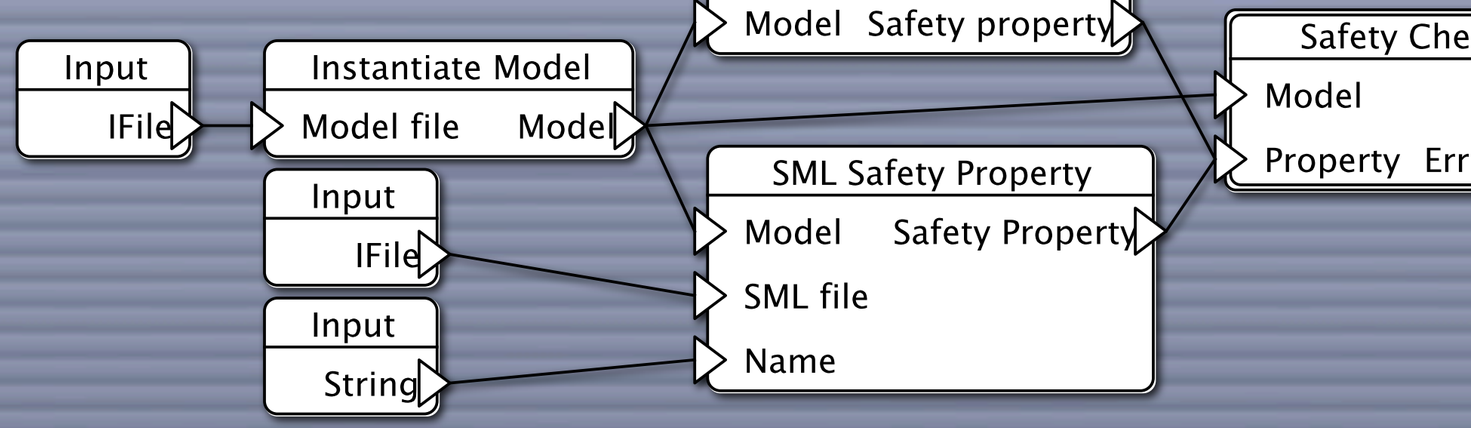
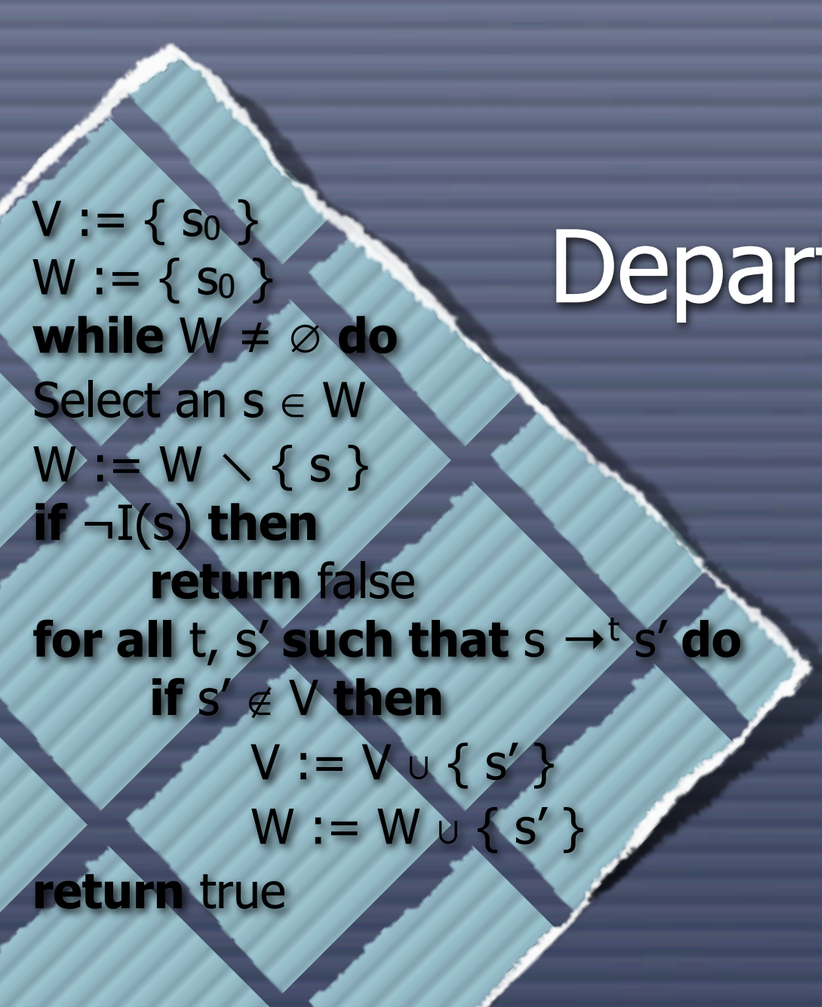
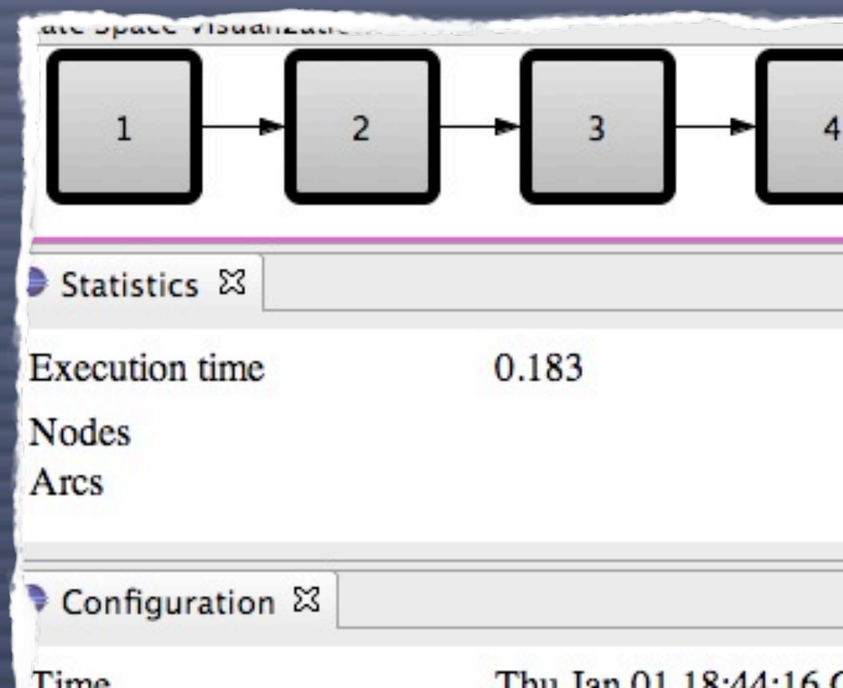


Advanced State Space Methods and ASAP

Michael Westergaard
Department of Computer Science
Aarhus University
mw@cs.au.dk

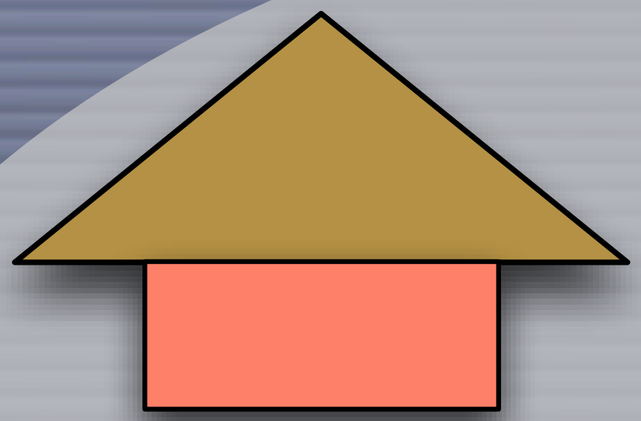




Note…

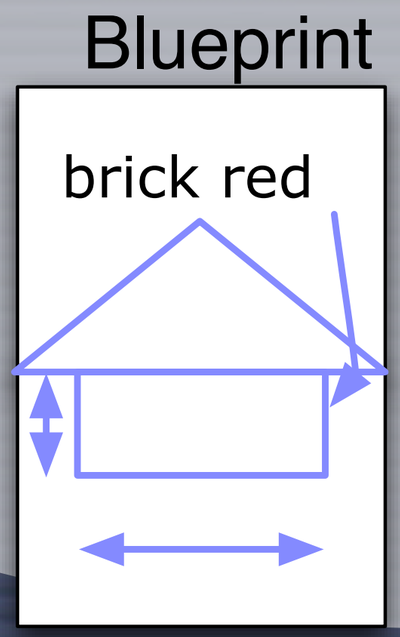
- Please ask questions along the way!

