Composing, Analyzing and Validating Models to Assess the Performability of Competing Design Candidates

CptS 580.1 / 483.1 Software Specification and Analysis

Frederick T. Sheldon
Assistant Professor of Computer Science
Washington State University
Research Agenda

⊕ Goal: Verification and validation of systems and software
⊕ Modern high-assurance systems
⊕ Advantages of a formal approach
⊕ How do we get there from here: Modeling Cycle
⊕ Safety and reliability analysis:
  ◊ Railroad Switching System including Design-to-Cost
  ◊ Vehicle Braking/Traction/Steering Control System
  ◊ Operating System with Dynamic Priority Mechanism
⊕ Summary of ongoing work
**Verification and Validation**

**Verification** determines if the products of a given phase of the **SW life cycle** fulfill the **requirements** established during the previous phase.

- Formal proof of program correctness
- Reviewing, inspecting, testing, checking, auditing, or otherwise establishing and documenting whether or not items, processes, services, or documents conform to specified requirements (ANSI/ASQC A3-1978).

**Validation** checks if the program, as implemented, meets the expectations of the customer in such a way to ensure compliance with software requirements.
Modern High-Assurance Systems

Share five key attributes:

- **Reliable**, meaning they are correct,
- **Available**, meaning they remain operational,
- **Safe**, meaning they are impervious to catastrophe (fail-safe),
- **Secure**, meaning they will never enter a hazardous state,
- **Timely**, meaning their results will be produced on time and satisfy deadlines (timing correctness).
Advantages of Formal Specification

⊕ Provides insights into the requirements / design
⊕ Specifications may be analyzed mathematically
   ⊙ Demonstrate consistency and completeness
   ⊙ Prove the implementation corresponds to the specification
   ⊙ Help identify appropriate test cases
   ⊙ Characterize aspects of the specification more precisely:
     • Structural, Functional, and Logical
     • Behavioral
       – Dynamic: timing combined with probabilistic nature
     • Data oriented.
⊕ And, the potential for cost savings….
Expenditure Profile Changes

From Ian Sommerville, Software Engineering (5th Ed.)
The Vision

Methods and tools are needed for the creation of safe and correct systems. . .

Reduce the effort of constructing reliable models for . . .

- Application level safety, performance and reliability analysis
- Improved tractability for verifying correctness and for solving large stochastic models
- Reasoning about unambiguous specifications and designs

Need for an integrated environment to provide interoperability among formalisms

- Link stochastic analysis with correctness checking
- Allow various formal methods to be applied independently based on a common representation form.
- Demonstrate on industrial strength problems
- Learn what works and what doesn’t
Integrated Environment to Provide Interoperability

**Modeling Formalisms**
(independent languages/methods, theories and tools)

- **High Level Description Language**
  - Mosel
  - Promela
  - P-CSP

- **Stochastic Analysis**
  - Mosel

- **Model Checking**
  - Promela

- **Stochastic Analysis**
  - P-CSP

- **Model Verification and Validation**
  - MOSES
  - SPIN
  - SPNP

**Formalisms Interoperate**
(integrated together in an open toolkit with a common interface)

- **High Level Description Language**
  - Mosel
  - Promela

- **Graphical Editor**
  - Panda

- **Meta Language**
  - CSPL

- **Graph layout**
  - DUO Solvers
  - GUK and FTA
  - SPNP (black box)

- **Model Verification and Validation**
  - MOSES
  - Exists

© F. T. Sheldon
Washington State University
The Modeling Cycle

⊕ Descriptive modeling
⊕ Computational modeling
  ☒ Making it tractable
⊕ Model solution
⊕ Validation and model refinement
  ☒ Operational
  ☒ Proposed
Railway Switching System

Hope that gate closes in time!

Requirements Analysis and Specification

...for the purpose of Safety Assurance and Design-to-Cost
Railway Switching Example

Requirements:

⊕ Two Basic Properties the system must satisfy
  ⊖ Safety property – the gate is down during all occupancy intervals
  ⊖ Utility property – the gate is open when no train is in the crossing

⊕ The Solution in General Terms:
  ✥ Two Processes: The TRAIN and the GATE
  ✥ TRAIN sends an "arriving" signal to the GATE as it nears the intersection and proceeds towards the intersection.
  ✥ GATE, upon receiving the signal, closes the gate and remains closed until the train departs.
  ✥ TRAIN sends a "departing" signal after leaving the intersection.
  ✥ GATE, upon receiving the signal opens the gate and remains open.
  ✥ The two processes repeat continuously.

