Toward Architecture-based Reliability Estimation

Roshanak Roshandel & Nenad Medvidovic

Computer Science Department
University of Southern California
{roshande,neno}@usc.edu

Motivation

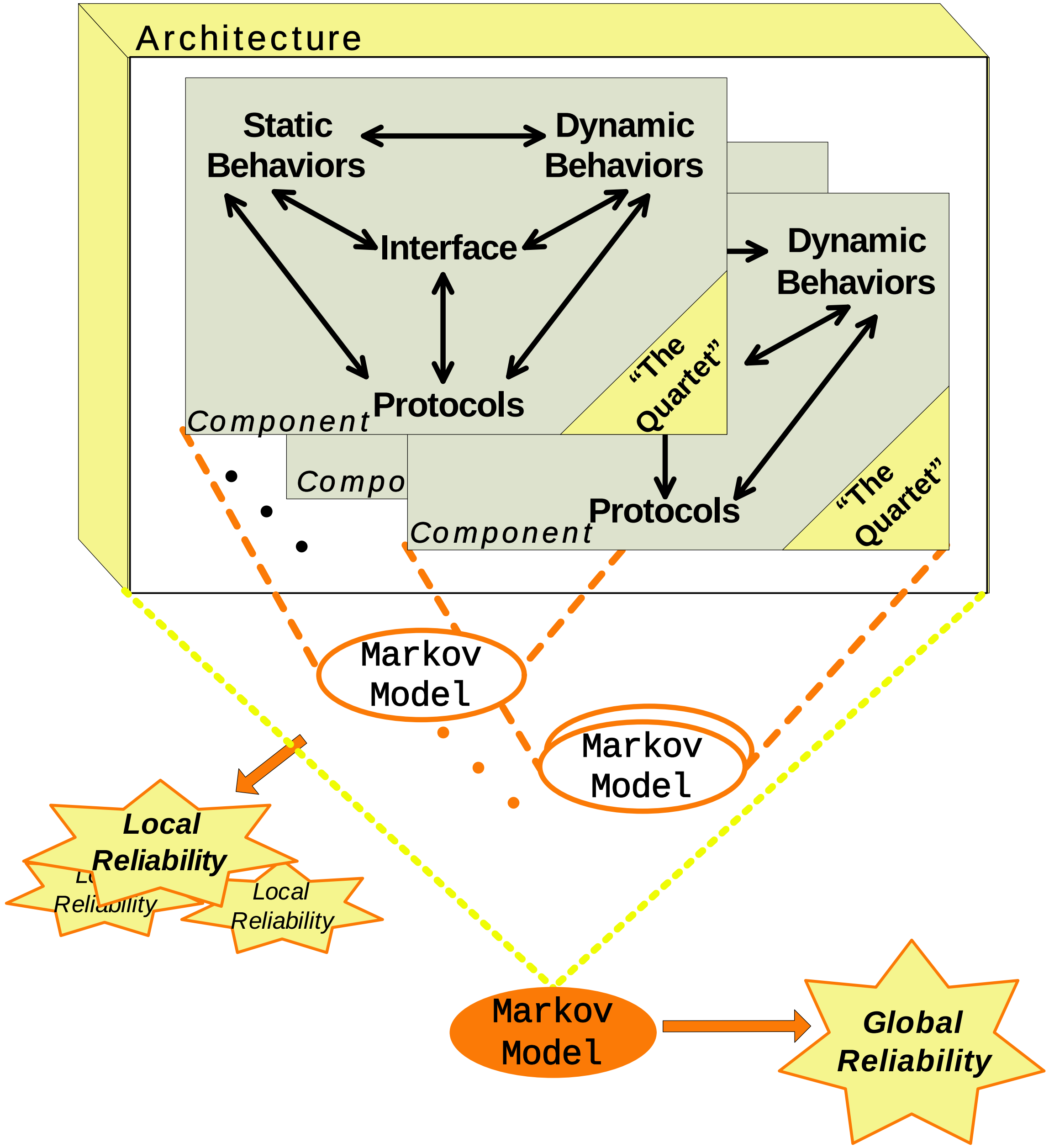
- Software reliability: probability that the system performs its intended functionality without failure
- Software reliability techniques aim at reducing or eliminating failure of software systems
- Complementary to *testing*, rely on implementation
- **How do we go about building reliable systems?**
- **How do we measure reliability early?**

Software Architecture

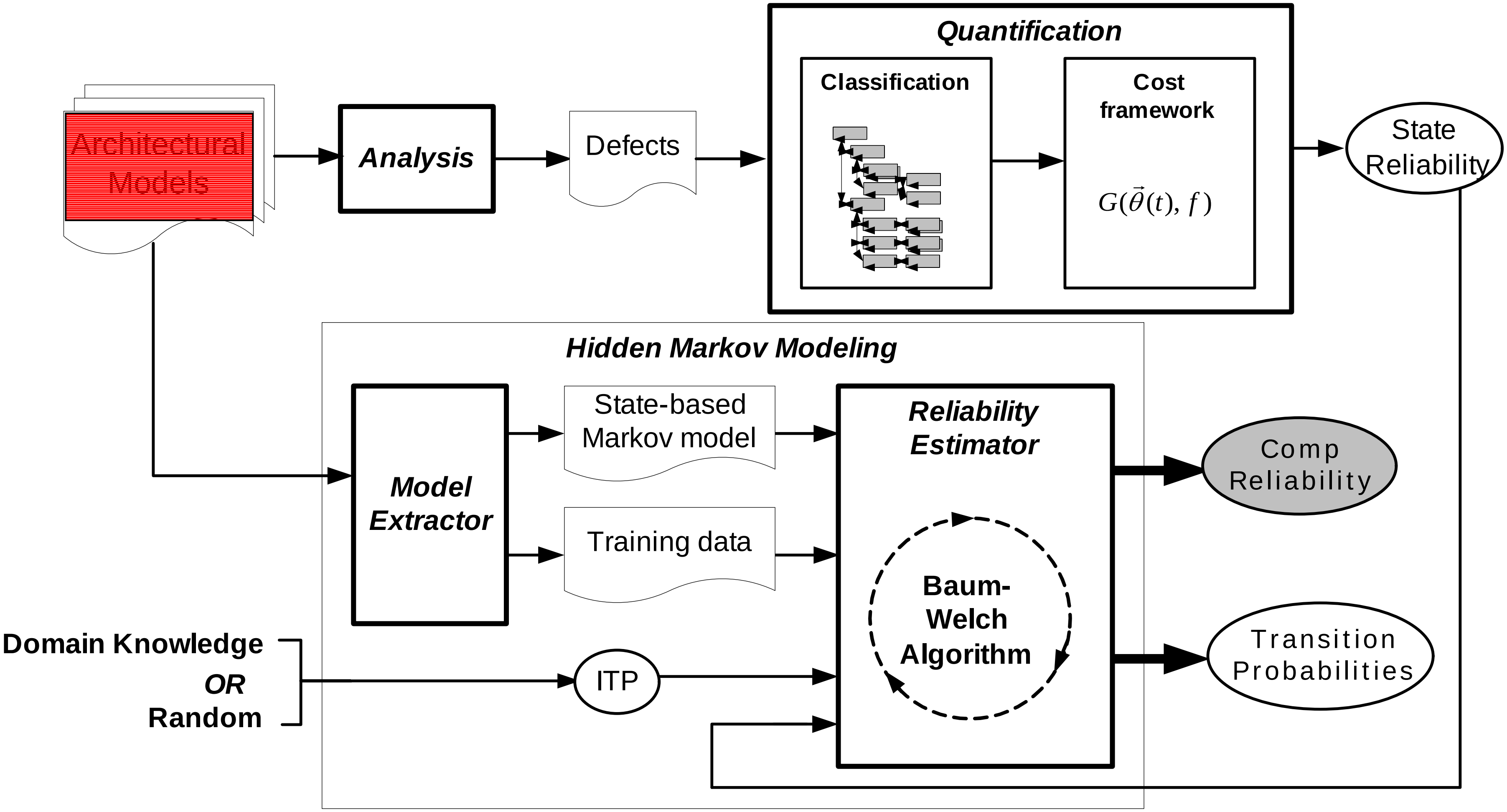
- High-level abstractions describing
  - Structure, Behavior, Constraints
- Coarse-grain building blocks, promote separation of concerns, reuse
  - Components, Connectors, Interfaces, Configurations
- Architectural decisions directly affect aspects of software dependability
  - Reliability
- ADLs, Formal modeling notations, related analysis
  - Often lack *quantification* and *measurement*