Motivation



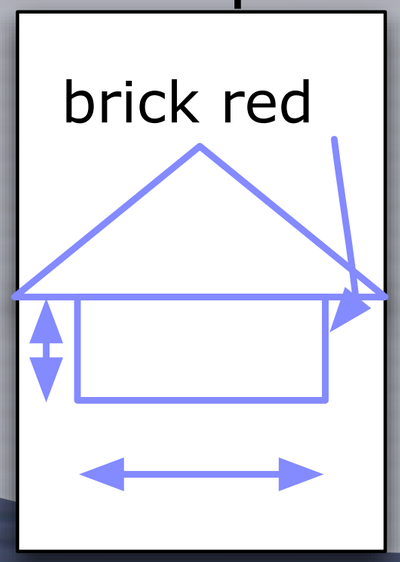
Motivation

Motivation



Motivation

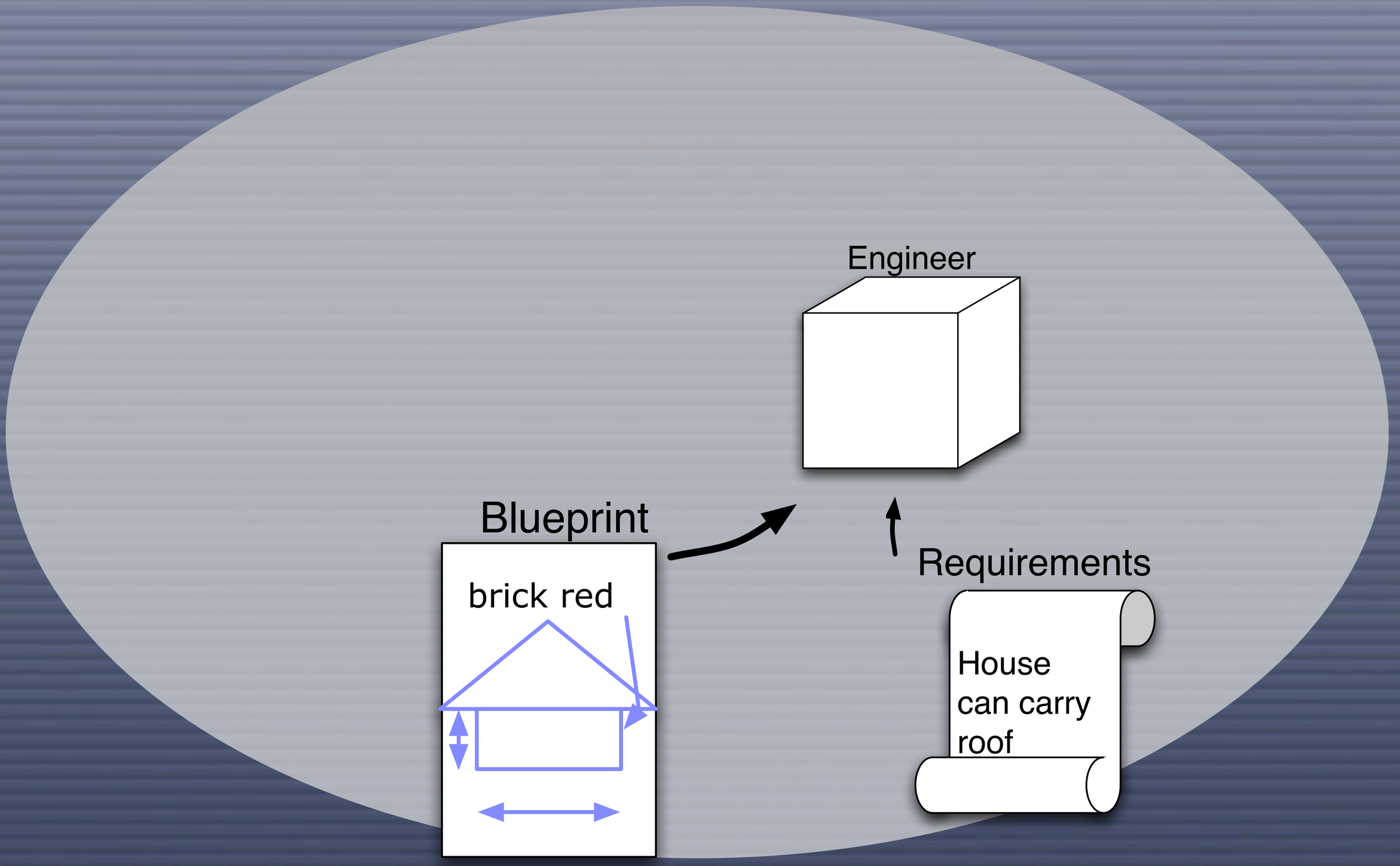
Blueprint



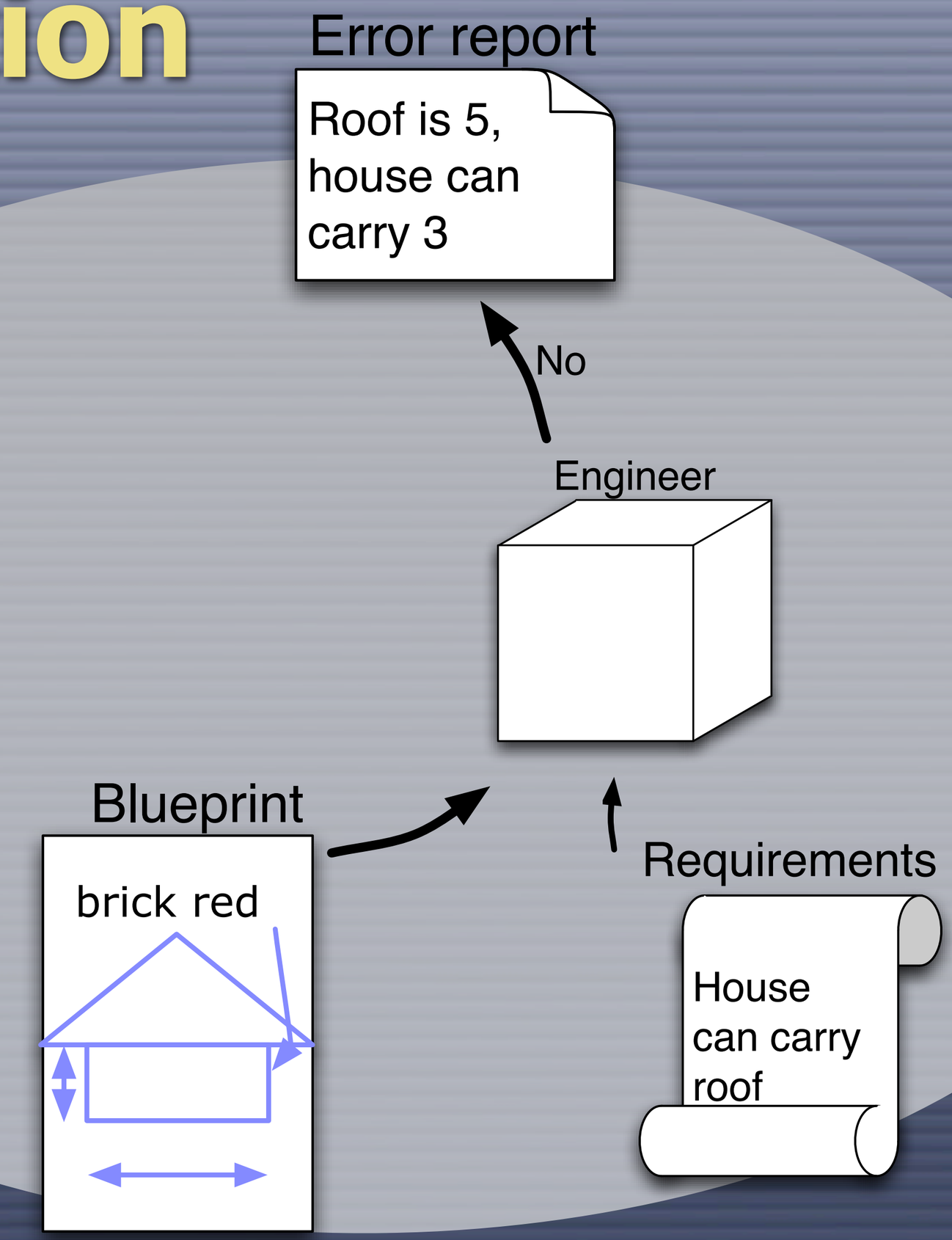
Requirements

House
can carry
roof

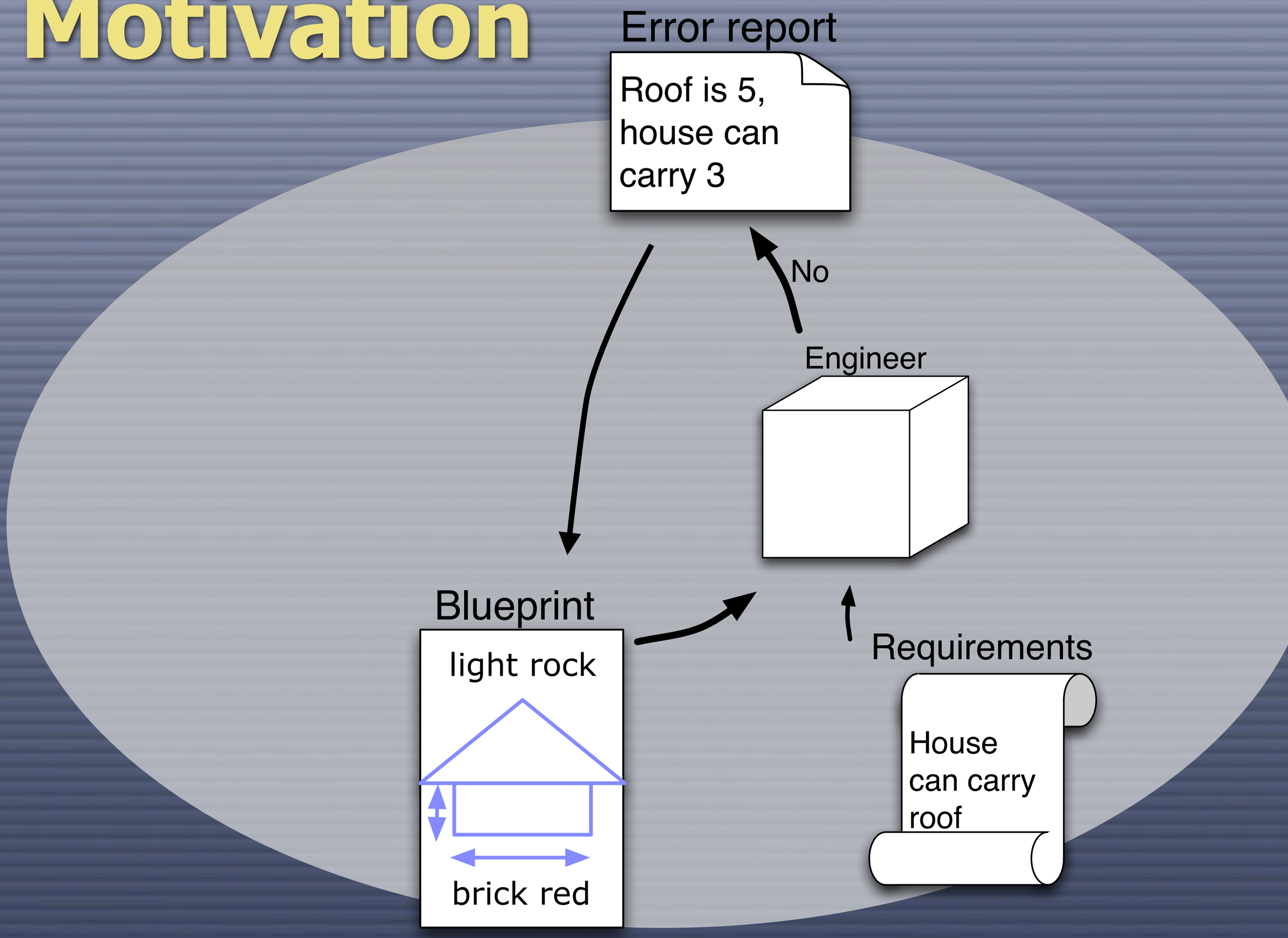
Motivation


Engineer
Blueprint
brick red
Requirements
House
can carry
roof