This model encompasses the environment which includes the train(s) and the gate, as well as the interface between them.
Compose a Functional Model Using the Process Algebra CSP translated to SPNs

TRAIN =
(IN_TRANSIT);
(GATE ! a → AT_INTERSECTION);
(GATE ! d → TRAIN)

GATE =
(TRAIN ? a → CLOSE);
(TRAIN ? d → OPEN → GATE)

RAIL_ROAD_CROSSING = TRAIN ||_{a,d} GATE

Problem: A hazard exists which becomes more evident viewed as a Petri net

Several possible failure modes exist: (1) communication failure [t_2, t_4, t_5 and t_7], (2) mechanical failure [t_6 and t_9], and (3) timing failure [t_3 occurs before t_6] (i.e., train arrives at intersection before the gate has closed).
Refined System Model

Hazard Removed

\[
\text{TRAIN} = \\
(\text{IN\_TRANSIT}); \\
(GATE \triangleleft a \rightarrow \text{GATE} \triangleright \text{ok} \rightarrow \\
\text{AT\_INTERSECTION}); \\
(GATE \triangleright d \rightarrow \text{TRAIN})
\]

\[
\text{GATE} = \\
(\text{TRAIN} \triangleright a \rightarrow \text{CLOSE} \rightarrow \text{TRAIN} \triangleright \text{ok}); \\
(\text{TRAIN} \triangleright d \rightarrow \text{OPEN} \rightarrow \text{GATE})
\]

\[
\text{SAFER}\_\text{RAIL}\_\text{ROAD}\_\text{CROSSING} = \\
\text{TRAIN} \parallel [a,\text{ok},d] \text{ GATE}
\]
Several possible failure modes exist: (1) communication failure \([t_2, t_4, t_5 \text{ and } t_8]\), (2) mechanical failure \([t_6 \text{ and } t_9]\), and (3) timing failure \([t_3 \text{ occurs before } t_7]\) (i.e., train arrives at intersection before the gate has completely closed).
Generate the ERG/RG $\rightarrow$ Markov

\[ M_1 \xrightarrow{\mu_1} M_2 \xrightarrow{\mu_2} M_3 \xrightarrow{\lambda_3} M_4 \xrightarrow{\lambda_4} M_5 \xrightarrow{\lambda_5} M_6 \xrightarrow{\lambda_6} M_7 \xrightarrow{\lambda_7} M_8 \xrightarrow{\lambda_8} M_9 \xrightarrow{\lambda_9} M_{10} \]

- **Safe states**
- **Hazardous states**
- **Failed states**

- $\lambda_m = \text{mechanical failure rate}$
- $\lambda_c = \text{communication failure rate}$
- $\mu = \text{repair rate}$

**Critical Failure**
- Train at intersection, but approaching msg never received!

**Timing Failure**
- Train at intersection, msg rcv'd and gate closing!

**Non-Critical Failure**
- Train gone, but the gate failed to open properly!
Reliability Prediction

Results:

- Run 1: $\text{Rel}[10,000] = 4.58042 \times 10^{-40}$, Mttf = $1.09934 \times 10^5 \text{tus}$
- Run 2: $\text{Rel}[10,000] = 4.58554 \times 10^{-9}$, Mttf = $5.20472 \times 10^5 \text{tus}$
- Run 3: $\text{Rel}[10,000] = 1.07427 \times 10^{-5}$, Mttf = $8.73755 \times 10^5 \text{tus}$
- Run 4: $\text{Rel}[10,000] = 2.34974 \times 10^{-5}$, Mttf = $9.37937 \times 10^5 \text{tus}$
- Run 5: $\text{Rel}[10,000] = 2.56342 \times 10^{-5}$, Mttf = $9.45662 \times 10^5 \text{tus}$
- Run 6: $\text{Rel}[10,000] = 2.58888 \times 10^{-5}$, Mttf = $9.46547 \times 10^5 \text{tus}$
- Run 7: $\text{Rel}[10,000] = 3.44604 \times 10^{-1}$, Mttf = $6.15169 \times 10^6 \text{tus}$