Architectural Reliability

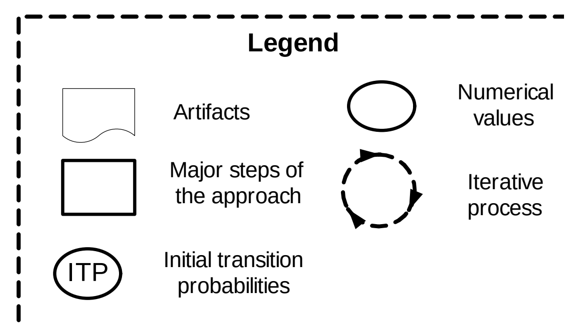
- Lightly explored
- Require availability of implementation to:
  - Build behavioral model of the software system
  - Obtain each component's reliability
- Software architecture offers compositional approaches to modeling and analysis
- The challenge is *quantifying* these results
  - Presence of uncertainty
  - Unknown operational profile
  - Improper behavior



Architecture
Static Behaviors
Dynamic Behaviors
Interface
Protocols
Component
"The Quartet"
Compo
Dynamic Behaviors
Protocols
Component
"The Quartet"
Markov Model
Markov Model
Local Reliability
Reliability
Local Reliability
Markov Model
Global Reliability

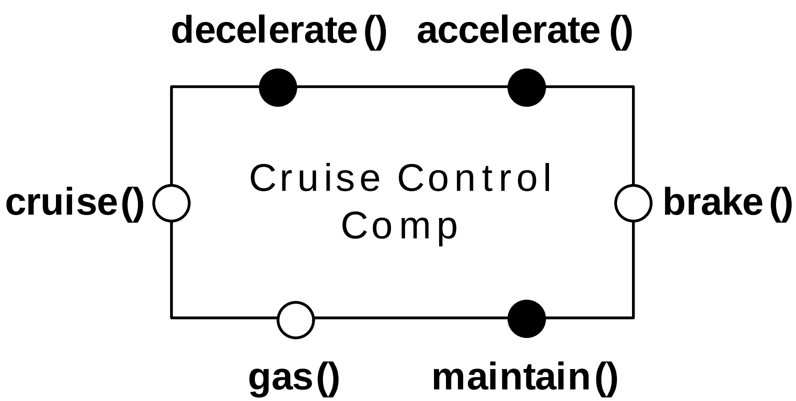


Component Reliability

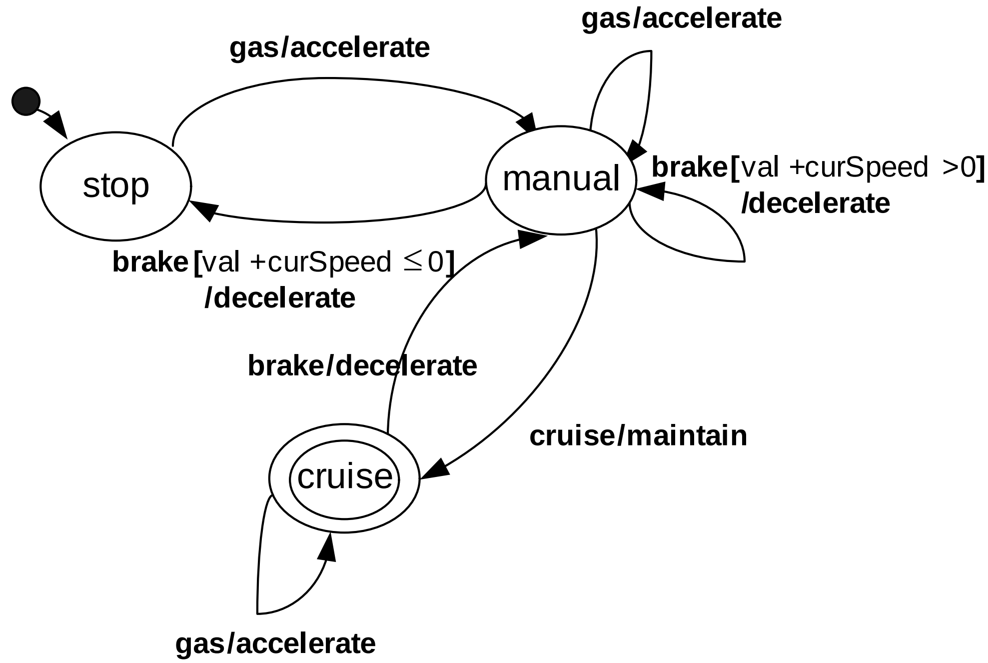


The *Quartet*

1. *Interface*
    - Point by which a component interacts with other components
2. *Static behavior*
    - Discrete functionality of a component
    - i.e., at particular "snapshots" during the system's execution
3. *Dynamic behavior*
    - Continuous view of *how* a component arrives at different states throughout its execution
4. *Interaction protocol*
    - E*xternal* view of the component
    - Specifies its legal interactions with other components in the system



DYNAMIC BEHAVIOR



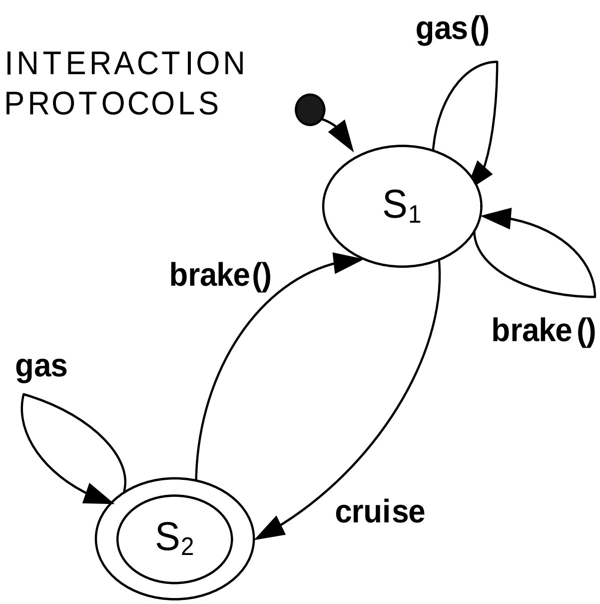
INTERFACES

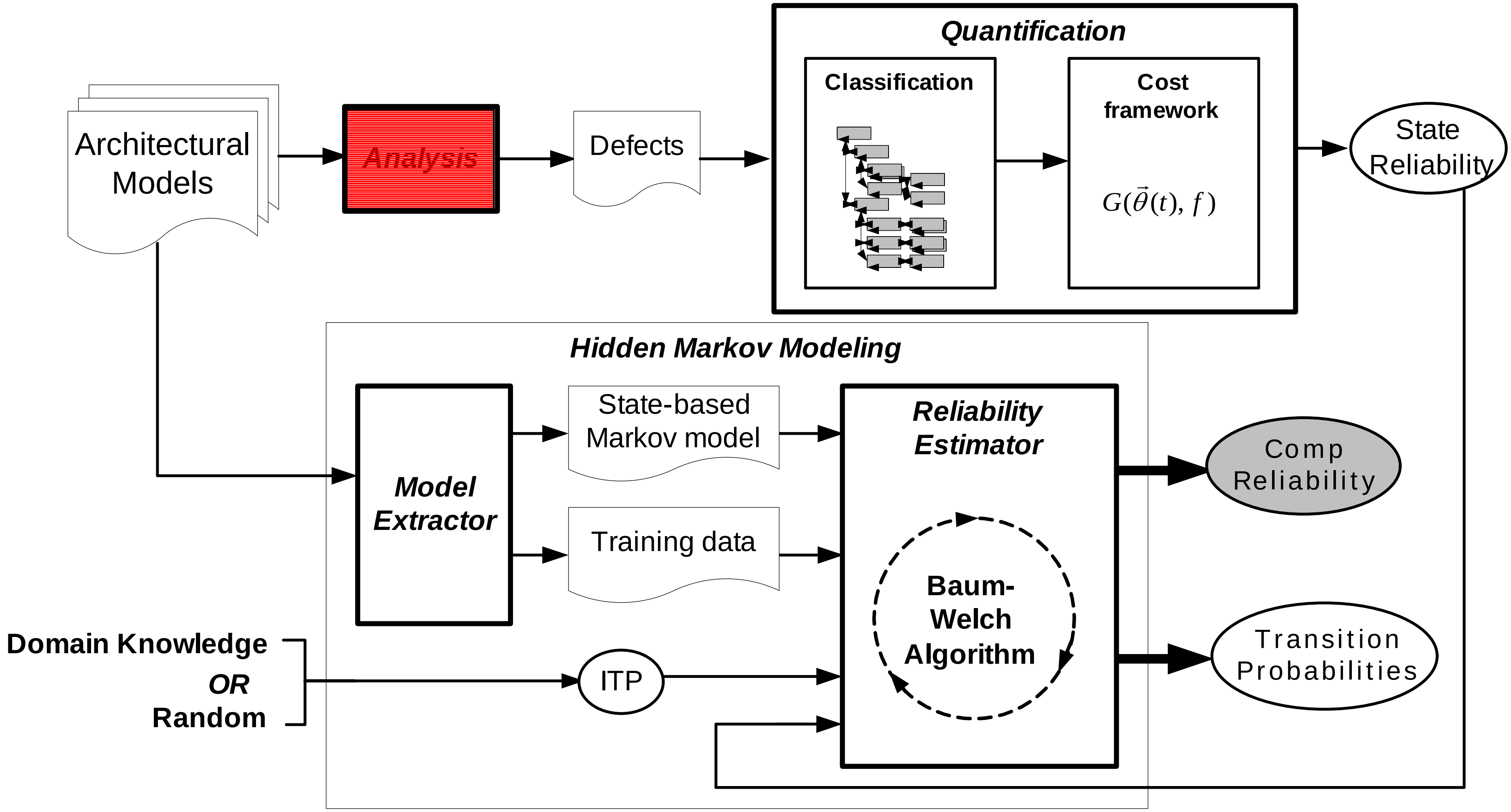
```
PROV gas(val:SpeedType):SpeedType;
PROV brake(val:SpeedType):SpeedType;
PROV cruise(speed:SpeedType);Boolean;
```

