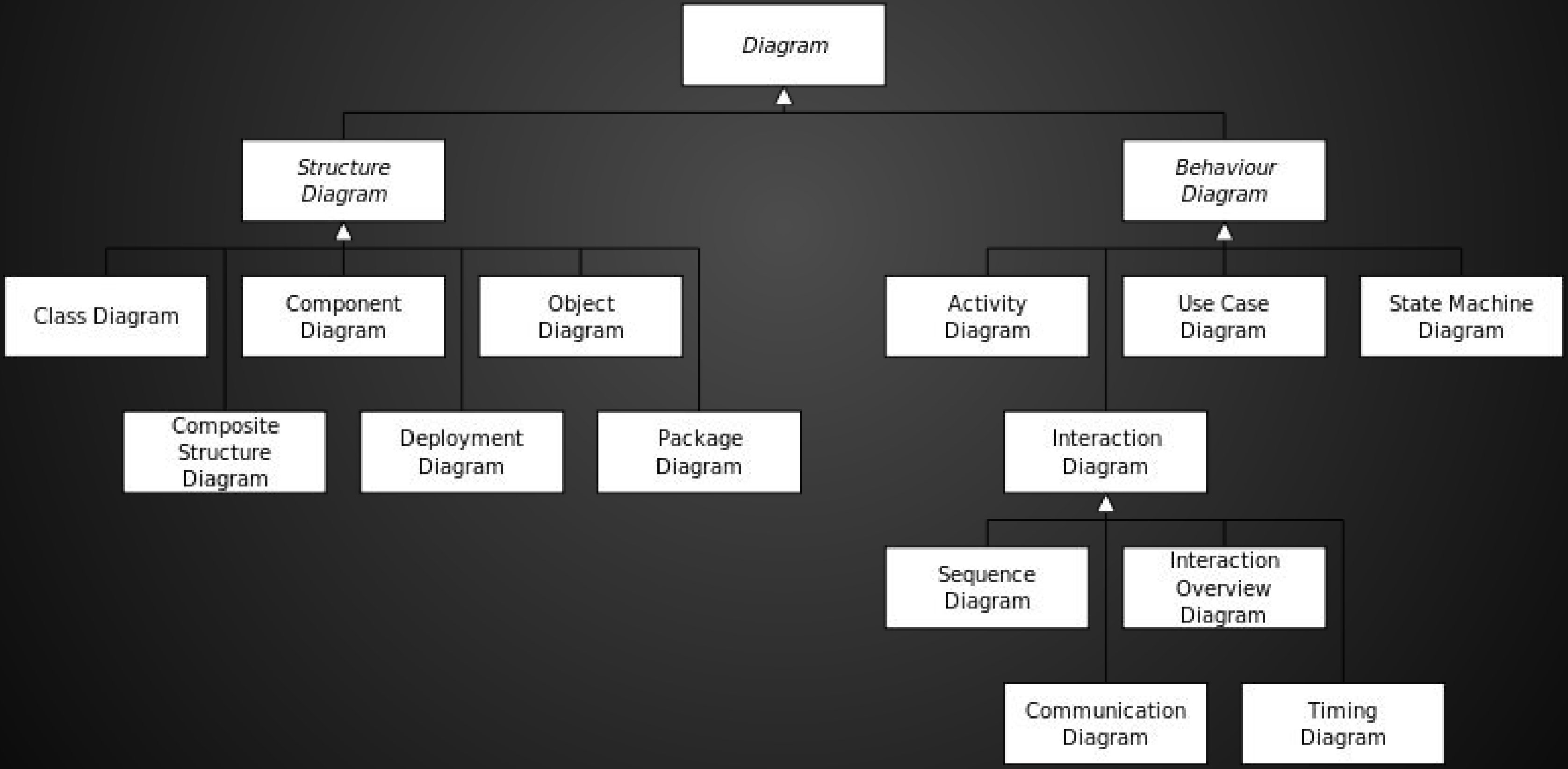
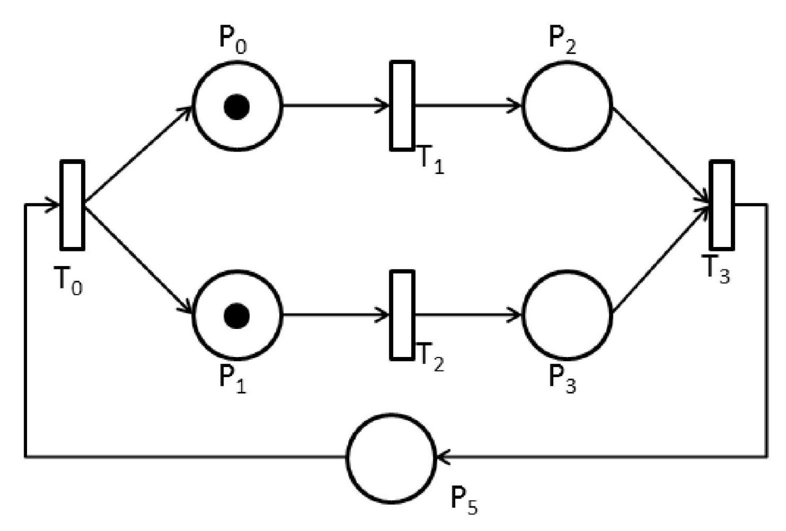