Input Parameters:

1. $\tau_5 = 0.00908$, $\lambda_{3, 4, 8, 9} = 1.0 \times 10^{-7}$, $\lambda_5, 10 = 1.0 \times 10^{-4}$
2. $\tau_5 = 0.000908$, $\lambda_{3, 4, 8, 9} = 1.0 \times 10^{-7}$, $\lambda_5, 10 = 1.0 \times 10^{-4}$
3. $\tau_5 = 0.0000908$, $\lambda_{3, 4, 8, 9} = 1.0 \times 10^{-7}$, $\lambda_5, 10 = 1.0 \times 10^{-4}$
4. $\tau_5 = 0.00000908$, $\lambda_{3, 4, 8, 9} = 1.0 \times 10^{-7}$, $\lambda_5, 10 = 1.0 \times 10^{-4}$
5. $\tau_5 = 0.0$, $\lambda_{3, 4, 8, 9} = 1.0 \times 10^{-7}$, $\lambda_5, 10 = 1.0 \times 10^{-4}$
6. $\tau_5 = 0.0$, $\lambda_{3, 4, 8, 9} = 0.0$, $\lambda_5, 10 = 1.0 \times 10^{-4}$
7. $\tau_5 = 0.0$, $\lambda_{3, 4, 8, 9} = 0.0$, $\lambda_5, 10 = 1.0 \times 10^{-5}$

*Time units: each x-axis tick is 1000tus. If 1 tu = second, then ~16mins/tick, or 10,000 ticks ~2778hrs (full range of data).

**Constants: $\mu_1 = 0.0001$, $\mu_2-4, 7, 8 = 1.0$, $\mu_9, 10 = 1.0$, while $\mu_5$ and $\mu_6$ were held set at 0.1 and 0.01 respectively.
Design-to-Cost

Evaluate (judiciously) the costs (and benefits) for providing fault-avoidance and/or fault-tolerance using a cost function to optimize design parameters.

\[ Q = \omega_p(failure) + \phi \int_0^\infty \theta P_\theta(\theta) d\theta + \nu \]

....where \( w \) = cost of failure, \( f \) = cost of delay/time units, \( n \) = cost of the gate/train passing and the average train travel time is

\[ = \int_0^\infty \theta P_\theta(\theta) d\theta \]

\[ \nu(gmpt) = \frac{(40-gmpt)^4 + 20,000}{100} \]

....is the gate cost per run as a function of the gmpt (gate most probable closing time).

†These numbers have been exaggerated intentionally to make the variations of the cost function more visible. Otherwise, a gate that cost $20,000 plus better operate more than just 100 times!
Costs May Be Correlated to Design Parameters

Cost

Train Arrival Time

Gate Close Time

© F. T. Sheldon
Washington State University
Sure hope I can stop this in time!

Safety and Reliability Analysis
TC/ABS Functional Description
(Traction Control / Antilock Brake System)

ABS maintains steer-ability and driving stability under skidding conditions

Anti-Slip control maintains adhesion to the road and driving stability

Electronic Stability program maintains limits among yaw-rate, steering-angle, and lateral velocity preventing under/over-steer
TC/ABS Schematic

Antilock Breaking / Antiskid Controller

Break Pressure

Electronic brake control module (EBCM)

Hydraulic modulator valve assembly

Master break cylinder

Disc break (4 indpt)
Wheel speed sensor (4 indpt)
B₁-4 = Brakes (LF, RF, LR, RR)
S₁-4 = Speed sensors (LF, RF, LR, RR)
R₁-₂ Turing angles (of the vehicle and the tires respectively)
Skid+Steering Control System

- **If** Any-Wheel-Locks **then**
  Pulsate-Locked-Wheel

- **If** Either-Rear-Wheel-Slips **then**
  Brake-Slipping-Wheel

- **If** Under-Steer-Left **then**
  Brake(Left-Front, Left-Rear)

- **If** Under-Steer-Right **then**
  Brake(Right-Front, Right-Rear)

- **If** Over-Steer-Left **then**
  Brake(Right-Rear, Right-Front)

