Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with Modest

Holger Hermanns, Arnd Hartmanns
Saarland University, Germany
based on joint work with
Jonathan Bogdoll, Henrik Bohnenkamp, Pedro R. D’Argenio, Alexandre David, Ernst Moritz Hahn, and Joost-Pieter Katoen
All models are wrong, but some models are useful.

(George E. P. Box)

System under study / implementation

Model checking

Correctness  Safety  Performance  Costs

(Slide inspired by Jan Tretmans, Embedded Systems Institute, Eindhoven)

Holger Hermanns  Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
All models are wrong, but some models are useful.

(George E. P. Box)

What are useful models?

Wireless Sensor Networks:
- concurrency
- message loss
- transmission delays
- randomised algorithms
- limited battery power
All models are wrong, but some models are useful.

(George E. P. Box)

What are useful models?

**ETCS Level 3:**

- concurrency
- transmission delays
- message loss
- measurement errors
- continuous dynamics

Holger Hermanns  
*Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST*
Quantitative models are useful.

- 1% probability of message loss
- 20 mW needed in send mode
- Expected time for transmission \( \leq 8 \text{s} \)?
- Fraction of time in send mode \( \leq 0.2 \) ?

\[
\dot{v} = a \land v \cdot v_{\text{max}} \leq 2 \cdot B_{ON} \cdot (s_f - auth) \\
pos_{\text{seen}} = \mathcal{N}(pos_{\text{real}}, 5\text{m})
\]

\[\text{Prob}(\text{crash within 15 years}) \leq 10^{-5} ?\]
Quantitative models are useful.

Quantities in models

- time
- probabilities
- costs
- continuous dynamics

Quantities in requirements/properties

- Quantified safety: \( \text{Prob(\text{crash within 15 years})} \leq 10^{-5} \)
- Performance: Expected time for transmission \( \leq 8 \) s
- Dependability, Performability, Survivability, ...
- + qualitative requirements in a quantitative setting

\( \dot{v} = a \land v \cdot v_1 \)
The automata-based approach

```c
while(true)
    next:
    get_data(buf);
    n = 2;
    while(n > 0)
        e = snd_data(buf);
        if(e == SUCCESS)
            report_success();
            goto next;
        if(e == TIMEOUT)
            n = n - 1;
            report_failure();
```

Holger Hermanns  Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Modelling and Verification

The automata-based approach

Properties of interest
– Absence of deadlocks
– Safety
– Liveness
– LTL or CTL formulas
  e.g. $\forall \Box \exists \Diamond$ success

Boolean requirements
Quantitative Models

A quantitative automata family

Labelled Transition Systems

LTS nondeterminism

Holger Hermanns

Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Quantitative Models

A quantitative automata family

Labelled Transition Systems
Discrete-Time Markov Chains

Holger Hermanns
Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Quantitative Models

A quantitative automata family

Labelled Transition Systems
Discrete-Time Markov Chains
Markov Decision Processes
Probabilistic Automata

Holger Hermanns  Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Quantitative Models

A quantitative automata family

Labelled Transition Systems
Discrete-Time Markov Chains
Markov Decision Processes
Probabilistic Timed Automata

Holger Hermanns
Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Quantitative Models

A quantitative automata family

Labelled Transition Systems
Discrete-Time Markov Chains
Markov Decision Processes
Probabilistic Timed Automata
Stochastic Timed / Hybrid Automata

Holger Hermanns
Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
A Stochastic Hybrid Automaton (Network)

\[ s = 200 \quad , \quad v = 0 \]

**FreeRun**

\[
\begin{align*}
v & \in [0, v] \\
a & \in [a, a] \\
\frac{\dot{v}}{t} & = a \\
\frac{\dot{s}}{t} & = v \\
v \cdot v & \leq 2b \quad (s - auth)
\end{align*}
\]