STATIC BEHAVIOR

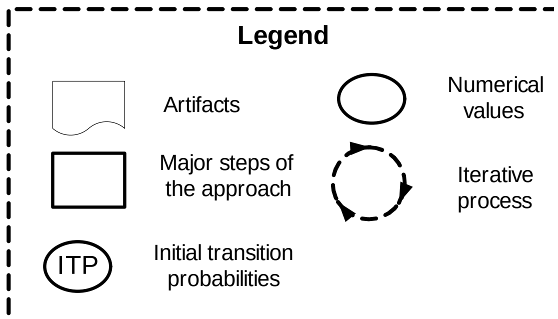
```
STATE-VAR:
   curSpeed:SpeedType;
   isCruising:Boolean;
INVARIANT:
   0 ≤ curSpeed ≤ MAX;
OPERATIONS:
   gas.preCond (val > 0);
   gas.postCond (~curSpeed = curSpeed + val);
   brake.preCond (val < 0);
   brake.postCond (~curSpeed = curSpeed + val
                  AND isCruising = false);
   cruise.preCond (speed > 0);
   cruise.postCond (~curSpeed = speed
                   AND isCruising = true);
```

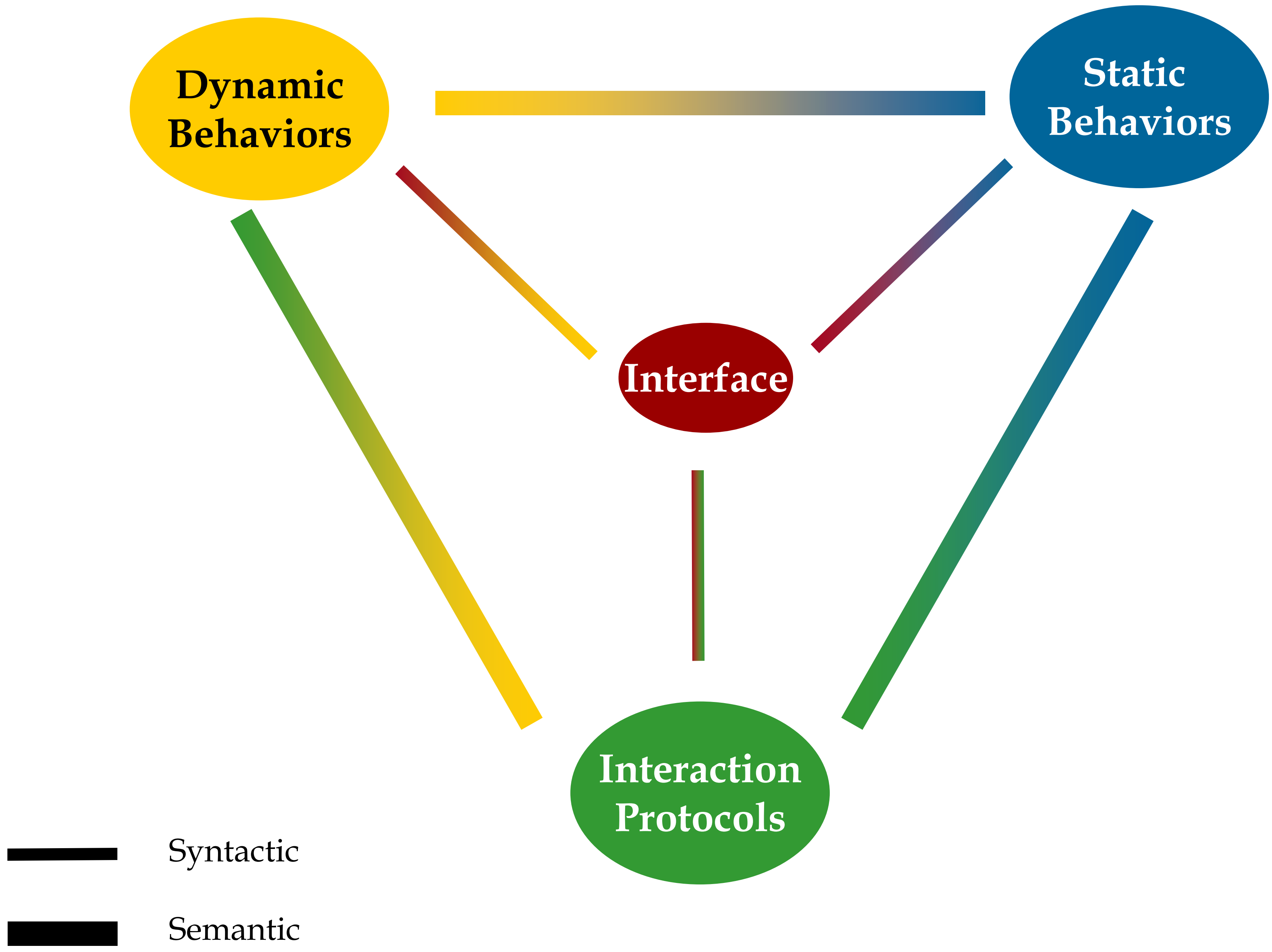
INTERACTION
PROTOCOLS

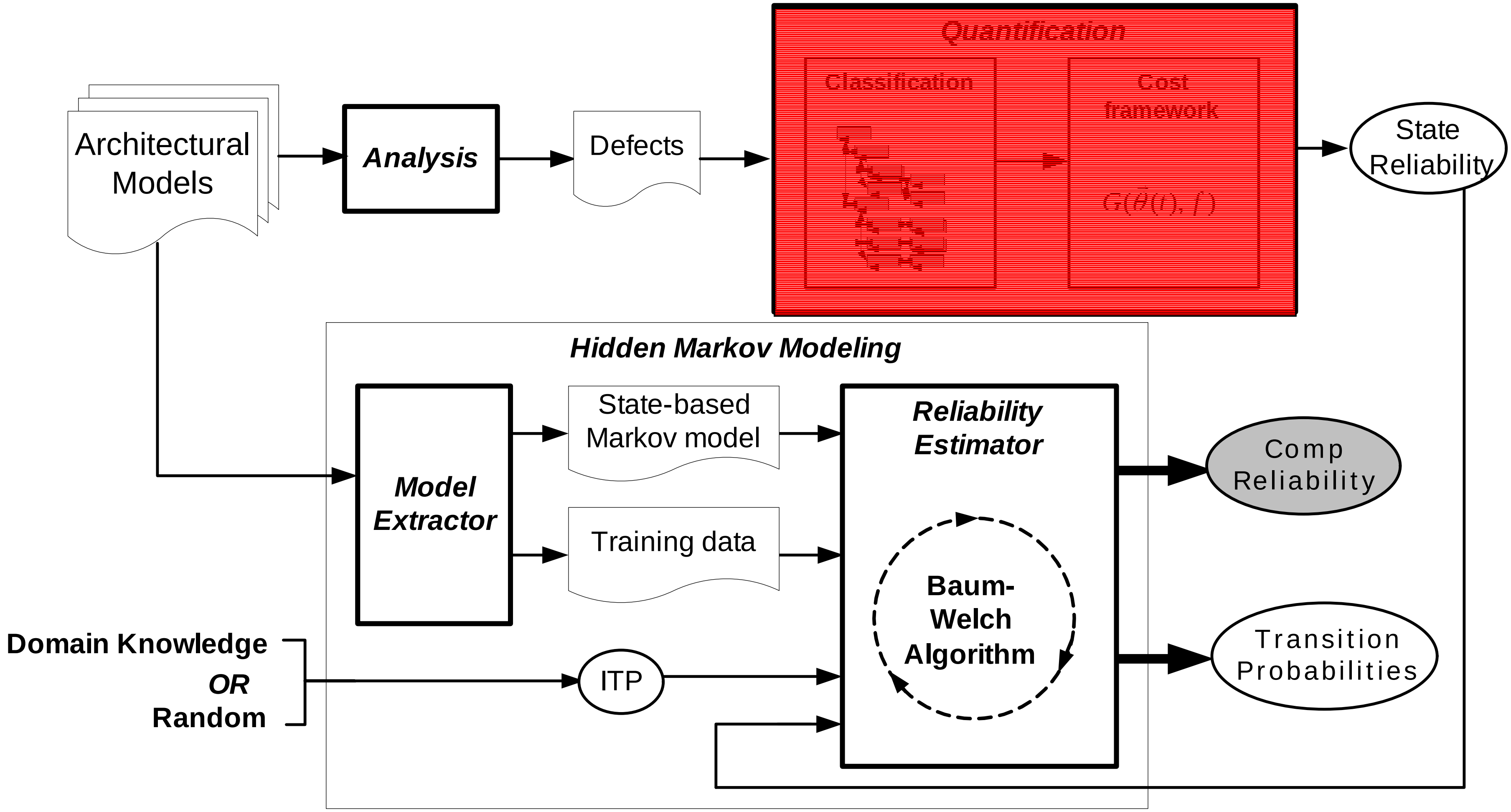