- **If** Over-Steer-Right **then**
  Brake(Left-Rear, Left-Front)
Deciding how the faults affect nominal and off nominal operation

Failure modes
- Loss of vehicle
- Loss of stability
- Degraded function
- Over/Under-steer
Entity Life History Diagram

⊕ Descriptive Modeling
⊕ View of the system
- Braking
- Steering
- Skidding (not shown)

⊕ Structure Chart
- Invocation structure
- Choices (pathways)
- Flow

© F. T. Sheldon
Washington State University
ABS Skidding Control

- Computational Modeling
- Skidding of any tire may be detected
  - Compensation mechanism cycles (loop counter-clock-wise) until skidding ceases
  - Fault may occur activating a failure mode causing:
    - Loss of vehicle
    - Loss of stability
    - Degraded function
    - Over/Under-steer
Slipping/Traction Control

⊕ Rear wheels lose traction
  ⊘ Compensation mechanism is one shot process
  ⊘ Fault may occur activating a failure mode causing:
    • Loss of stability
    • Degraded function
Over/Under-Steer Control

⊕ When over/under-steer threshold is detected

⊗ Compensation mechanism is a one shot process

⊗ Fault may occur activating a failure mode causing:
  • Loss of stability
  • Degraded function
  • Over/Under-steer
Derive Failure Rate Mappings

- Determine causality
  - Fault
  - Symptom
  - Suspect component
- Calculate cumulative failure rates
  - Assign to failure transitions in SPN

<table>
<thead>
<tr>
<th>Fault &gt;</th>
<th>One Wheel (PL)</th>
<th>One Wheel (LB)</th>
<th>One Axle (PL)</th>
<th>One Axle (LB)</th>
<th>Both Axles (PL)</th>
<th>Both Axles (LB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wheel Speed Sensor</td>
<td>2.00E-10</td>
<td>2.00E-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pressure Sensor</td>
<td></td>
<td>1.50E-10</td>
<td>1.50E-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Brake Cylinder</td>
<td></td>
<td></td>
<td>1.00E-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pressure Limiting Valve</td>
<td></td>
<td></td>
<td>6.00E-12</td>
<td>6.00E-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inlet Valve</td>
<td></td>
<td></td>
<td>6.00E-12</td>
<td>6.00E-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drain Valve</td>
<td></td>
<td></td>
<td>6.00E-12</td>
<td>6.00E-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toggle Switching Valve</td>
<td></td>
<td></td>
<td>6.00E-12</td>
<td>6.00E-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic Pump</td>
<td></td>
<td></td>
<td>6.80E-10</td>
<td>6.80E-10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pressure Tank</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.00E-11</td>
<td></td>
</tr>
<tr>
<td>Controller</td>
<td></td>
<td></td>
<td>6.00E-11</td>
<td>6.00E-11</td>
<td>6.00E-11</td>
<td>6.00E-11</td>
</tr>
<tr>
<td>Steering Angle Sensor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lateral Accel Sensor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yaw Rate Sensor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tubing</td>
<td></td>
<td></td>
<td>3.00E-11</td>
<td>3.00E-11</td>
<td>3.00E-11</td>
<td></td>
</tr>
<tr>
<td>Piping</td>
<td></td>
<td></td>
<td>4.00E-11</td>
<td>4.00E-11</td>
<td>4.00E-11</td>
<td></td>
</tr>
<tr>
<td>Cumulative Failure Rate</td>
<td>3.36E-10</td>
<td>2.66E-10</td>
<td>1.62E-09</td>
<td>2.28E-10</td>
<td>1.08E-09</td>
<td>2.10E-10</td>
</tr>
</tbody>
</table>
Modeling an Operating System with Stochastic Petri Nets
Dynamic Priority OS
Functional Level Abstraction

⊕ Each Elementary Block
   ⊖ Analytic Sub-model
⊕ Dynamic Priorities
   ⊖ Guarantee high priority jobs get shorter response times

Goal: Evaluate dynamic increasing/decreasing priority assignments.
SPN of Dynamic Priority OS

+ Top: complete system contexts
  - Kernel (SIH)
  - System (SYS)
  - IO
  - User

+ Bottom:
  - Detailed User Context
Complete System SPN

SYS context
- t-SYS-ser
- SYS-ser
- t-CPU-sys
- t-SYS

SIH context
- SIH
- t-CPU-sih
- SIH-ser
- t-SIH-ser
- t-arrival

IO context
- DISK
- t-disk
- I-disk-ser

USER context
- CPU
- P0
- S1
- t-CPU-USER
- USERser
- TS
- Stop
- t-end
- decision

t-preemption

© F. T. Sheldon
Washington State University
Detailed User Context

All priority (L < N) jobs stored here (N=number of priorities)
User Context: Basic Characteristics

⊕ Lower priority than other contexts

⊗ Gets CPU when there are no jobs to be processed in other contexts.