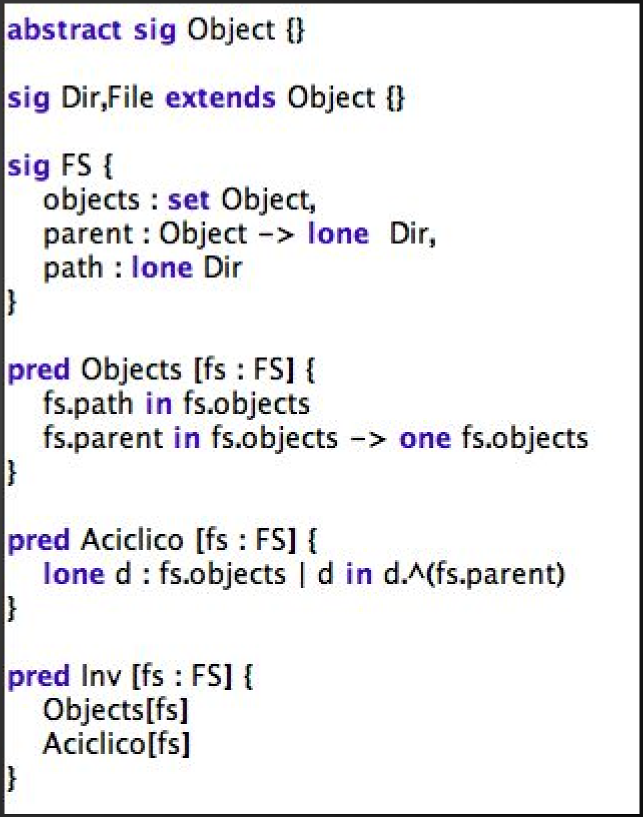