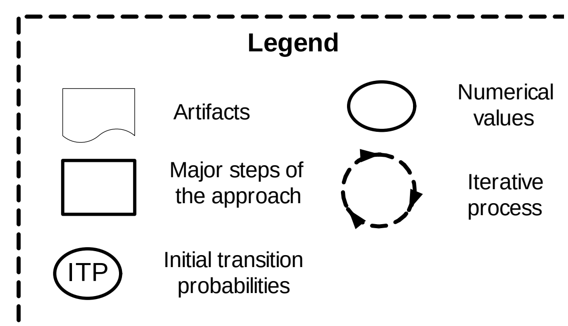
Component Reliability




Dynamic Behaviors
Static Behaviors
Interface
Interaction Protocols
Syntactic
Semantic



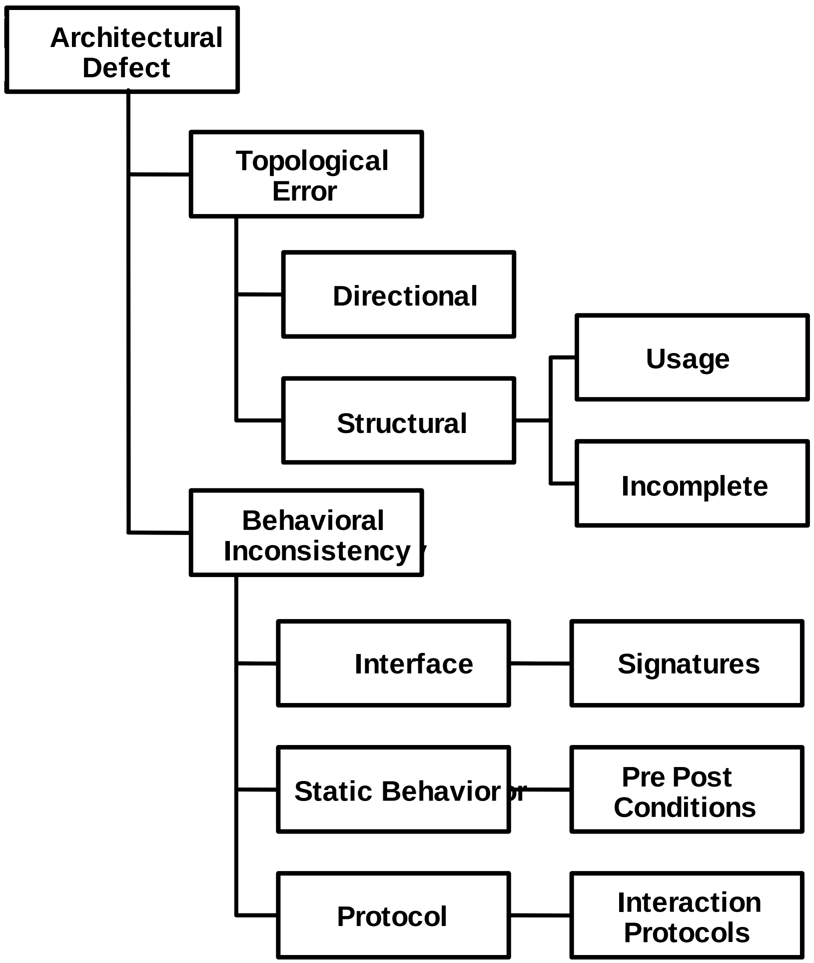
Component Reliability



Defect Quantification

- Architectural defects could affect system Reliability
- Different defects affect the Reliability differently
  - e.g., interface mismatch vs. protocol mismatch
- The cost of mitigating defects varies based on the defect type
- Other (domain specific) factors may affect the quantification
- **Classification + Cost framework**