**AutoBrake**

\[
\begin{align*}
v & \in [0, v] \\
a & = a \\
\frac{\dot{v}}{t} & = a \\
\frac{\dot{s}}{t} & = v \\
v \cdot v & \geq 2b \quad (s - auth)
\end{align*}
\]

\[ v = 83.4, \quad len = 200 \quad , \quad sd = 400 \quad , \quad a = -1.4 \]

**Leader**

\[ s_l = 1400 \quad , \quad v_l = 0 \]

**FreeRun**

\[
\begin{align*}
v_l & \in [0, v] \\
a_l & \in [a, a] \\
\frac{\dot{v}_l}{t} & = a_l \\
\frac{\dot{s}_l}{t} & = v_l
\end{align*}
\]

\[ auth = 800 \]

**Moving Block**

\[ c' = 0 \quad , \quad auth' = m - len - sd \]

\[ c' = 0 \]

\[ c \geq 8 \]

\[ m' = N(s_l, \sigma) \]

\[ c \geq 8 \]

\[ \frac{m}{c} = 0 \quad , \quad \frac{c}{c} = 1 \quad , \quad c \leq 8 \]

\[ v = 0.7, \quad b = -0.7, \quad b = -0.3 \]

Holger Hermanns

Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Quantitative Models

A quantitative automata family

Nondeterminism
– structural or temporal

Probabilistic choices
– discrete or continuous
– over next state or delay

Time
– discrete or continuous
– nondeterministic or random delays

Holger Hermanns  Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Quantitative Models

Automata modelling formalisms and model checking tools

Modest
– The Modest Toolset

Guarded commands
– PRISM, PASS, ...

graphical
UPPAAL TA – UPPAAL

Promela etc – SPIN etc

Holger Hermanns Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Models for Simulation

Modest: A Modelling and Description Language for Stochastic Timed Systems

Language features:
- Variables and assignments
- Processes and recursion
- Exception handling
- Deadlines & invariants
- Random variable sampling
- bool, int, arrays
- Clocks
- Rewards/costs
- Probabilistic branching

Example: Lossy channel with transmission delay

process Channel() {
    clock c;
    snd? palt {
        : 2: {==} // msg lost
        :98: {= c = 0, x = Uni(0, TD) =};
        invariant(c <= x) when(c >= x) rcv!
    };
    Channel()
}

Stochastic Timed Automata Semantics

Holger Hermanns  Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Modest – the language

high-level language
focus on readability, expressivity and conciseness

process Sender() {
    bool bit;
    int(0..MAX) rc;
    new_file {= i = 0, rc = 0 =};
    try {
        do {
            when(i < N) {= i = i + 1 =};
        } do {
            put_k {= ff = (i == 1), lf = (i == N), ab = bit =}
        alt {
            get_l {= bit = !bit, rc = 0 =};
            break
        } when(rc == MAX && i < N)
        s_nok {= rc = 0 =};
        throw(error)
        ...
    }
}
The Modest Toolset

mctau – mcptau – prohver – modes – mime – mosta

four analysis tools

semantics

GUI

Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST

Holger Hermanns
mctau – mcpta – prohver – modes – mime – mosta

mctau  Model-checking for TA using UPPAAL
Export from Modest to UPPAAL with layout
Overapproximation of probabilistic choices

Bogdoll, David, H., H.: mctau: Bridging the Gap between Modest and UPPAAL (SPIN 2012)

Holger Hermanns  Stochastic, Hybrid and Real-Time Systems: From
The Modest Toolset

mctau – mcpta – prohver – modes – mime – mosta

mctau  Model-checking for TA using UPPAAL
       Export from Modest to UPPAAL with layout
       Overapproximation of probabilistic choices

mcpta  Model-checking for PTA using PRISM
       Export from Modest to Guarded Commands


Holger Hermanns  Stochastic, Hybrid and Real-Time Systems: From
The Modest Toolset

mctau – mcptu – prohver – modes – mime – mosta

mctau  Model-checking for TA using UPPAAL
       Export from Modest to UPPAAL with layout
       Overapproximation of probabilistic choices

mcpta  Model-checking for PTA using PRISM
       Export from Modest to Guarded Commands

modes  Simulation & Statistical Model Checking for STA
       with spurious nondeterminism