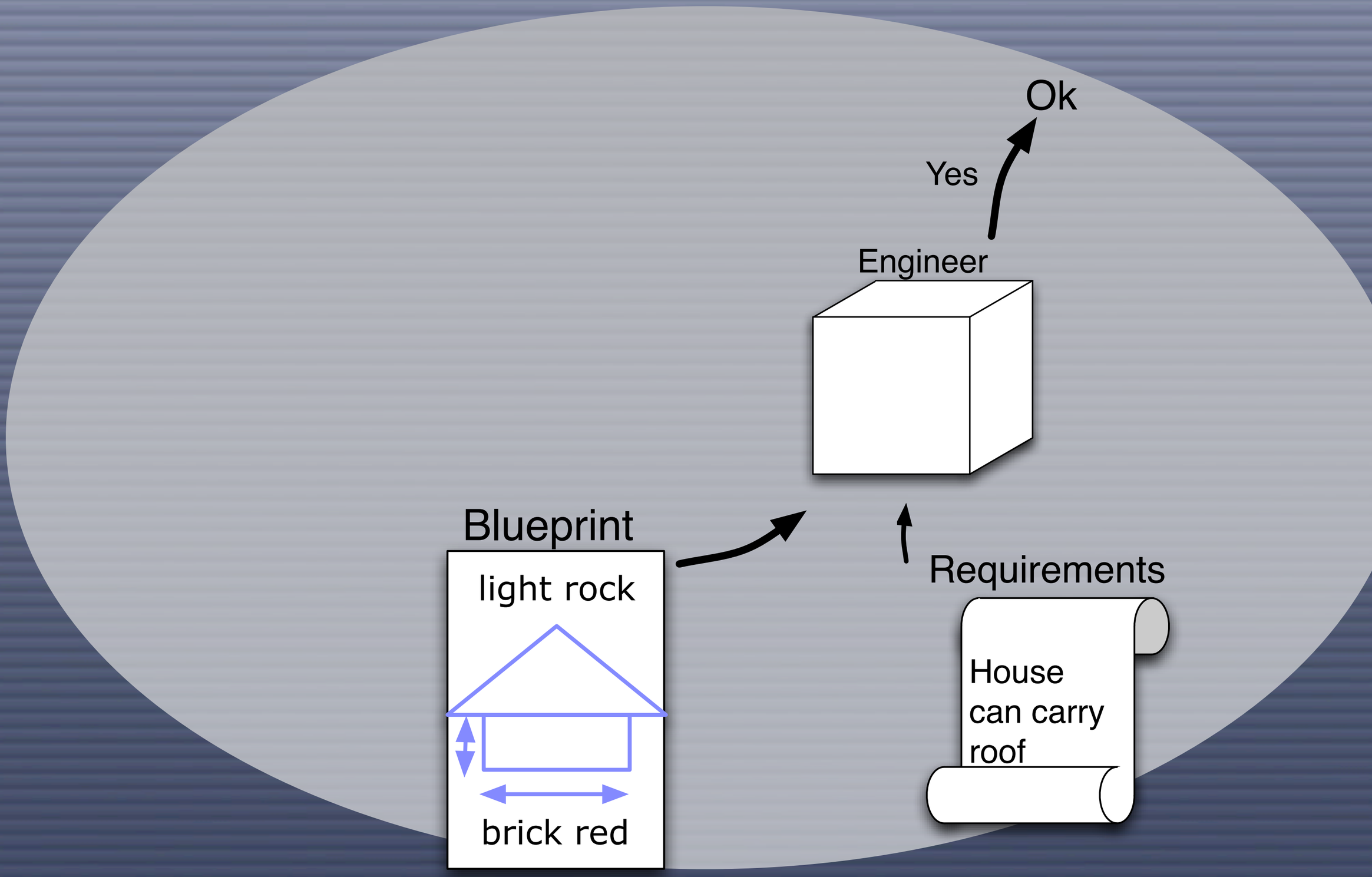
Motivation


Error report
Roof is 5, house can carry 3
No
Engineer
Blueprint
brick red
Requirements
House can carry roof

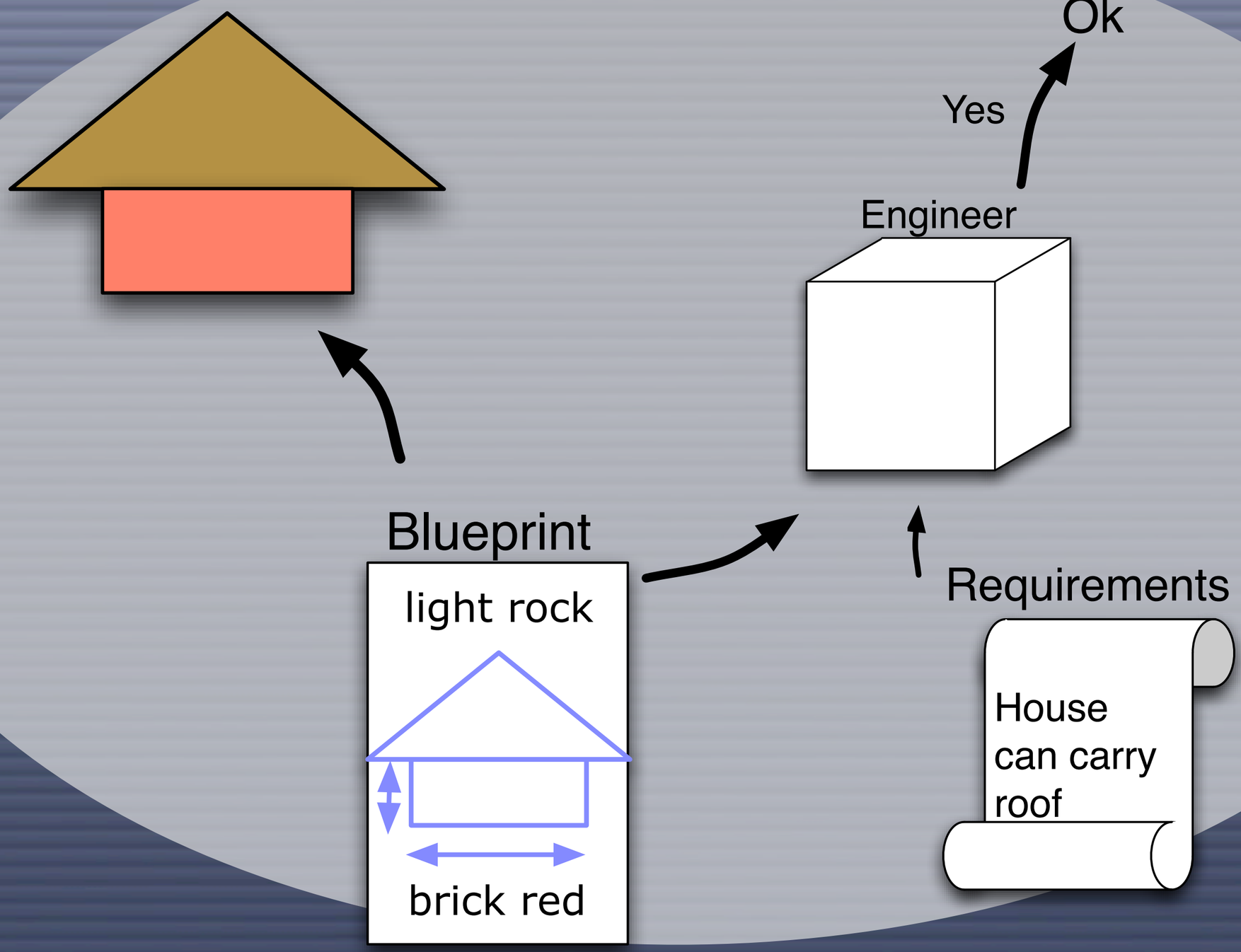
Motivation


Error report
Roof is 5, house can carry 3
No
Engineer
Blueprint
light rock
brick red
Requirements
House can carry roof