Classification + Cost Framework

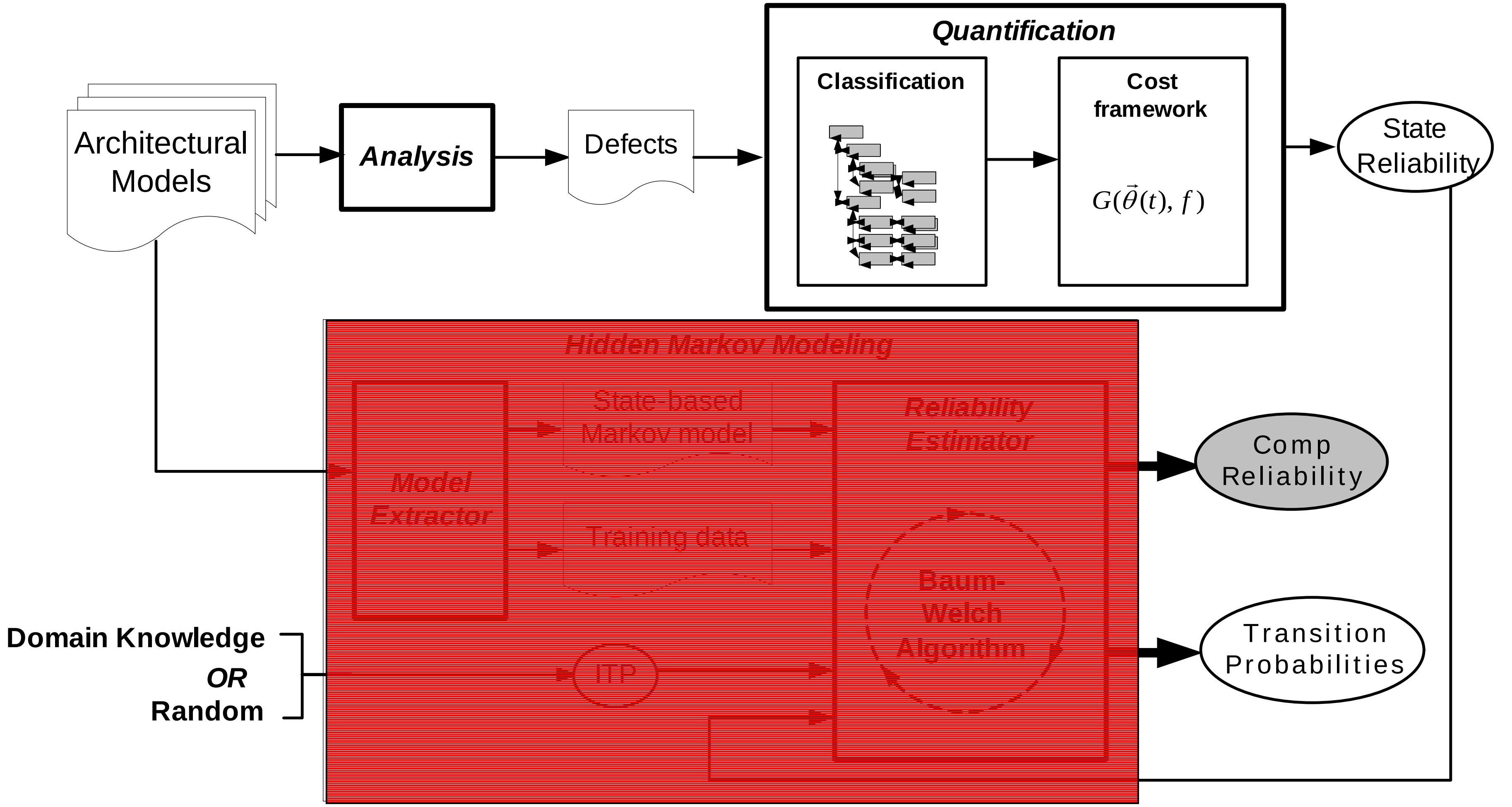


- Pluggable/Adaptable
- Identify the important factors within a domain
- For a defect class *t*

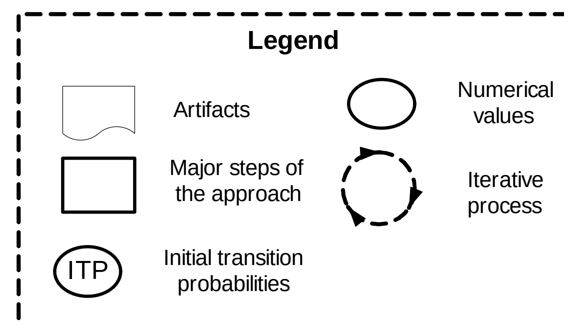
$$c_t = G(\vec{\theta}(t), f), \ where$$

$$\vec{\theta}(t) = [\theta_1(t), \theta_2(t), \ldots, \theta_n(t)]$$

- *f*: Frequency of occurrence
- And $\vec{\theta}(t)$ vector of all relevant factors
- Result will be used in reliability estimation



Component Reliability



Reliability Techniques

- Non-Homogenous Poisson Processes, Binomial Models, Software Reliability Growth Models, …
- Markovian Models
  - Suited to architectural approaches
  - Consider a system's structure, compositional
  - Stochastic processes
  - Informally, a finite state machine extended with transition probabilities

Our Reliability Model

- Built based on the *dynamic behavioral model*
- Assume Markov property
  - Discrete Time Markov Chains
- Transition probabilities may be unknown
- Complex behavior results in lack of a correspondence between events and states
- Event/action pairs to describe component interactions

➔**Augmented Hidden Markov Models (AHMM)**

Evaluation

- Uncertainty analysis
  - Operational profile
  - Incorrect behavior
- Sensitivity analysis
  - Traditional Markov-based sensitivity analysis combined with the defect quantification
- Complexity
- Scalability

Conclusion and Future Work

- Step toward closing the gap between architectural specification and its effect on system's reliability
- Handles two types of uncertainties associated with early reliability estimation
- Preliminary results are promising
- Need further evaluation
- Build compositional models to estimate system reliability based on estimated component reliabilities

Questions?

AHMM

$S: Set\ of\ all\ possible\ States, S = \{S_1, ..., S_N\}$

$N: Number\ of\ states$

$q_t: state\ at\ time\ t$

$E: Set\ of\ all\ events,\ E = \{E_1, ..., E_M\}$

$M: Number\ of\ events$

$F: Set\ of\ all\ actions,\ F: \{F_1, ..., F_K\}$

$K: Number\ of\ actions$

$We\ now\ define:$

$\lambda = (A, B, \pi)\ is\ a\ Hidden\ Markov\ Model\ such\ that:$

$A: state\ transition\ probability\ distribution$

$A = \{a_{ij}\}, a_{ij} = \Pr[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N$

$B: Interface\ probability\ distribution\ in\ state\ j$

$B = \{b_j(m)\}$

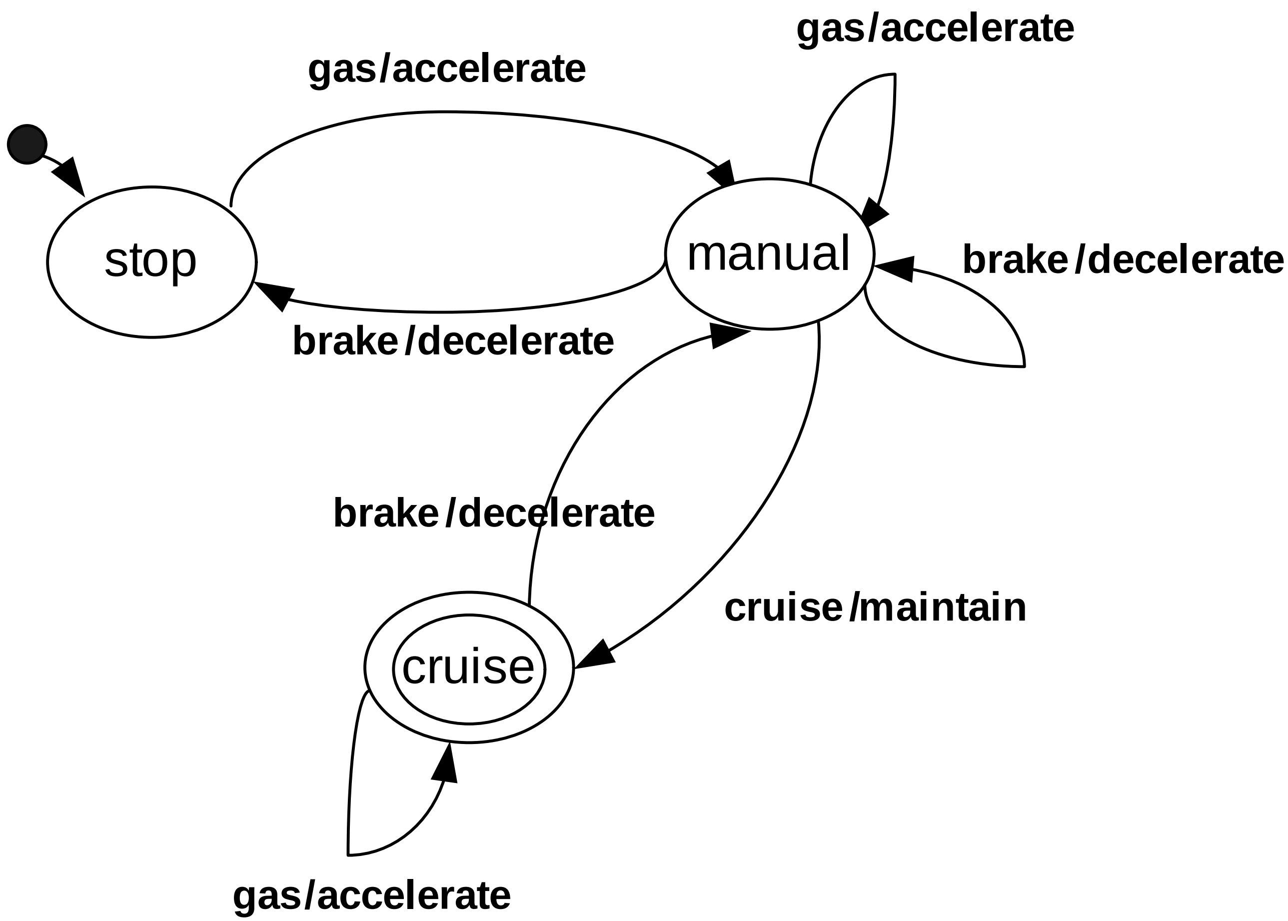
$b_j(m) = \Pr[E_m / F_k\ at\ t | q_t = S_j], 1 \leq j \leq N, 1 \leq m \leq M, 1 \leq k \leq K$