⊗ Lower priority is assigned to transitions $T_i$... than to transitions $t_{CPU_{sys}}$ and $t_{CPU_{sih}}$.

⊗ Transitions $T_i$... enabled when no other jobs are being served $\Rightarrow$ number of tokens in places $PP_i = 0$.

⊕ When transition $T_i$... fires a token in the CPU place is removed.

⊗ Jobs are processed in priority order.

⊗ Inhibitor arc from $P1$ ($P_i$) to $T2$ ($T_{i+1}$) guarantees a priority class $i$ job is processed before class $i+1$.

⊗ Token in $S1 \Rightarrow$ the CPU is processing a USER context job of priority $i$ ($\Rightarrow$ by token in $PP_i$).
## System Parameters

### System Parameters (job arrival rate $\lambda_{arrival} = 0.005$)

<table>
<thead>
<tr>
<th>Component Definition</th>
<th>Transition Probability</th>
<th>Service Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Subsystem Context</td>
<td>$p_{io} = 0.05$</td>
<td>$s_{io} = 20$</td>
</tr>
<tr>
<td>System Context</td>
<td>$p_{sys} = 0.40$</td>
<td>$s_{sys} = 1.0$</td>
</tr>
<tr>
<td>User Subsystem Context</td>
<td>$p_{user} = 0.54$</td>
<td>$s_{user} = 1.0$</td>
</tr>
<tr>
<td>Kernel Subsystem Context</td>
<td>$p_{end} = 0.01$</td>
<td>$s_{sih} = 0.5$</td>
</tr>
</tbody>
</table>
Predicted vs. Measured Results

Transient + Steady State Analysis

Number of jobs in the queues vs. Time [sec]

Mean response time [sec] vs. Number of CPU's

- Arrival rate = 0.1
- Arrival rate = 0.001
- Measured values
Summary of Ongoing Work

**Ongoing**

- Extending the CSPN language
- GUI with SPN Editor $\leftarrow \rightarrow$ CSPL
- Promela-based models $\rightarrow$ SPNs (i.e., CSPL)
- CSPL $\rightarrow$ ERG $\rightarrow$ RG $\rightarrow$ Q-matrix $\rightarrow$ Solved analytically
- Fault-tree analysis (Erlangen)
- Implementation of solution methods (Erlangen)

**Exploring the concept of**

- Relate stochastic results back (mechanically) $\rightarrow$ original model as a process of refinement in light of prior runs (sensitivity analysis)
- CGI Web-based access to CSPN (and other components)
The Vision

Methods and tools are needed for the creation of safe and correct systems. ...

- Reduce the effort of constructing reliable models for ...
  - Application level safety, performance and reliability analysis
  - Improved tractability for verifying correctness and for solving large stochastic models
  - Reasoning about unambiguous specifications and designs

- Need for an integrated environment to provide interoperability among formalisms
  - Link stochastic analysis with correctness checking
  - Allow various formal methods to be applied independently based on a common representation form.
  - Demonstrate on industrial strength problems
  - Learn what works and what doesn’t
Integrated Environment to Provide Interoperability

Modeling Formalisms
(independent languages/methods, theories and tools)

High Level Description Language
- Mosel
- Promela
- P-CSP

Model Verification and Validation
- Mosel
- Promela (SPIN)
- P-CSP

Formalisms Interoperate
(integrated together in an open toolkit with a common interface)

High Level Description Language
- Mosel
- Promela (SPIN)
- P-CSP

Model Verification and Validation
- GUK and FTA
- DUO Solvers
- SPNP (black box)

Meta Language CSPL

Graph layout

© F. T. Sheldon
Washington State University
The end… time to shut down!

Questions?