Motivation

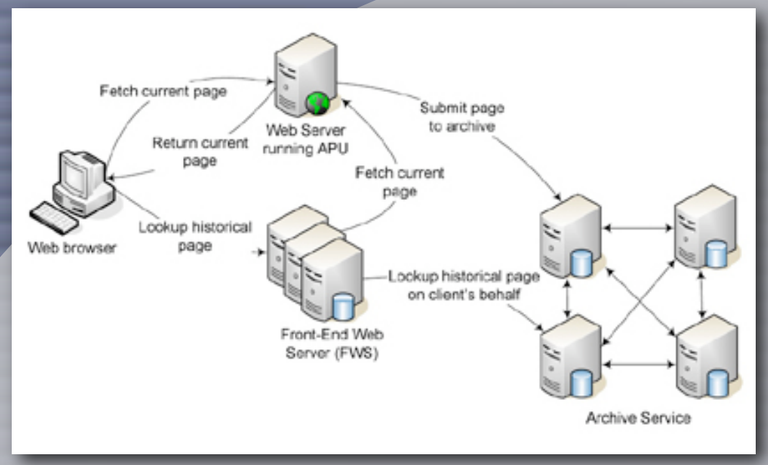

Ok
Yes
Engineer
Blueprint
light rock
brick red
Requirements
House
can carry
roof

Motivation


Ok
Yes
Engineer
Blueprint
light rock
brick red
Requirements
House
can carry
roof

Motivation

System



Error report

Deadlock

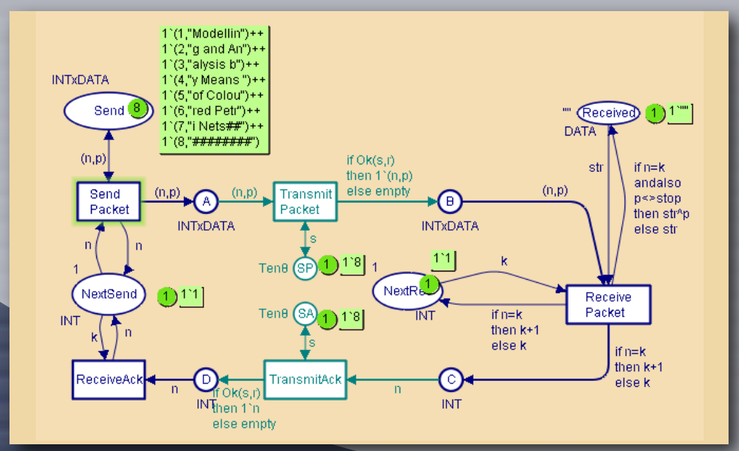
No

Yes

Ok

Verifier

Model



Requirements

No dead-locks

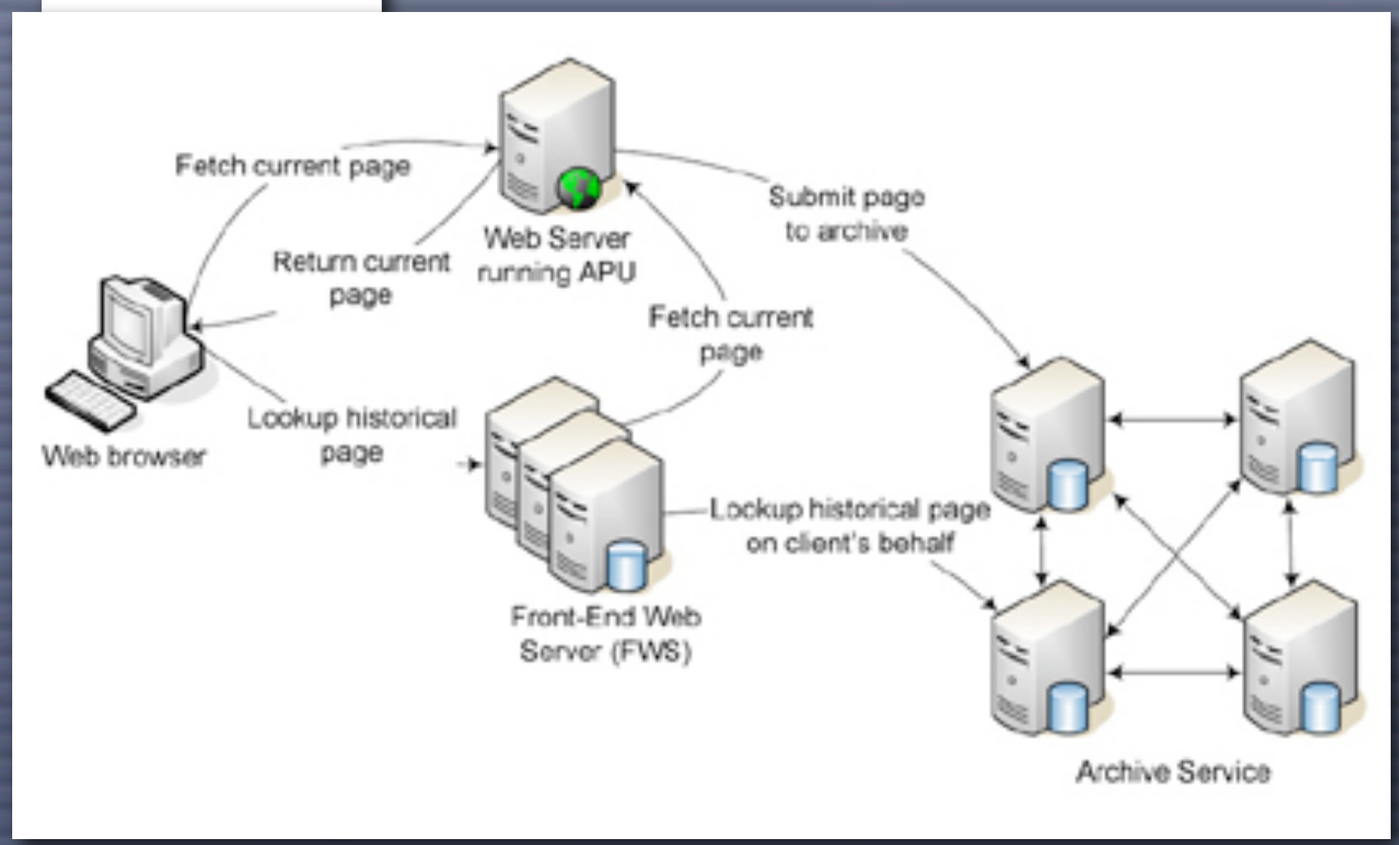
State Space Analysis

State space exploration is a main approach to computer-aided verification:

State Space Analysis

State space exploration is a main approach to computer-aided verification:

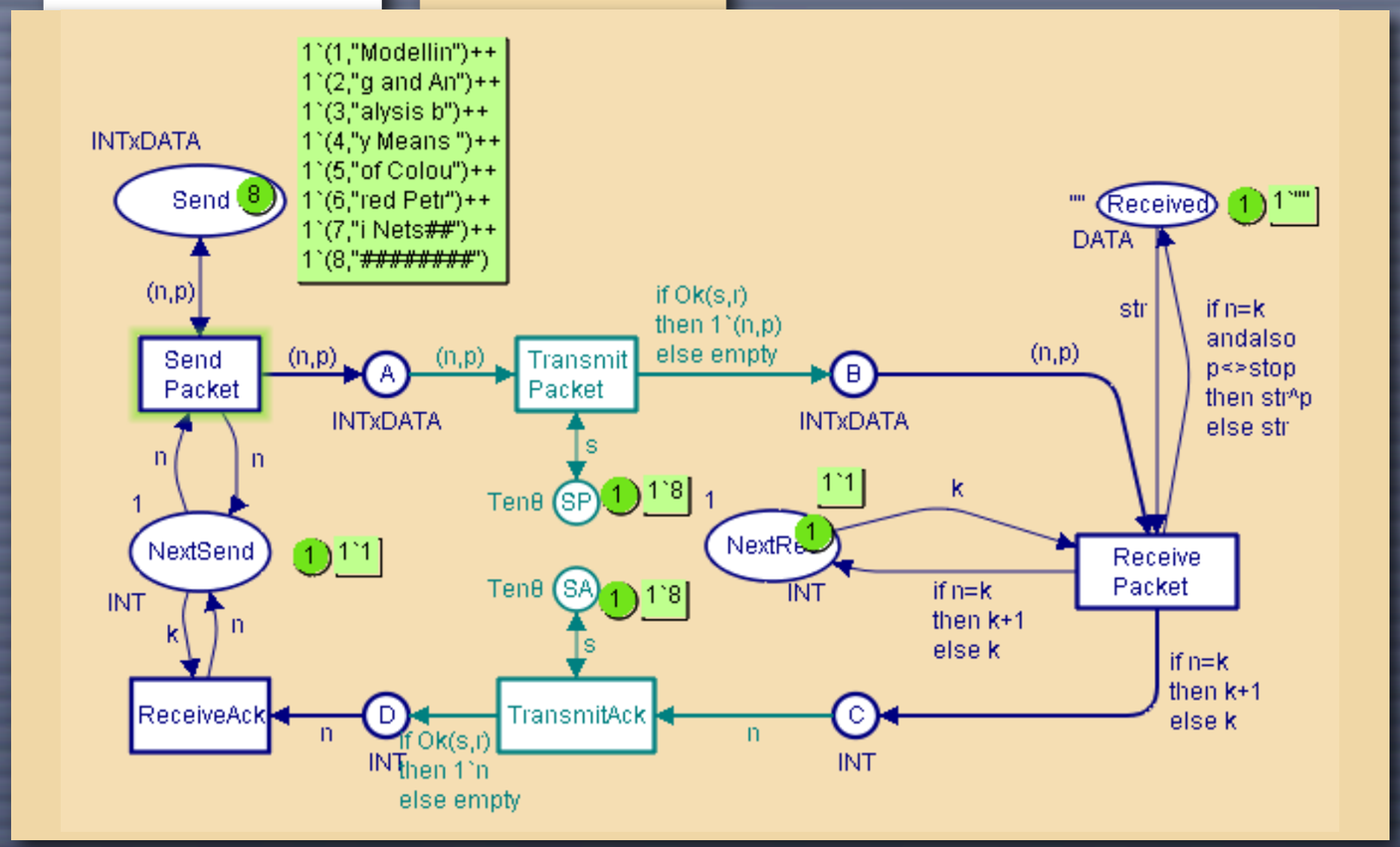
System



State Space Analysis

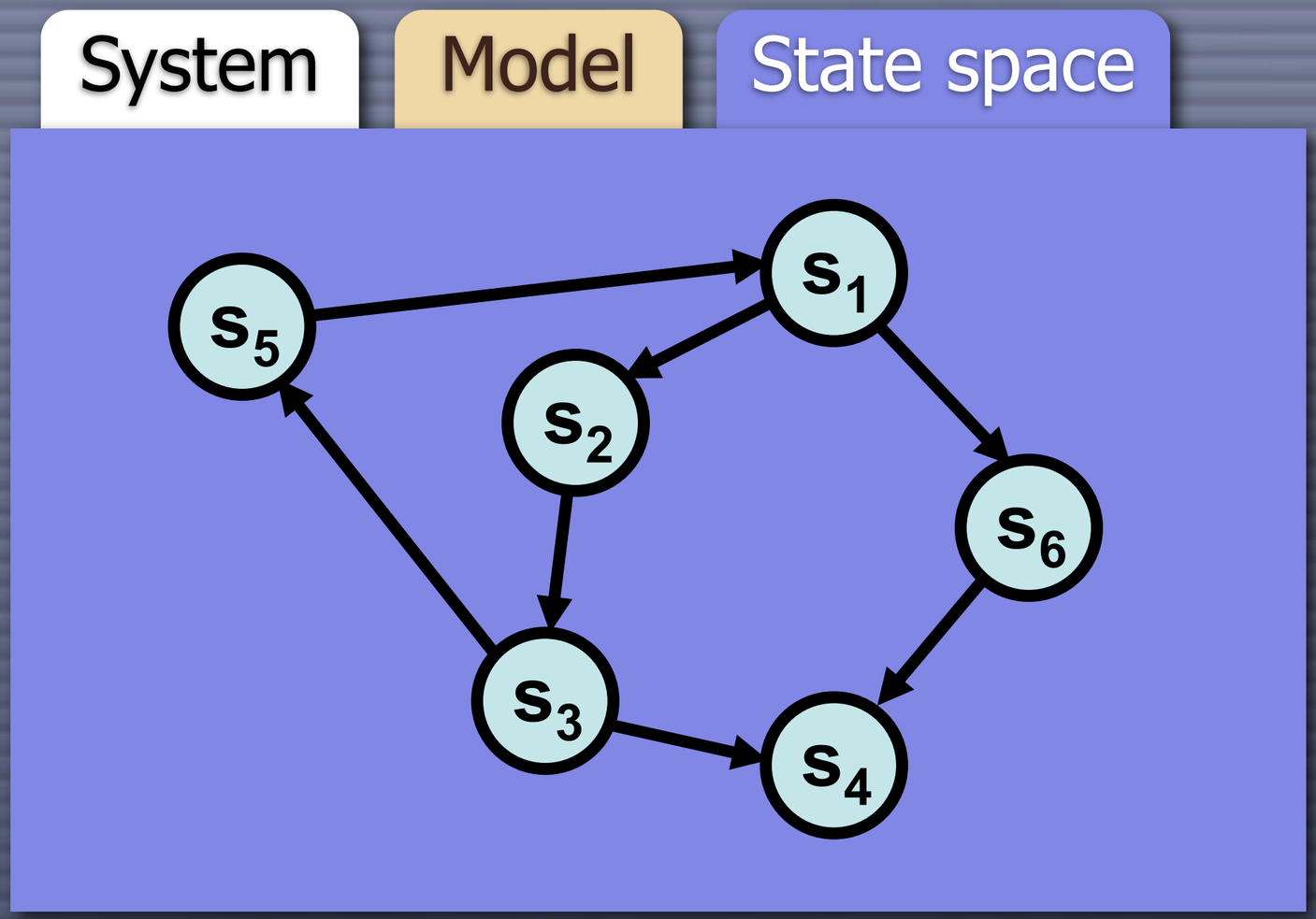
State space exploration is a main approach to computer-aided verification:

System | Model



State Space Analysis

State space exploration is a main approach to computer-aided verification:

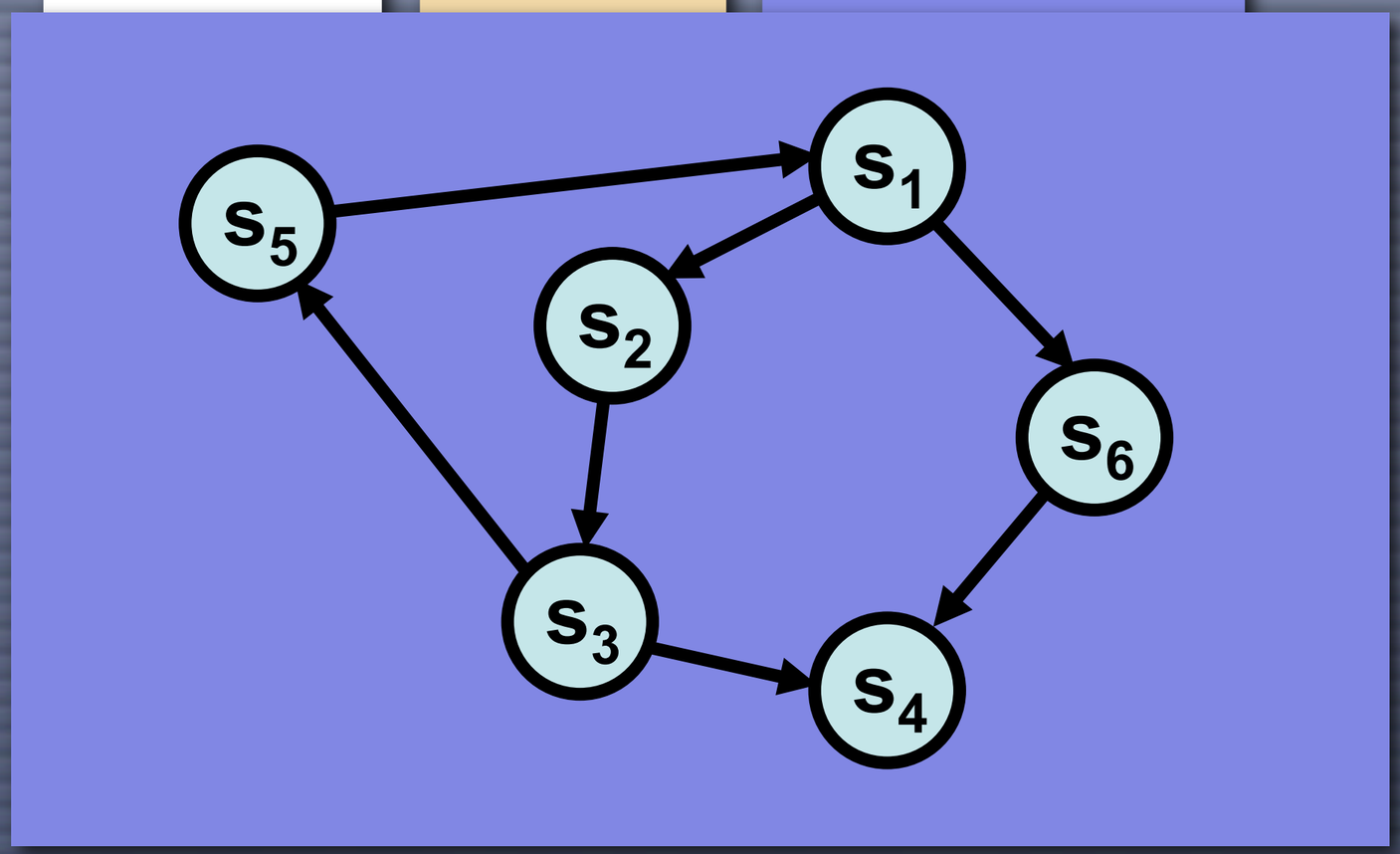


State Space Analysis

State space exploration is a main approach to computer-aided verification:

Nodes: Reachable states
Arcs: Transition executions
Paths: Execution sequences

System | Model | State space

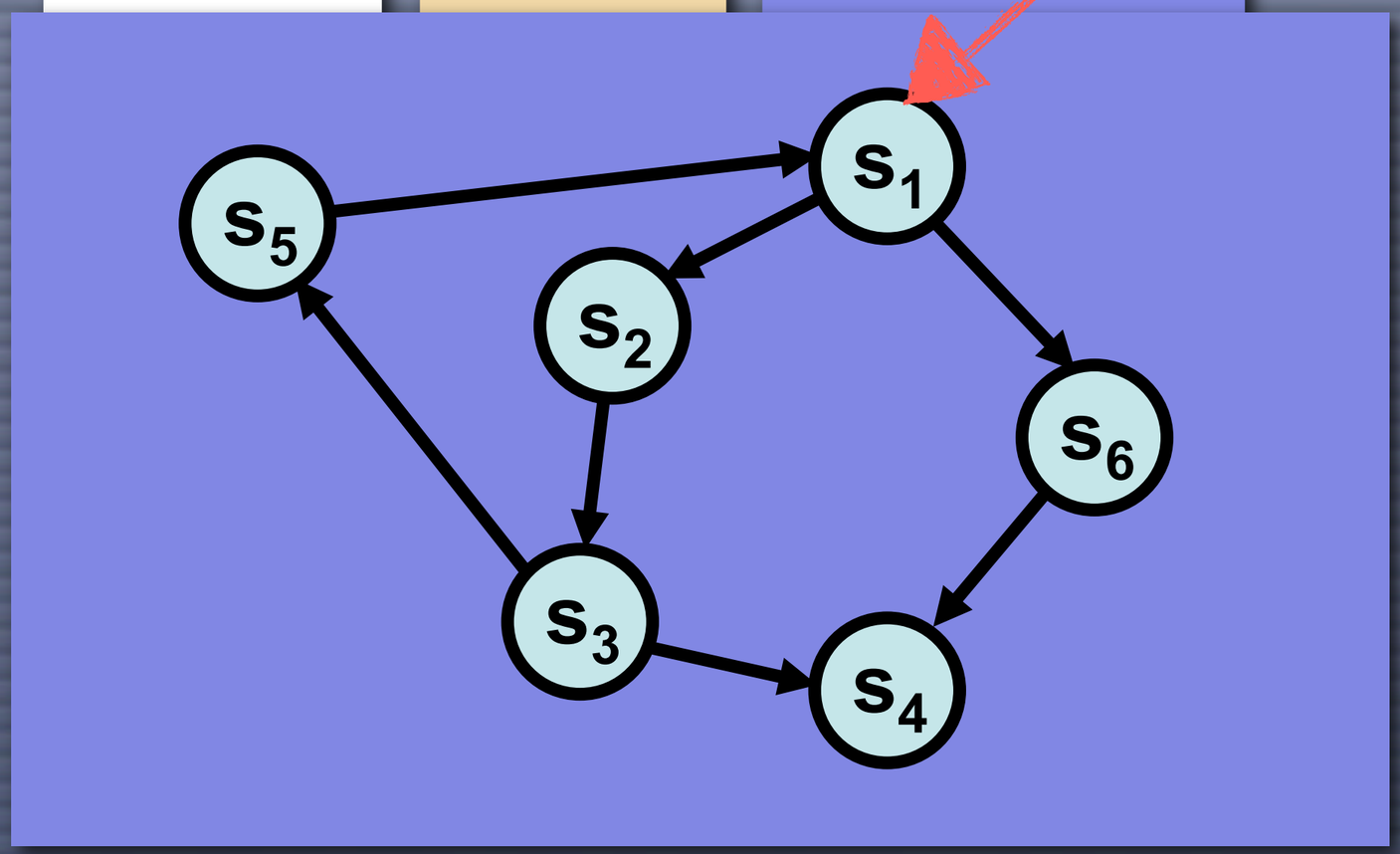


State Space Analysis

State space exploration is a main approach to computer-aided verification:

Nodes: Reachable states
Arcs: Transition executions
Paths: Execution sequences

System | Model | State space

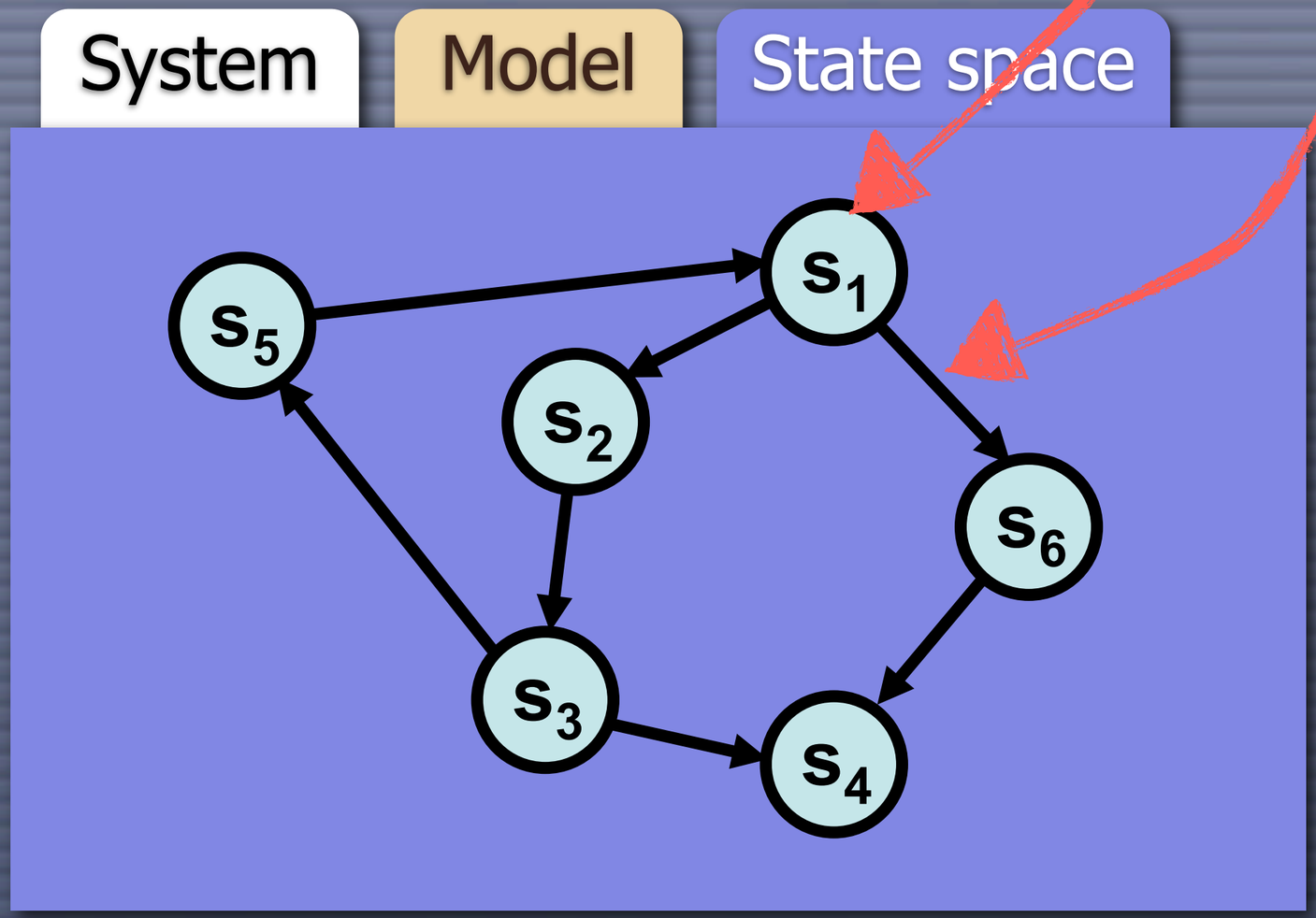


State Space Analysis

State space exploration is a main approach to computer-aided verification:

Nodes: Reachable states
Arcs: Transition executions
Paths: Execution sequences

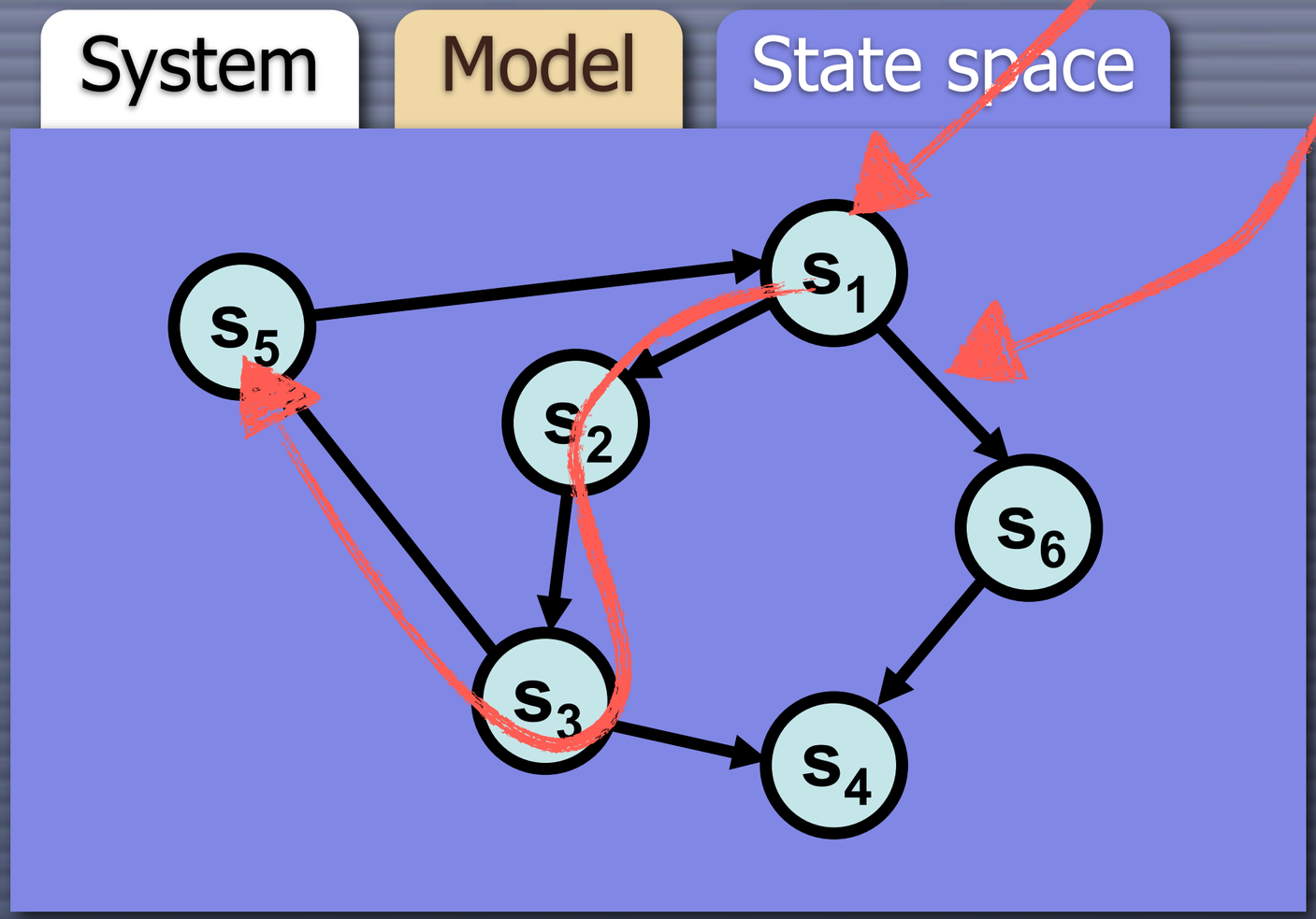
System | Model | State space



State Space Analysis

State space exploration is a main approach to computer-aided verification:

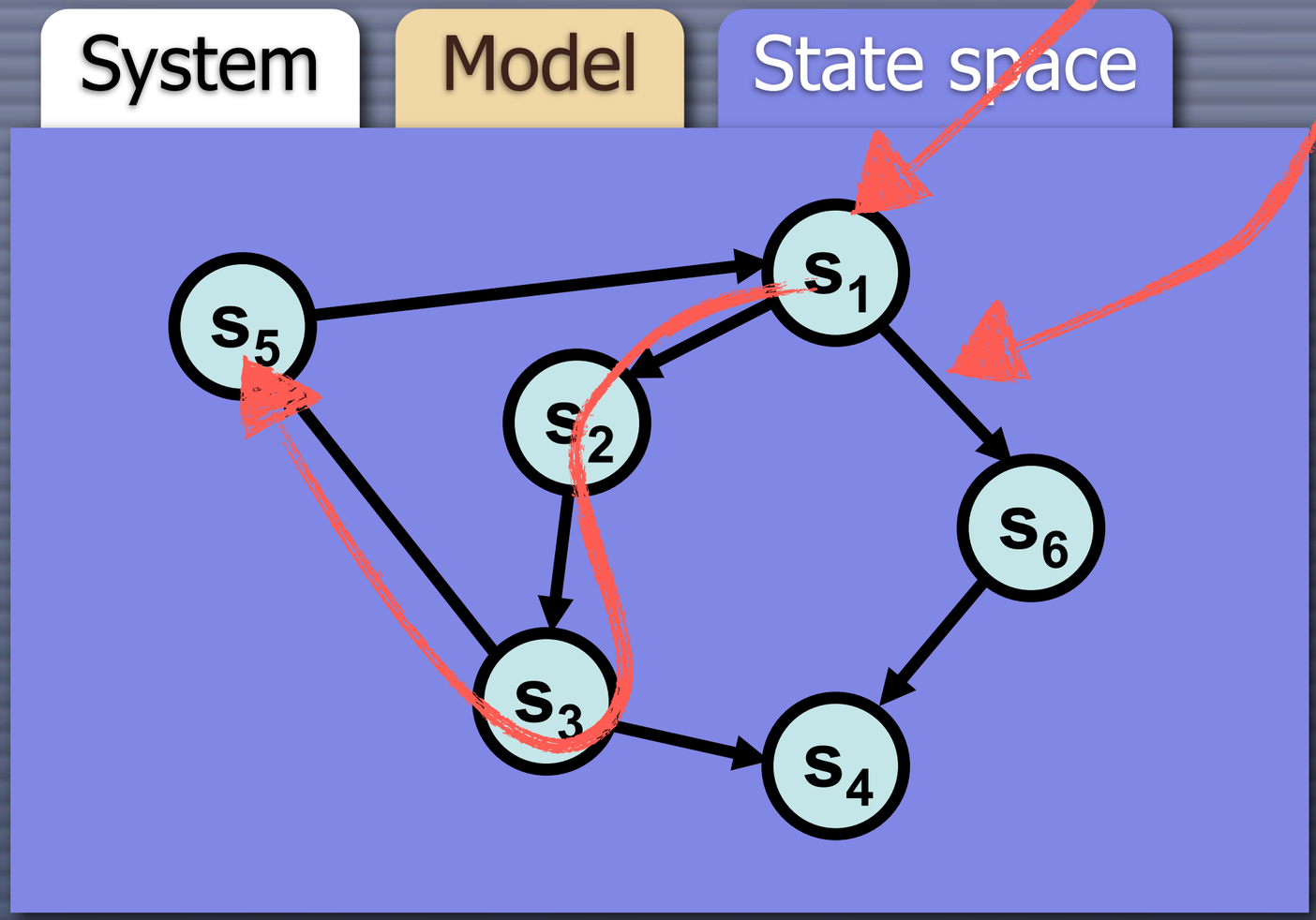
Nodes: Reachable states
Arcs: Transition executions
Paths: Execution sequences



State Space Analysis

State space exploration is a main approach to computer-aided verification:

System | Model | State space



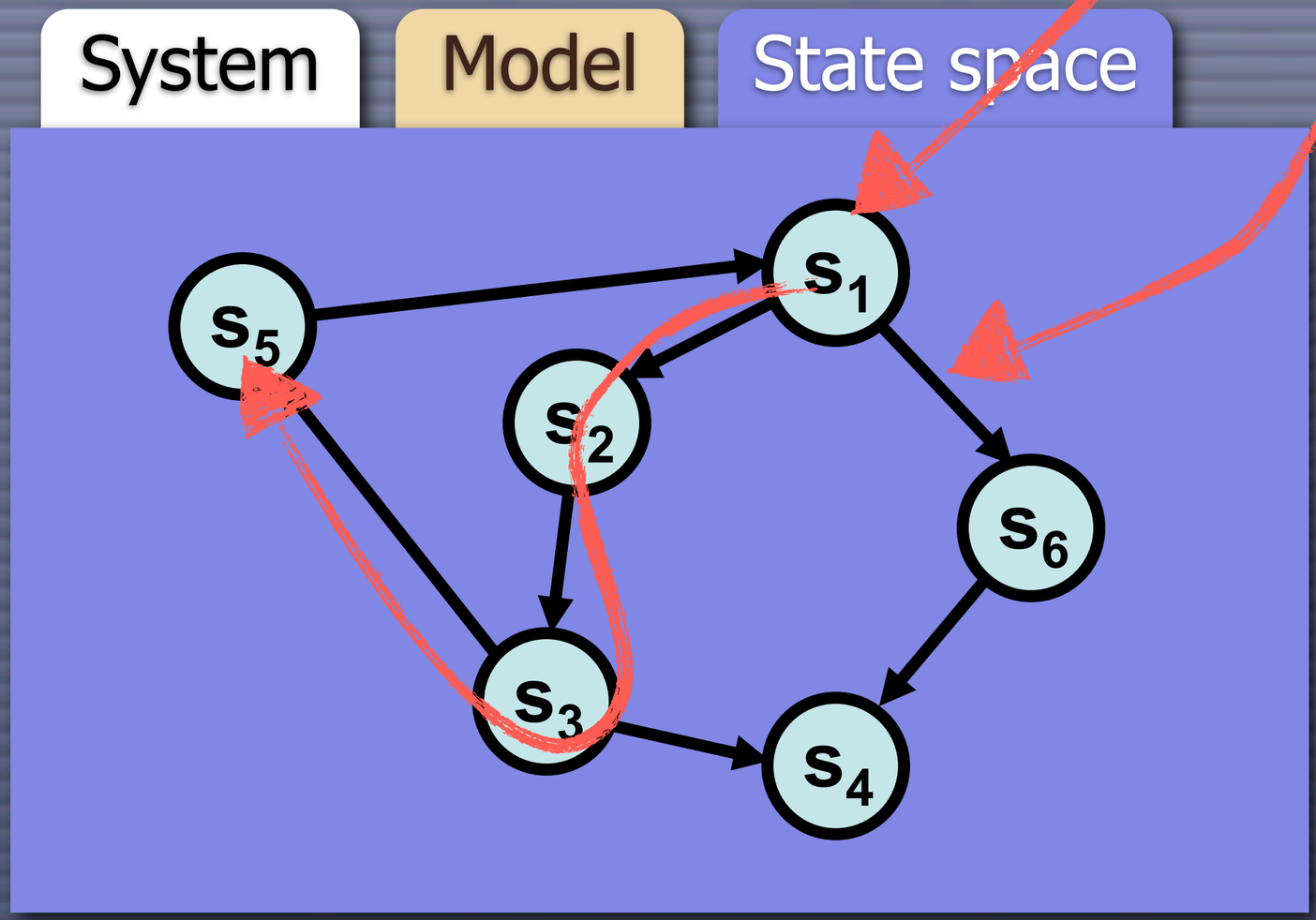
Nodes: Reachable states
Arcs: Transition executions
Paths: Execution sequences

- ✓Highly automatic
- ✓Rich set of behavioural properties
- ✓Diagnostic information

State Space Analysis

State space exploration is a main approach to computer-aided verification:

System | Model | State space



Nodes: Reachable states
Arcs: Transition executions
Paths: Execution sequences

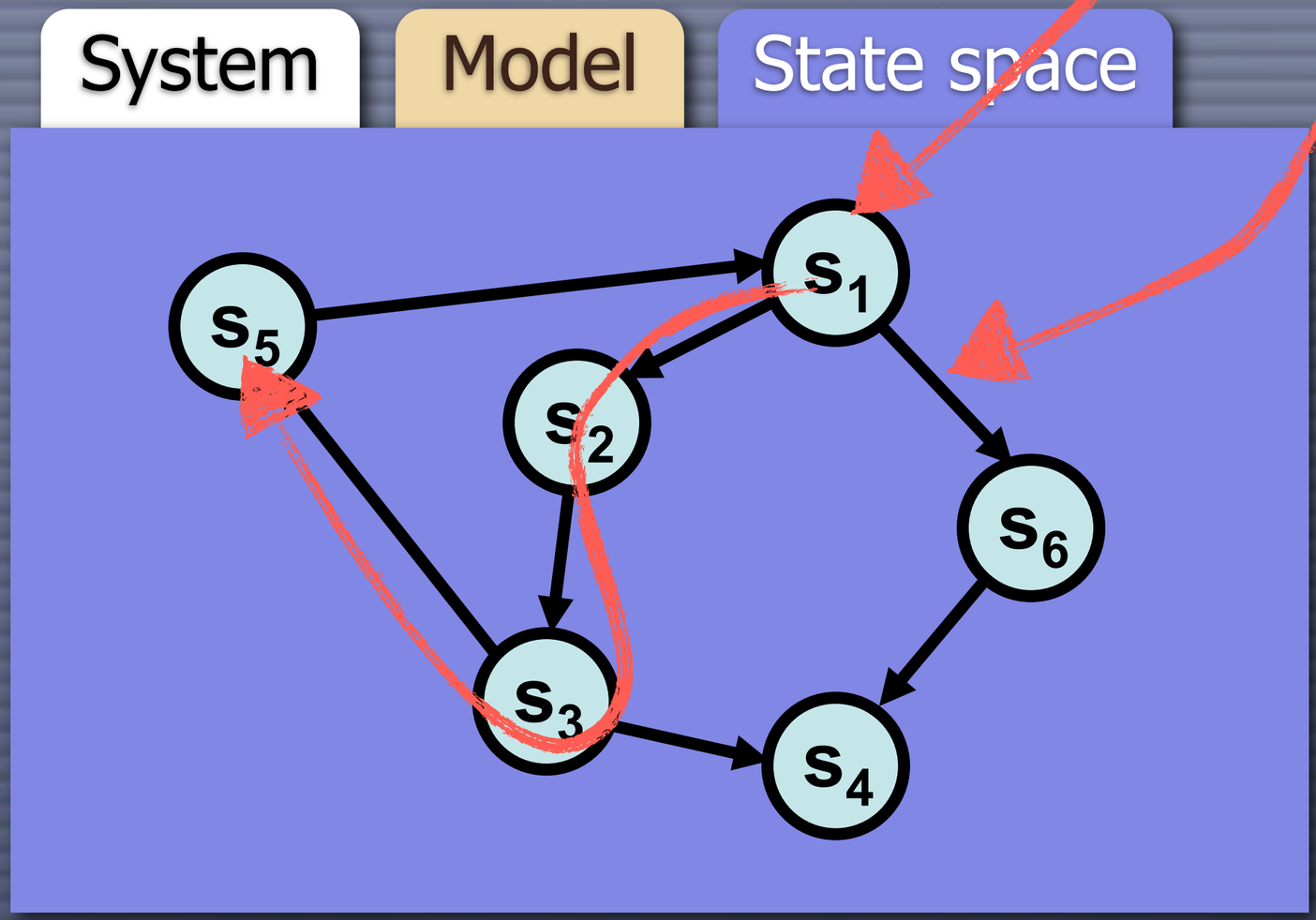
✓Highly automatic
✓Rich set of behavioural properties
✓Diagnostic information