$\pi: The\ initial\ probability\ distribution\ \pi = \{\pi_i\}$

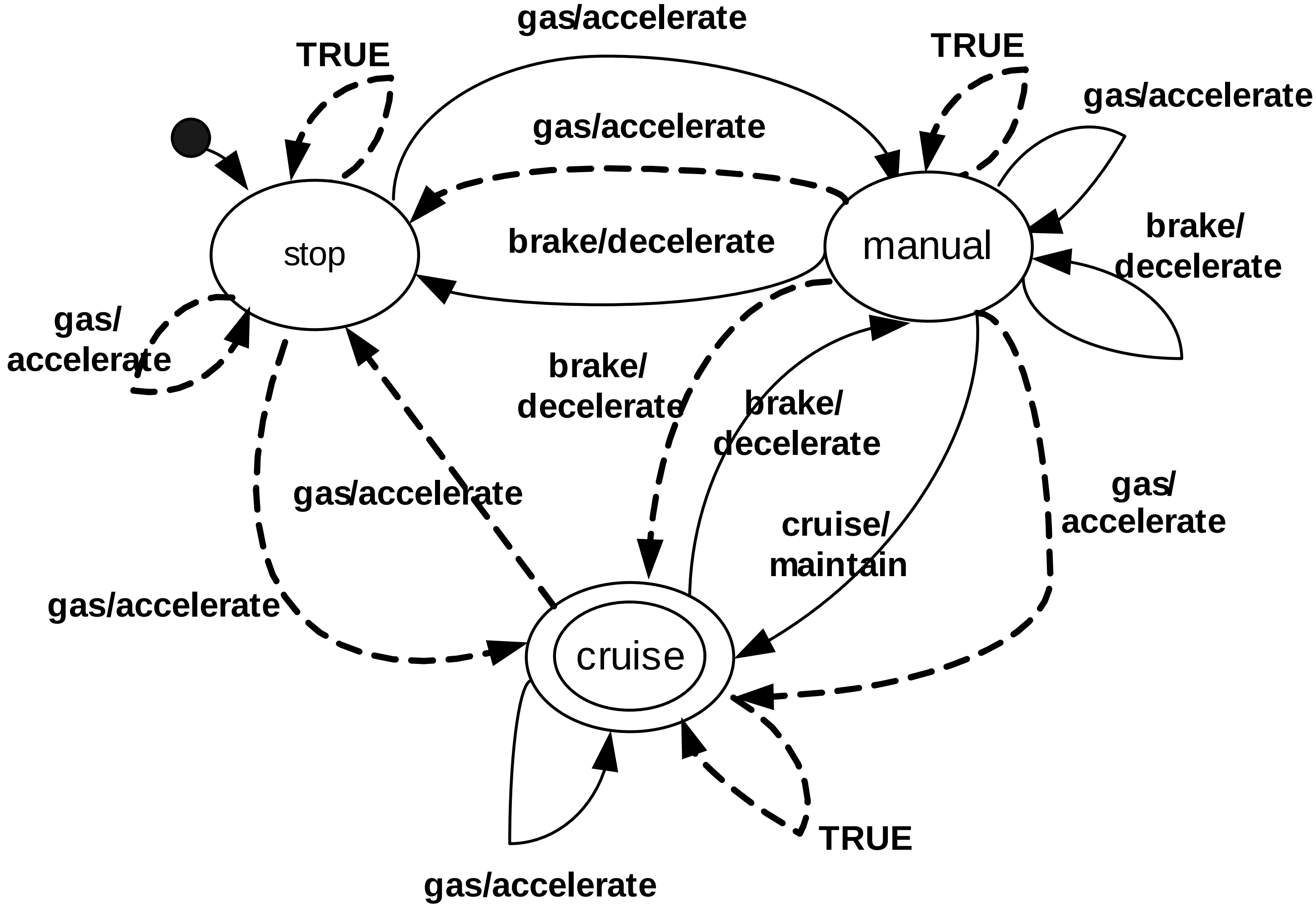
$\pi_i = \Pr[q_1 = S_i], 1 \leq i \leq n.$

Cruise Control Example

DYNAMIC BEHAVIOR



Partial Markov Extension



Transition Probabilities

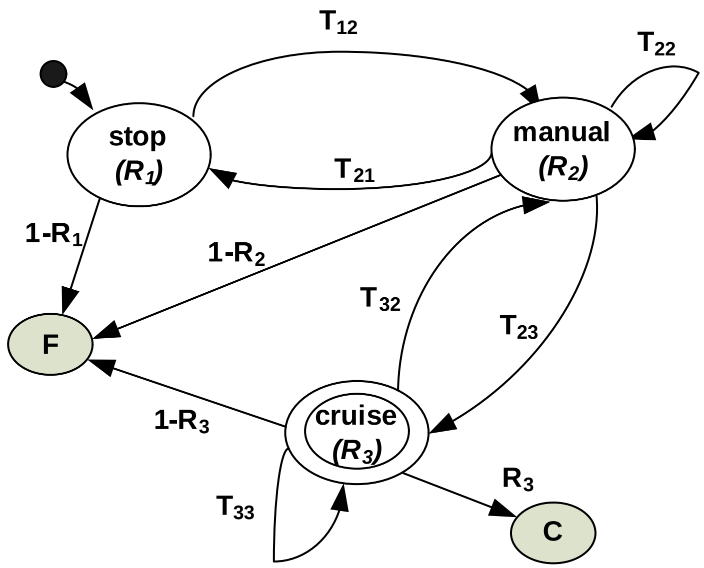
Origin State	Observation		Pr(O)	Reaction	Pr(R)	Total Pr Pr(O).Pr(R)	Dest. State
stop	TRUE	0.1	0.1	TRUE	1	0.1	***stop***
stop	gas		0.05	accelerate	1	0.05	***stop***
stop	gas	0.9	0.05	accelerate	1	0.05	***cruise***
stop	gas		0.8	accelerate	1	0.8	***manual***
cruise	break	0.8 5	0.85	decelerate	1	0.85	***manual***
cruise	TRUE	0.1	0.1	TRUE	1	0.1	***cruise***
cruise	gas	0.0 5	0.02	accelerate	1	0.02	***stop***
cruise	gas		0.03	accelerate	1	0.03	***cruise***
manual	TRUE	0.2	0.2	TRUE	1	0.2	***manual***
manual	gas		0.08	accelerate	1	0.08	***manual***
manual	gas	0.1	0.02	accelerate	0.6	0.012	***cruise***
manual	gas			accelerate	0.4	0.008	***stop***
manual	break		0.08	decelerate	1	0.08	***manual***
manual	break	0.1	0.01	decelerate	1	0.01	***cruise***
manual	break		0.01	decelerate	1	0.01	***stop***
manual	cruise	0.6	0.6	maintain	1	0.6	***cruise***

$$
\begin{array}{r} \\ ITP = \end{array}
\begin{array}{r c}
 & \begin{array}{ccc} stop & manual & cruise \end{array} \\
\begin{array}{r} stop \\ manual \\ cruise \end{array} & \begin{bmatrix} 0.15 & 0.8 & 0.05 \\ 0.018 & 0.36 & 0.622 \\ 0.02 & 0.85 & 0.13 \end{bmatrix}
\end{array}
$$

Baum-Welch

$$
P = \begin{bmatrix} 0.1178 & 0.8293 & 0.0529 \\ 0.0304 & 0.3672 & 0.6024 \\ 0.0135 & 0.8537 & 0.1328 \end{bmatrix}
$$