The Modest Toolset

mctau – mcpta – prohver – modes – mime – mos7a

mctau  Model-checking for TA using UPPAAL
        Export from Modest to UPPAAL with layout
        Overapproximation of probabilistic choices

mcpta  Model-checking for PTA using PRISM
        Export from Modest to Guarded Commands

modes  Simulation & Statistical Model Checking for STA
        with spurious nondeterminism

prohver  Safety Verification for SHA
         Using (modified) HA Solver Phaver

Hahn, H., H., Katoen: A Compositional Modelling and Analysis Framework For Stochastic Hybrid Systems (FMSD 13)
The Modest Toolset

Modest

mime

PRISM Guarded Commands

UPPAAL .xml

Networks of Stochastic Hybrid Automata

SHA

STA

PTA

MDPs

prohver

modes

mcpta

mctau

modified PHAVer

PRISM ≥ 4.0

UPPAAL ≥ 4.1

Holger Hermanns

Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Safety verification process for SHA in prohver

SHA → PHA → overapproximation of continuous distributions → decomposition → HA + probs → PHAVer → LTS → reconstruction → MDP → value iteration → Results

Holger Hermanns
Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Case Study - ETCS level 3

SHA model
- two trains – leader and follower – and Comm+RBC

Continuous aspects
- acceleration, deceleration, speed
- acceleration of leader nondeterministic (within train limits)

Stochastic aspects
- position measurements scattered with normal distribution
- message loss probability during communication
Case Study - ETCS level 3

\[ s = 200, \, v = 0^{-\infty} \]

**FreeRun**
\[
\begin{align*}
\mathbf{v} \in [0_{-s}, v] \\
\mathbf{a} \in [a, a] \\
\frac{\mathbf{v}}{t} &= \mathbf{a} \\
\frac{\mathbf{s}}{t} &= \mathbf{v} \\
v \cdot v &\leq 2b \ (s - auth)
\end{align*}
\]

**AutoBrake**
\[
\begin{align*}
\mathbf{v} \in [0_{-s}, v] \\
\mathbf{a} &= \mathbf{a} \\
\frac{\mathbf{v}}{t} &= \mathbf{a} \\
\frac{\mathbf{s}}{t} &= \mathbf{v} \\
v \cdot v &\geq 2b \ (s - auth)
\end{align*}
\]

\[ v = 83.4_{-s}, \ \text{len} = 200, \ \text{sd} = 400 \]

**Leader**
\[
\begin{align*}
s_i = 1400, \ v_i = 0^{-\infty} \\
\mathbf{v}_i \in [0_{-s}, v] \\
\mathbf{a}_i \in [a, a] \\
\frac{\mathbf{v}_i}{t} &= \mathbf{a}_i \\
\frac{\mathbf{s}_i}{t} &= \mathbf{v}_i
\end{align*}
\]

**Crash**
\[
\begin{align*}
v \cdot v &\geq 2b \ (s - auth) \\
v \cdot v &\leq 2b \ (s - auth)
\end{align*}
\]

\[ auth = 800 \]