Diagram
Structure Diagram
Behaviour Diagram
Class Diagram
Component Diagram
Object Diagram
Composite Structure Diagram
Deployment Diagram
Package Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Interaction Diagram
Sequence Diagram
Interaction Overview Diagram
Communication Diagram
Timing Diagram

How to model?



VS



```
abstract sig Object {}

sig Dir,File extends Object {}

sig FS {
   objects : set Object,
   parent : Object -> lone  Dir,
   path : lone Dir
}

pred Objects [fs : FS] {
   fs.path in fs.objects
   fs.parent in fs.objects -> one fs.objects
}

pred Aciclico [fs : FS] {
   lone d : fs.objects | d in d.^(fs.parent)
}

pred Inv [fs : FS] {
   Objects[fs]
   Aciclico[fs]
}
```

What is a good modeling language?

- Notation tailored for abstraction
- Support for automated analysis
- Formal but not intimidating

How to model?

“All models are wrong, but some are useful.”

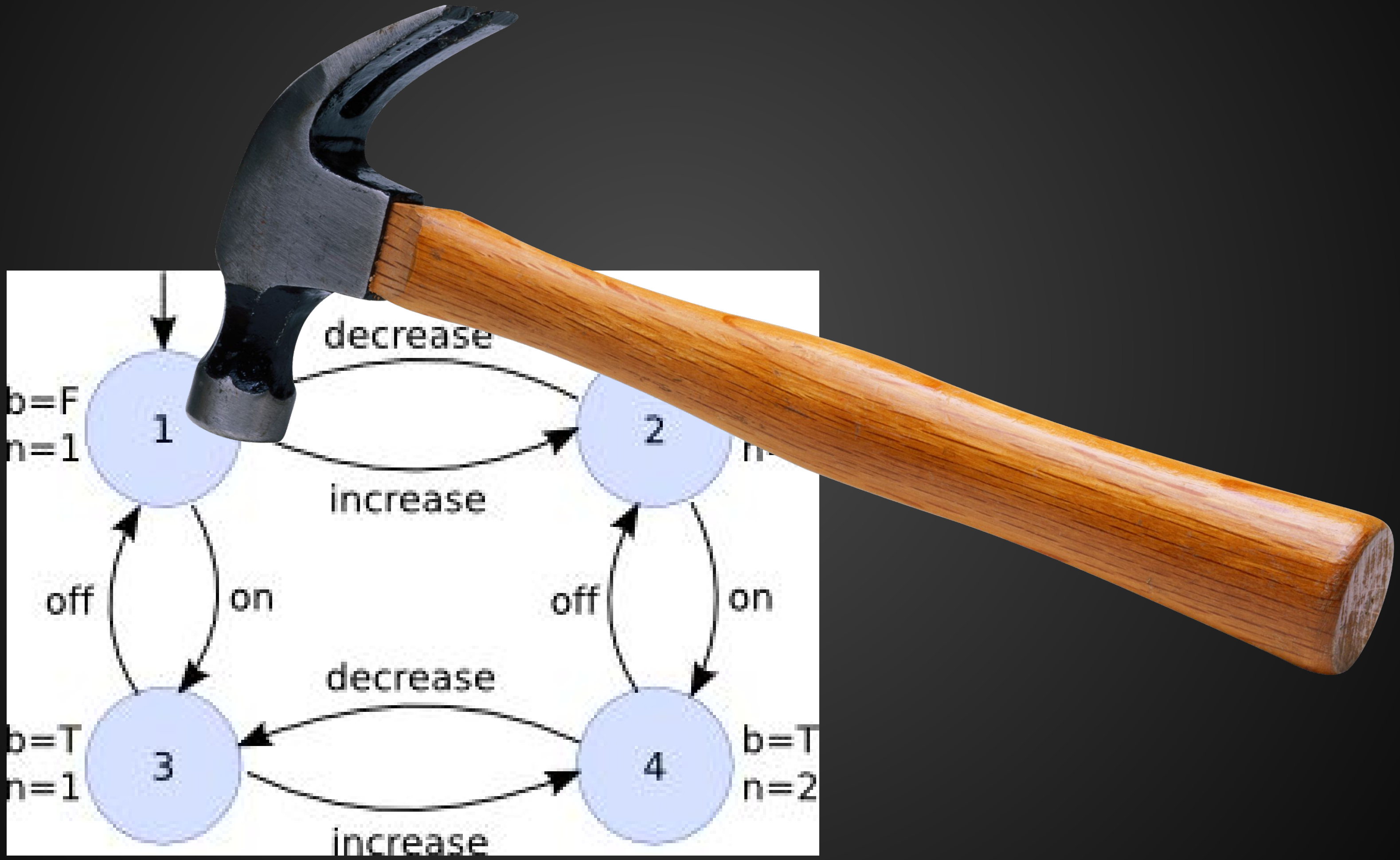
George E. P. Box

How to model?

“If I had a hammer I'd hammer in the morning,
I'd hammer in the evening all over this land.”

Pete Seeger and Lee Hays

How to model?


decrease
b=F
n=1
1
2
increase
off
on
off
on
decrease
b=T
n=1
3
4
b=T
n=2
increase

Specification

What is a specification?

“An act of identifying something precisely or stating a precise requirement”

Google Dictionary

How to specify?

- Natural language
- Formal language
  - First order logic
  - Relational logic
  - Temporal logic
  - ...

What to specify?

- Safety properties
  - Something bad will not happen
  - Can be contradicted by a finite sequence of steps
- Liveness properties
  - Something good will happen
  - Must be contradicted by an infinite sequence of steps

Verification

What is verification?

“The process of establishing the truth, accuracy, or validity of something”

Google Dictionary

Why verify?

“The first principle is that you must not fool yourself, and you are the easiest person to fool.”

Richard Feynman

How to verify?

- Infinite state systems
  - Likely undecidable
  - Manual verification
    - Maybe assisted by theorem provers
  - Safety can be proved by induction
  - Liveness proofs are similar to termination proofs

How to verify?

- Finite state systems
  - Automatic verification
  - Model checking
    - No bound on the length of counter-examples
  - Bounded model checking
    - The length of counter-examples is bounded

Back to the heavy chair

Take home message

“Simplicity does not precede complexity, but follows it.”

Epigram nº 31, *Alan Perlis*

Take home message

“There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.”

C. A. R. Hoare

This course

- Program
  - SMV: temporal logic, model checking
  - Alloy: relational logic, relational model finding, bounded model checking
- Grading
  - Individual written test (70%, >= 8), Jan 11th
  - Group work (30%, >= 10)
- Contacts
  - alcino@di.uminho.pt