Reliability Model



- **Adaptation of Cheung1980**

$\hat{P}^n(i, j)$ **Probability of reaching *j* from *i* after *n* steps.**

$$\hat{P} = \begin{array}{c} \\ C \\ F \\ S_1 \\ \dots \\ S_i \\ \dots \\ S_{N-1} \\ S_N \end{array} \begin{array}{c} \begin{array}{cccccccc} C & F & S_1 & S_2 & \dots & S_j & \dots & S_N \end{array} \\ \left[\begin{array}{cccccccc} 1 & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1-R_1 & R_1T_{11} & R_1T_{12} & \dots & R_1T_{1j} & \dots & R_1T_{1N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1-R_i & R_iT_{i1} & R_iT_{i2} & \dots & R_iT_{ij} & \dots & R_iT_{iN} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1-R_{N-1} & R_{N-1}T_{(N-1)1} & R_{N-1}T_{(N-1)2} & \dots & R_{N-1}T_{(N-1)j} & \dots & R_{N-1}T_{(N-1)N} \\ R_N & 1-R_N & R_NT_{N1} & R_NT_{N2} & \dots & R_NT_{Nj} & \dots & R_NT_{NN} \end{array} \right] \end{array}$$

$$R_{comp} = \hat{P}^n(S_1, C)$$

Example…

$$ITP = \begin{array}{c} \\ stop \\ manual \\ cruise \end{array} \begin{array}{c} \begin{array}{ccc} stop & manual & cruise \end{array} \\ \begin{bmatrix} 0.15 & 0.8 & 0.05 \\ 0.018 & 0.36 & 0.622 \\ 0.02 & 0.85 & 0.13 \end{bmatrix} \end{array}$$

$$P = \begin{bmatrix} 0.1178 & 0.8293 & 0.0529 \\ 0.0304 & 0.3672 & 0.6024 \\ 0.0135 & 0.8537 & 0.1328 \end{bmatrix}$$

R_{stop}=0.87, R_{manual}=0.9, R_{cruise}=0.76

$$\hat{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0.1300 & 0.0670 & 0.7444 & 0.0864 \\ 0 & 1 & 0.0147 & 0.3626 & 0.5227 \\ 0 & 0.2400 & 0 & 0 & 0 \end{bmatrix}$$

Q

$$R_{comp} = Q^{-1}(1, cruise) \times R_{cruise}$$

$$R_{comp} = 0.7444 \times 0.76$$

$$\approx 0.5657$$

$$\Rightarrow R_{comp} \approx \%56$$

More on the AHMM

- For states S_i and S_j, there may be several transitions E_m/F_k
- Probability of transition from S_i to S_j by means of a given E_m and all possible actions F_k

$$T_{ij} = \sum_{m=1}^{M} \sum_{k=1}^{K} P_{ijE_mF_k}$$

- But do we know what these are at the architecture level?

Parameter (re)estimation

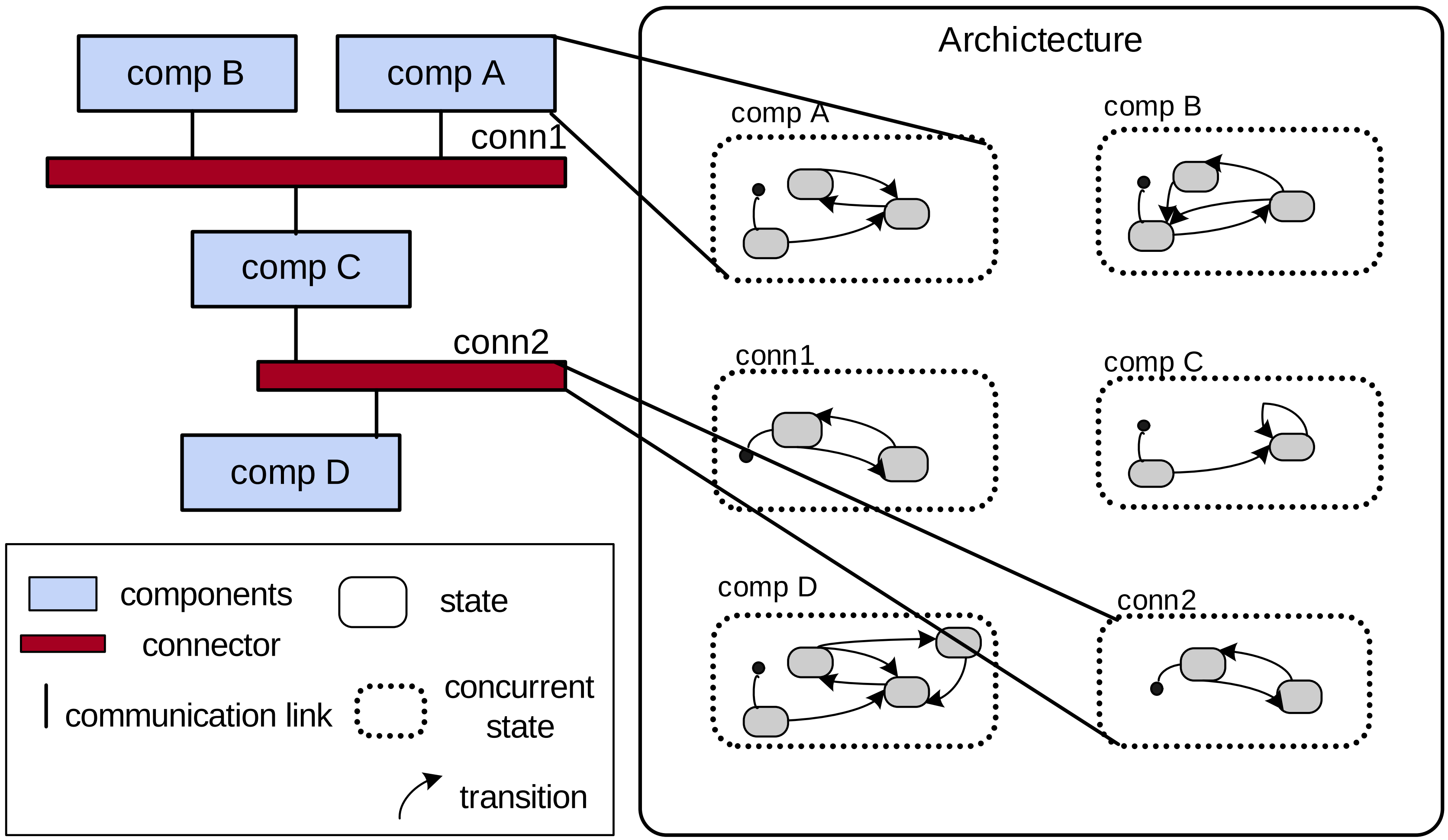
- Baum-Welch algorithm
  - Uses Expectation Maximization

$$\alpha_t(i) = \sum_j \alpha_{t-1}(j)\,\Pr_1(q_t = i \mid q_{t-1} = j)\,\Pr_0(x_t \mid q_t = i)$$

$$\beta_{t-1}(i) = \sum_j \Pr_1(q_t = j \mid q_{t-1} = i)\,\Pr_0(x_t \mid q_t = j)\,\beta_t(j)$$

  - Given a sequence of training data
    - Calculates the probability of a given observation sequence and the probability of transitions from S_i to S_j

System Reliability



Relationships

- **Interface vs. Other Models**
  - Syntactic
  - Interface as the *core*
  - *Static Behaviors* constrain interfaces using pre/post-conditions
  - Transition labels on *Dynamic Behaviors* and *Interaction Protocols* relate to interface as well
  - Dynamic Behaviors and Interaction Protocol model may have additional transitions that do not relate to component's interfaces
    - hierarchy and abstraction

Relationships II

- **Static Behaviors vs. Dynamic Behaviors**
  - Semantic
  - Transition Guard vs. Operation Pre-Condition
    - Union Guard:

$$UG = \bigvee_{i=1}^{n} G_i$$

$$UG \Rightarrow P$$

  - State Invariant vs. Component Invariant

StateInv => CompInv

  - State Invariants vs. Operation Post-Condition

StateInv => PostCond

Relationships III

- **Dynamic Behaviors vs. Interaction Protocols**
  - Semantic
  - The dynamic behavioral model may be more general than the protocol of interactions; any execution trace obtained by the protocol model, must result in a legal execution of component's dynamic behavioral model
- **Static Behaviors vs. Interaction Protocols**
  - Static Behaviors ←→ *Dynamic Behaviors* ←→ Interaction Protocols
  - Dynamic Behavioral model acts as a conceptual bridge
  - Interaction protocols specifies the valid sequence by which the component's interfaces may be accessed, oblivious to the component's internal state
    - No direct conceptual relationship

Uncertainty Analysis

- Two sources of uncertainty:
  - Unknown operation profile, and incorrect component behavior
- How important it is to estimate ITP accurately?
  - Complexity of the behavioral model directly relates to the importance of correct ITP initialization
- How about slight changes to ITP? How well the model can handle uncertainty?

