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

Local Reliability

Markov Model

Interface

Protocols

Component

Static Behaviors

Dynamic Behaviors

“The Quartet”

Global Reliability

Static Behaviors

Dynamic Behaviors

Component Interface Protocols

“The Quartet”

Global Reliability

Local Reliability
Component Reliability

- Analysis
- Defects
- Classification
- Cost framework
- State Reliability

- Domain Knowledge
- OR
- Random

- Model Extractor
- State-based Markov model
- Training data

- Hidden Markov Modeling
- Reliability Estimator
- Baum-Welch Algorithm
- Comp Reliability
- Transition Probabilities

- ITP
- Initial transition probabilities

Legend:
- Artifacts
- Major steps of the approach
- Numerical values
- Iterative process

G(\hat{\theta}(t), f)
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**
   - External view of the component
   - Specifies its legal interactions with other components in the system
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);
Architectural Models → Defects → Quantification

Quantification:
- Classification
- Cost framework $G(\tilde{\theta}(t), f)$

State Reliability

Domain Knowledge

Model Extractor
- State-based Markov model
- Training data

Hidden Markov Modeling

Reliability Estimator
- Baum-Welch Algorithm

Comp Reliability

Transition Probabilities

Legend:
- Artifacts
- Major steps of the approach
- Numerical values
- Iterative process
- Initial transition probabilities

Component Reliability
Component Reliability

Architectural Models → Analysis → Defects → State Reliability

Domain Knowledge
OR
Random

Hidden Markov Modeling

Model Extractor → State-based Markov model

ITP

Reliability Estimator

Baum-Welch Algorithm

Comp Reliability

Transition Probabilities

Legend:
- Artifacts
- Numerical values
- Major steps of the approach
- Iterative process
- Initial transition probabilities

Numerical values

Iterative process

Initial transition probabilities
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 $t$
  \[ c_t = G(\vec{\theta}(t), f), \text{ where} \]
  \[ \vec{\theta}(t) = [\theta_1(t), \theta_2(t), ..., \theta_n(t)] \]
- $f$: 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 $j$

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

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

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

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

DYNAMIC BEHAVIOR

- Stop to Manual
- Manual to Cruise
- Cruise to Manual
- Manual to Stop
- Cruise to Stop
- Stop to Cruise
- Brake/Decelerate to Manual
- Gas/Accelerate to Manual
- Brake/Decelerate to Cruise
- Cruise to Brake/Decelerate
- Brake/Decelerate to Stop
- Gas/Accelerate to Stop
- Cruise to Gas/Accelerate
- Gas/Accelerate to Cruise
- Brake/Decelerate to Gas/Accelerate
Partial Markov Extension

Diagram:
- Stop
- Gas/accelerate
- Brake/decelerate
- Cruise
- Gas/accelerate
- Cruise/maintain
- Brake/decelerate
- Gas/accelerate
- Manual
- Gas/accelerate
- Brake/decelerate
- Gas/accelerate
- TRUE
## Transition Probabilities