÷State spaces are large

State Space Analysis

State space exploration is a main approach to computer-aided verification:

System | Model | State space

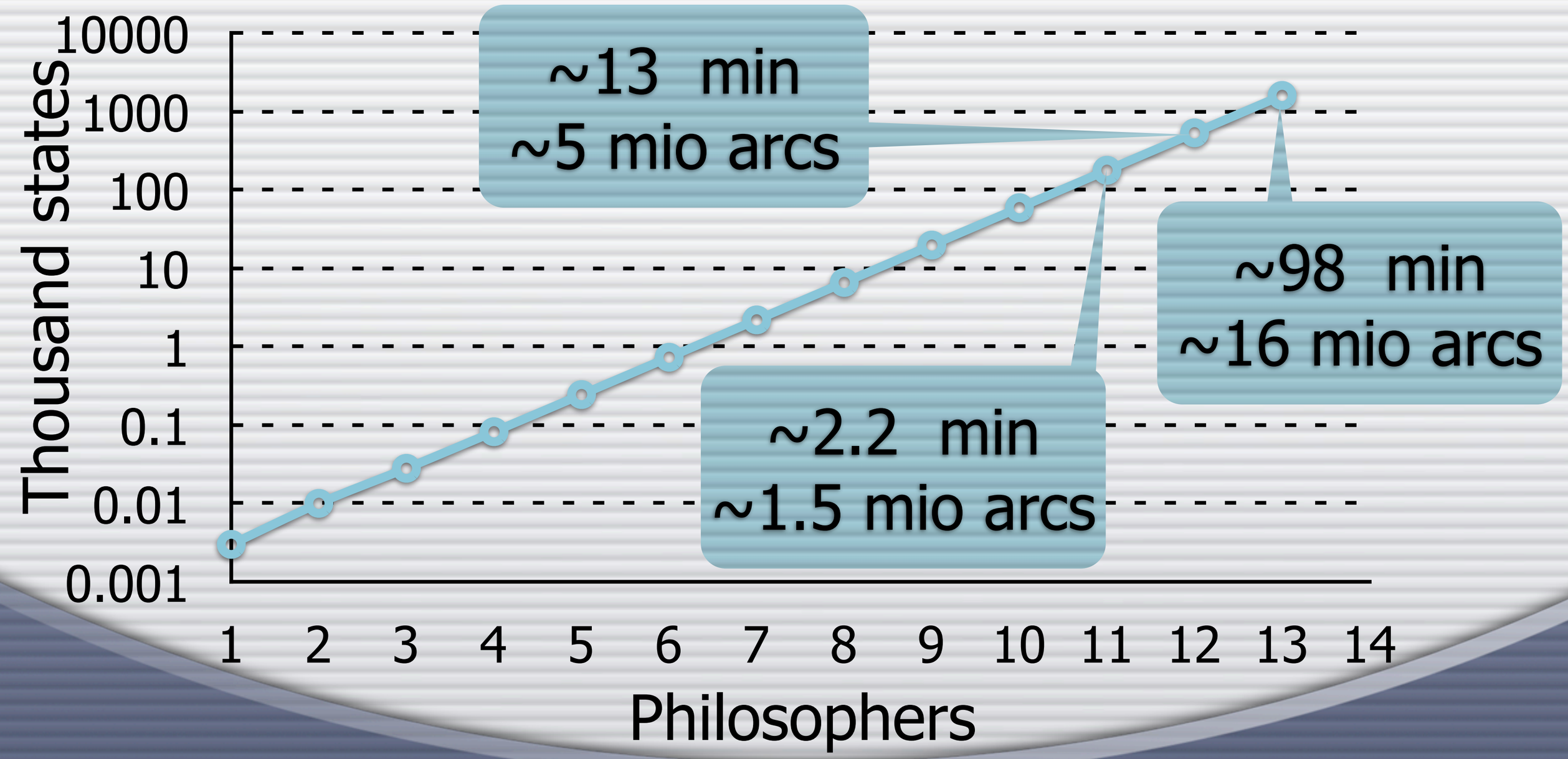


Nodes: Reachable states
Arcs: Transition executions
Paths: Execution sequences

✓Highly automatic
✓Rich set of behavioural properties
✓Diagnostic information

÷State spaces are large

A main research challenge is techniques for alleviating the state explosion problem



The State Explosion Problem

State Space Methods

- Store states compactly
- Delete states during exploration
- Store only some states
- Use external memory

State Space Method Zoo

State Space Method Zoo

Partial order reduction

Symbolic Model Checking

Sweep-line method

μ-calculus

CTL

LTL

State caching

Symmetry reductio

Bit-state hashing

Modular state spaces

Hash compaction

- Exploits system characteristics
- No method works well in general
- Trade-off between memory, time, and expressive power

State Space Method Zoo

Partial order reduction

Symbolic Model Checking

Sweep-line method

μ-calculus

CTL

LTL

State caching

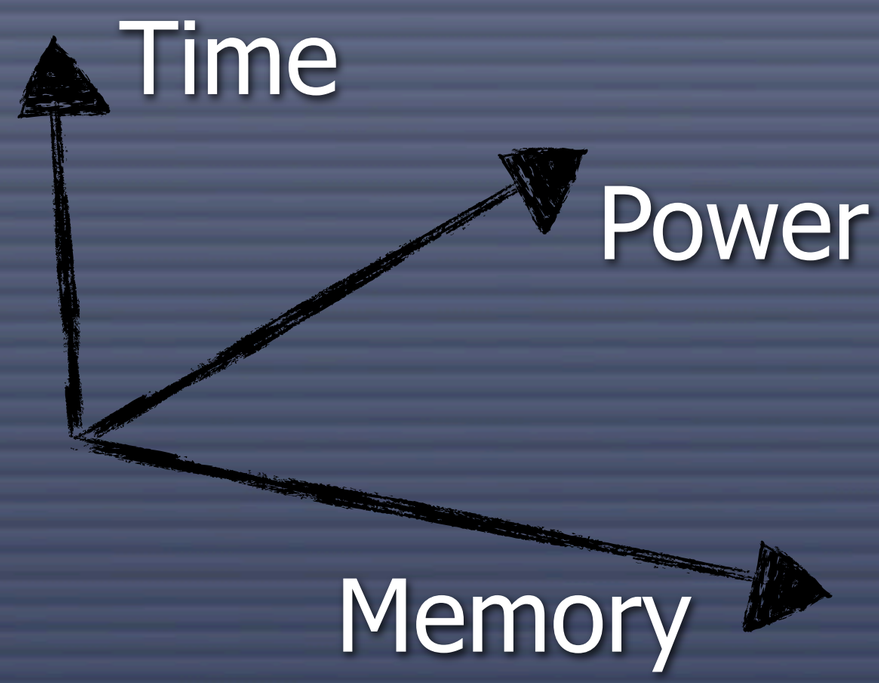
Symmetry reductio

Bit-state hashing

Modular state spaces

Hash compaction

- Exploits system characteristics
- No method works well in general
- Trade-off between memory, time, and expressive power



Observations

- State space methods can now be used to validate industrial-sized practical systems
- Active research: better method are devised all the time
- Implications for computer tools
  - A tool must support a wide range of methods
  - A tool must be extensible

ASAP

- Next generation tool for state space analysis
- ASAP = ASCoVeCo State space Analysis Platform
- ASCoVeCo = Advanced State space Methods and Computer Tools for Verification of Communication Protocols
  - 3 year research project

Aim and Vision

- Support 3 (4) usage scenarios
  - **Research**: Implementation and experiments
  - **Education**: User and implementation perspective
  - **Industrial use**: Ease of use, stability, highly automatic, pragmatic methods and expressiveness
  - (**Facilitator for other tools**: Building blocks that are easy to use in other contexts)
- Challenge: How to facilitate all of this in a single tool

Outline for Rest of Tutorial

- ASAP Practical Use
  - Verification jobs and JoSEL
- Architecture of ASAP
- Simple methods
- **Break**
- Advanced Methods
- Extending ASAP
- Status and Outlook