**Moving Block**
\[
\begin{align*}
c' &= 0 \\
auth' = m - \text{len} - sd
\end{align*}
\]

**Idle**
\[
\begin{align*}
c &= 1 \\
c &\leq 8
\end{align*}
\]

**Send**
\[
\begin{align*}
c &= 0 \\
c &= 1 \\
c &\leq 8
\end{align*}
\]

**Send**
\[
\begin{align*}
0.7^{-\infty}, \ b &= -0.7^{-\infty} \\
b &= -0.3^{-\infty}
\end{align*}
\]

Holger Hermanns

Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Case Study - ETCS level 3

```java
const real TIME_BOUND;
property P_Crash = Pmax(<> (s_f >= s_l - L) && time <= TIME_BOUND);

process Leader()
{
  var a; // acceleration
  var v = 0; der(v) = a; // speed

  // The leading train can exhibit acceleration and max speed
  // within its acceleration and max speed range except for driving backwards
  invariant(der(s_l) == v && A_MIN <= a && a <= A_MAX && v <= v_MAX)
}

process Follower()
{
  var a; // acceleration
  var v = 0; der(v) = a; // speed

  invariant(der(s_f) == v && 0 <= v)
  do {
    // train is running normally
    invariant(A_MIN <= a && a <= A_MAX && v * v_MAX <= 2 * B_MAX)
    when(v * v_MAX >= 2 * B_ON * (s_f - auth)) tau;
    // forced braking by ETCS system
    invariant(a == A_MIN && v * v_MAX >= 2 * B_OFF * (s_f - auth))
  }
}
```

<table>
<thead>
<tr>
<th>time bound</th>
<th>probability ((\sigma = 10, 15, 20))</th>
<th>build (s)</th>
<th>states</th>
</tr>
</thead>
<tbody>
<tr>
<td>60s</td>
<td>7.110E-19 6.215E-09 2.141E-05</td>
<td>65</td>
<td>571</td>
</tr>
<tr>
<td>80s</td>
<td>1.016E-18 8.879E-09 3.058E-05</td>
<td>201</td>
<td>1440</td>
</tr>
<tr>
<td>100s</td>
<td>1.219E-18 1.066E-08 3.669E-05</td>
<td>470</td>
<td>2398</td>
</tr>
<tr>
<td>120s</td>
<td>1.524E-18 1.332E-08 4.587E-05</td>
<td>1260</td>
<td>4536</td>
</tr>
<tr>
<td>140s</td>
<td>1.727E-18 1.509E-08 5.198E-05</td>
<td>2541</td>
<td>6568</td>
</tr>
<tr>
<td>160s</td>
<td>2.031E-18 1.776E-08 6.116E-05</td>
<td>5764</td>
<td>10701</td>
</tr>
</tbody>
</table>

Holger Hermanns  
*Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST*
Case Study - Power Grid Control Strategies

All over Germany, masses of photovoltaic microgenerators are rolled out:

- **2009**: 10 GW
- **2011**: 25 GW
- **2020**: ?? GW

Current state of control:

**EN 50438:2007**, in force since 2007:
- Switch off when frequency > 50.2 Hz

**VDE-AR-N 4105**, required today:
- Output linear function of frequency in [50.2, 51.5] Hz
- Emergency switchoff above 51.5 Hz
- Switch on again when < 50.05 Hz for 1 minute

Holger Hermanns

*Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST*
Stability of Grids and Controllers

Simulation of synthetic background load scenarios

- On-off
- Linear
- AIMD
- Frequency dependent probabilistic switching with exponential backoff

Holger Hermanns
Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with MODEST
Fairness of Controllers

Max/min/average output per generator:
Modest Applications

Communication protocols

Wireless sensor networks

Dependability evaluation

Industrial production scheduling

Renewable electric power generation
**Modest and SHA**

- language and model
  - for quantitative systems
    - with quantitative requirements

\[
\{ = x = \text{uni}(0, 3) = \} \quad E_{\text{max}} [\text{time to finish}]
\]

\[
\text{var } v, a; \\
\text{invariant(der(v) == a) ...}
\]

\[
\text{invariant(c <= TD\_MAX)}
\]

\[
\text{par } \{ \\
\quad \text{snd palt } \{ \\
\quad \quad :99: \text{rcv} \\
\quad \quad : 1: \text{tau} \}
\}
\]

\[
\Rightarrow \text{single-formalism,}
\]

\[
\text{multiple-solution approach}
\]
The Modest Toolset - Summary

modelling language: Modest

+ PRISM guarded commands
+ UPPAAL xml

prohver for SHA - using Phaver
mcpta for PTA/MDP - using PRISM
mctau for TA - using UPPAAL
modes for simulation despite nondeterminism

Demo at demo session on Friday!

Installation assistance anytime!