Evaluation

- Uncertainty analysis
  - Operational profile
  - Incorrect behavior
- Sensitivity analysis
  - Traditional Markov-based sensitivity analysis combined with the defect quantification
- Complexity
- Scalability

Uncertainty Analysis

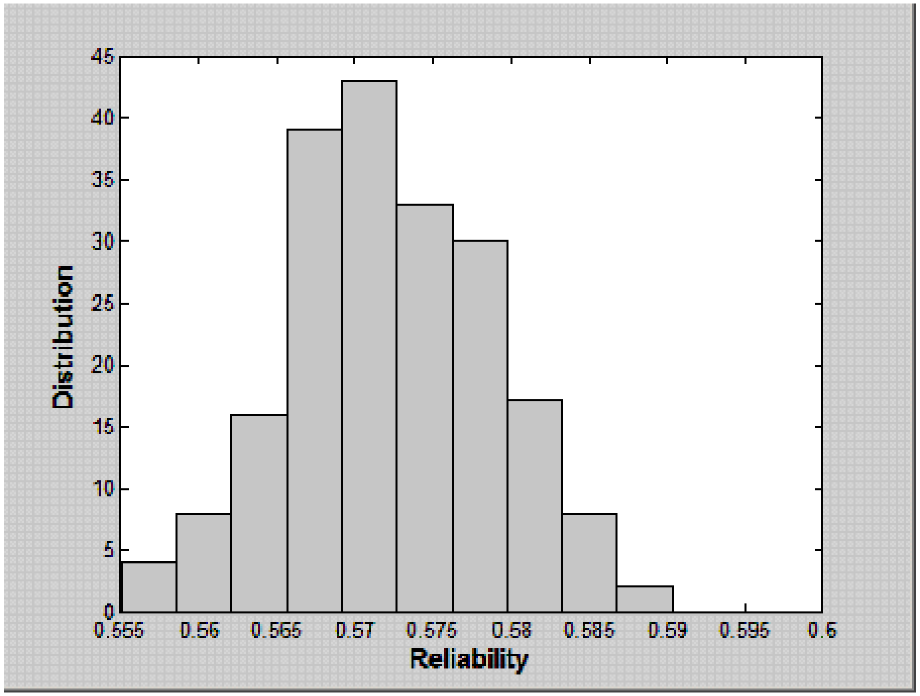
- Two sources of uncertainty:
  - Unknown operation profile, and incorrect component behavior
- How important it is to estimate ITP accurately?
  - Complexity of the behavioral model directly relates to the importance of correct ITP initialization
- How about slight changes to ITP? How well the model can handle uncertainty?

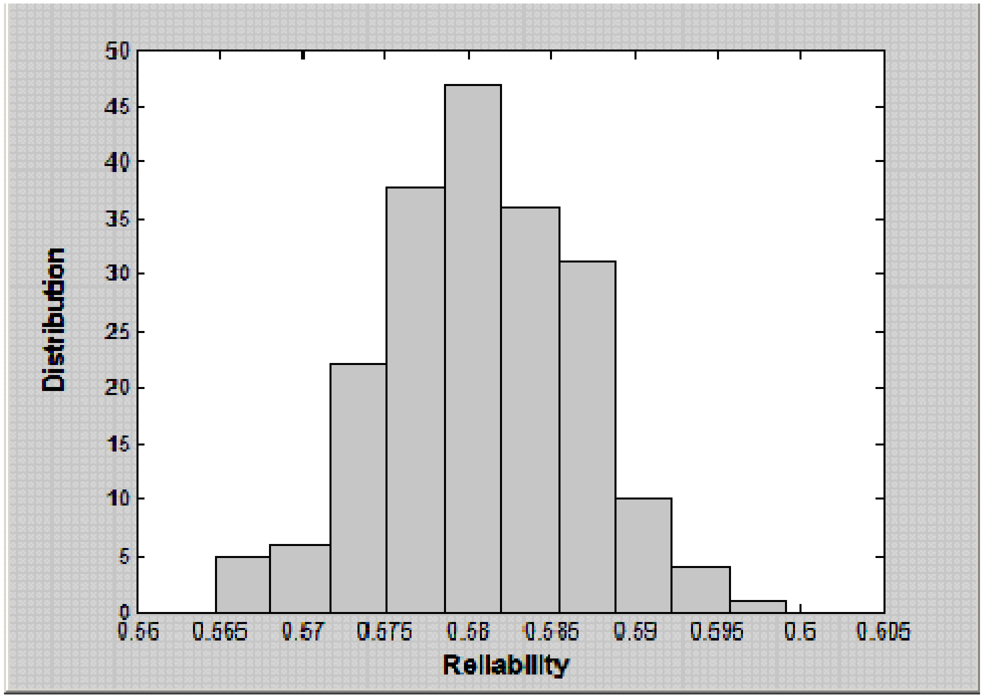
Example

$$ITP = \begin{bmatrix} 0.15 & 0.8 & 0.05 \\ 0.018 & 0.36 & 0.622 \\ 0.02 & 0.85 & 0.13 \end{bmatrix}$$

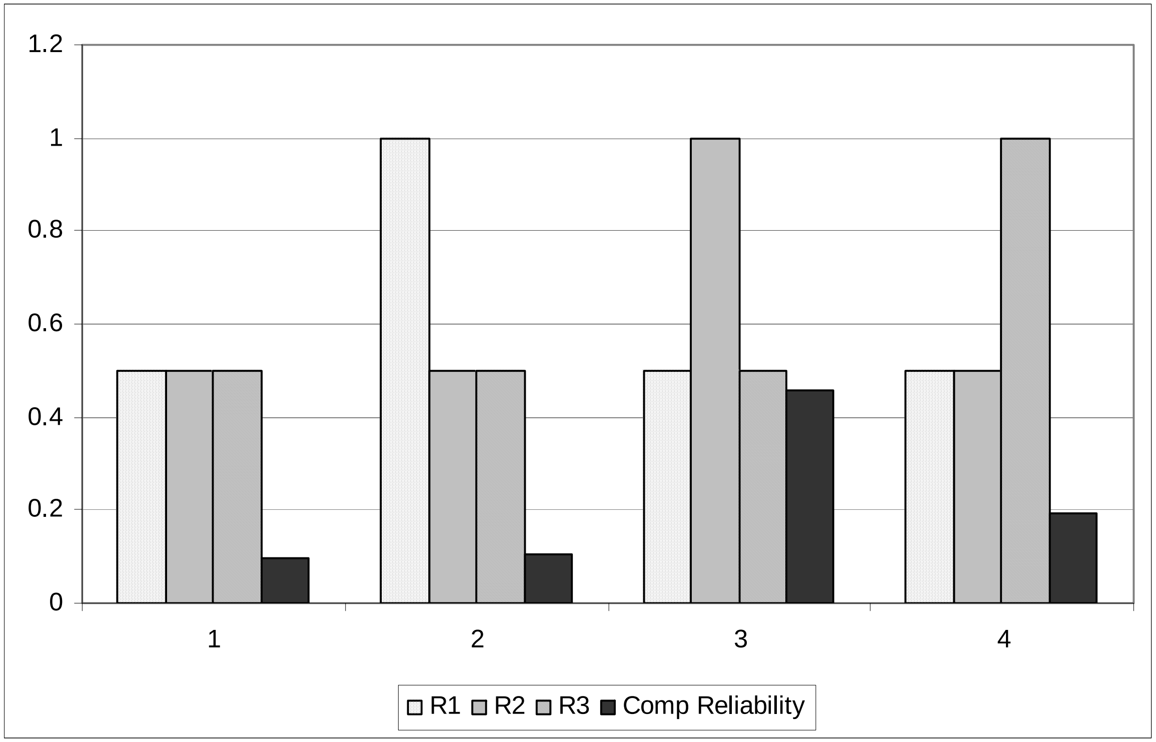
$$ITP' = \begin{bmatrix} 0.05 & 0.9 & 0.05 \\ 0.018 & 0.36 & 0.622 \\ 0.22 & 0.65 & 0.13 \end{bmatrix}$$

$$Rand.Fluc.\% = \begin{bmatrix} \text{-93.33\%} & 12.50\% & 80.00\% \\ 555.55\% & 55.55\% & \text{-48.23\%} \\ 900.00\% & \text{-23.52\%} & 15.38\% \end{bmatrix}$$





Sensitivity Analysis



- Tied with the cost framework can offer cost-effective mitigation strategies

Complexity and Scalability

- Complexity of event-based Markov Model: $O(N^2 \times M \times T)$
- Our event/action based model: $O(N^2 \times M \times K \times T)$
  - N: num states, M: num events
  - K: num actions, T: length of training data
- M and K are fixed, but N can be reduced using *hierarchy*