<table>
<thead>
<tr>
<th>Origin State</th>
<th>Observation</th>
<th>Pr(O)</th>
<th>Reaction</th>
<th>Pr(R)</th>
<th>Total Pr</th>
<th>Dest. State</th>
</tr>
</thead>
<tbody>
<tr>
<td>stop</td>
<td>TRUE</td>
<td>0.1</td>
<td>TRUE</td>
<td>1</td>
<td>0.1</td>
<td>stop</td>
</tr>
<tr>
<td>stop</td>
<td>gas</td>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stop</td>
<td>gas</td>
<td>0.05</td>
<td>accelerate</td>
<td>1</td>
<td>0.05</td>
<td>stop</td>
</tr>
<tr>
<td>stop</td>
<td>gas</td>
<td>0.05</td>
<td>accelerate</td>
<td>1</td>
<td>0.05</td>
<td>cruise</td>
</tr>
<tr>
<td>cruise</td>
<td>break</td>
<td>0.85</td>
<td>decelerate</td>
<td>1</td>
<td>0.85</td>
<td>manual</td>
</tr>
<tr>
<td>cruise</td>
<td>TRUE</td>
<td>0.85</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cruise</td>
<td>gas</td>
<td>0.02</td>
<td>accelerate</td>
<td>1</td>
<td>0.02</td>
<td>stop</td>
</tr>
<tr>
<td>cruise</td>
<td>gas</td>
<td>0.03</td>
<td>accelerate</td>
<td>1</td>
<td>0.03</td>
<td>cruise</td>
</tr>
<tr>
<td>manual</td>
<td>TRUE</td>
<td>0.2</td>
<td>TRUE</td>
<td>1</td>
<td>0.2</td>
<td>manual</td>
</tr>
<tr>
<td>manual</td>
<td>gas</td>
<td>0.08</td>
<td>accelerate</td>
<td>1</td>
<td>0.08</td>
<td>manual</td>
</tr>
<tr>
<td>manual</td>
<td>gas</td>
<td>0.02</td>
<td>accelerate</td>
<td>0.6</td>
<td>0.012</td>
<td>cruise</td>
</tr>
<tr>
<td>manual</td>
<td>gas</td>
<td>0.02</td>
<td>accelerate</td>
<td>0.4</td>
<td>0.008</td>
<td>stop</td>
</tr>
<tr>
<td>manual</td>
<td>break</td>
<td>0.08</td>
<td>decelerate</td>
<td>1</td>
<td>0.08</td>
<td>manual</td>
</tr>
<tr>
<td>manual</td>
<td>break</td>
<td>0.01</td>
<td>accelerate</td>
<td>1</td>
<td>0.01</td>
<td>cruise</td>
</tr>
<tr>
<td>manual</td>
<td>break</td>
<td>0.01</td>
<td>decelerate</td>
<td>1</td>
<td>0.01</td>
<td>stop</td>
</tr>
<tr>
<td>manual</td>
<td>cruise</td>
<td>0.6</td>
<td>maintain</td>
<td>1</td>
<td>0.6</td>
<td>cruise</td>
</tr>
</tbody>
</table>

\[
P = \begin{bmatrix} 0.15 & 0.8 & 0.05 \\ 0.018 & 0.36 & 0.622 \\ 0.02 & 0.85 & 0.13 \end{bmatrix}
\]

\[
 \text{ITP} = \begin{bmatrix} 0.6 \\ 1 \\ 0.6 \\ 0.85 \\ 0.13 \end{bmatrix}
\]

**Baum-Welch**
Reliability Model

- Adaptation of Cheung1980

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

\[
\hat{P}(i,j) = \begin{bmatrix}
C & F & S_1 & S_2 & \ldots & S_j & \ldots & S_N \\
C & 1 & 0 & 0 & 0 & \ldots & 0 & \ldots & 0 \\
F & 0 & 1 & 0 & 0 & \ldots & 0 & \ldots & 0 \\
S_i & 0 & 1 - R_i & R_i T_{i1} & R_i T_{i2} & \ldots & R_i T_{ij} & \ldots & R_i T_{iN} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
S_{N-1} & 0 & 1 - R_{N-1} & R_{N-1} T_{(N-1)1} & R_{N-1} T_{(N-1)2} & \ldots & R_{N-1} T_{(N-1)j} & \ldots & R_{N-1} T_{(N-1)N} \\
S_N & R_N & 1 - R_N & R_N T_{N1} & R_N T_{N2} & \ldots & R_N T_{Nj} & \ldots & R_N T_{NN}
\end{bmatrix}
\]

\[ R_{comp} = \hat{P}^n(S_1, C) \]
Example…

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

\[
\begin{bmatrix}
  0.15 & 0.8 & 0.05 \\
  0.018 & 0.36 & 0.622 \\
  0.02 & 0.85 & 0.13 \\
\end{bmatrix}
\]

\[
R_{\text{stop}} = 0.87, \quad R_{\text{manual}} = 0.9, \quad R_{\text{cruise}} = 0.76
\]

\[
\hat{P} = \begin{bmatrix}
  1 & 0 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 & 0 \\
  0 & 0.13 & 0.067 & 0.744 & 0.086 & 0 \\
  0 & 0.014 & 0.362 & 0.522 & 0 & 0 \\
  0 & 0.24 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[
R_{\text{comp}} = Q^{-1}(1, \text{cruise}) \times R_{\text{cruise}}
\]

\[
R_{\text{comp}} = 0.7444 \times 0.76 \\
\approx 0.5657
\]

\[
\Rightarrow R_{\text{comp}} \approx %56
\]
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
  \( StateInv \Rightarrow CompInv \)

- State Invariants vs. Operation Post-Condition
  \( StateInv \Rightarrow PostCond \)
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

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

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

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