Nobodys’s Perfect: Monitoring Systems that Work Most of the Time

Maximilian Schwenger
Joint work with Jan Baumeister, Peter Faymonville, Bernd Finkbeiner, Malte Schledjewski, Marvin Stenger, Leander Tentrup, Hazem Torfah
Why Should We Bother?
Our Vision

System

Formal Guarantees on Runtime Behavior
1. Never injure humans.
2. Obey orders.
3. Protect yourself

Analysing the system, we have:

**Analysis**

**Spec**

**Monitor**

**System**

**Health**

Our Take on Runtime Verification
1. Never injure humans.
2. Obey orders.
3. Protect yourself.

**Analysis**

The status of the GPS module deteriorates.
1. Never injure humans.
2. Obey orders.
3. Protect yourself.

Analysis

Path planner ran into exponential case.
1. Never injure humans.
2. Obey orders.
3. Protect yourself.

**Static Verification**

System S
Controller C
Specification $\phi$

**Verify:**

$$\forall \sigma \in \text{runs}(S \parallel C): \sigma \models \phi$$
Scalability

\[ \dot{p} = Rv \]
\[ \dot{R} = R\dot{\omega} \]
\[ \dot{v} = -\omega \times v + R^T\tilde{g} + f_v(\omega, v, \alpha, \beta, \omega_r, \delta_c, \delta_r) \]
\[ \dot{\omega} = -J^{-1}(\omega \times J\omega) + f_w(\omega, v, \alpha, \beta, \omega_r, \delta_c, \delta_r) \]
\[ \dot{\alpha} = f_\alpha(\omega, v, \alpha, \beta, \omega_r, \delta_a, \delta_e) \]
\[ \dot{\beta} = f_\beta(\omega, v, \alpha, \beta, \omega_r, \delta_a, \delta_e) \]
\[ \dot{\omega}_r = f_r(\omega, \omega_r, \delta_c, \delta_r) \]
1. Never injure humans.
2. Obey orders.
3. Protect yourself.

**Verify:**

\[ \forall \sigma \in \text{runs}(S \parallel C): \sigma \models \varphi \]
Testing

Verify:
\[ \forall \sigma \in \text{runs}(S \parallel C): \sigma \models \varphi \]

\[ \exists S' \subseteq \text{runs}(S \parallel C): \forall \sigma \in S': \sigma \models \varphi \]
Given $\sigma \in \text{runs}(S \parallel C)$:

$\sigma \models \phi$

Verify:

$\forall \sigma \in \text{runs}(S \parallel C): \sigma \models \phi$
Prior to Deployment

System

After Deployment
Never injure humans.
2. Obey orders.
3. Protect yourself.

Our Take on Runtime Verification
STREAM-BASED RUNTIME VERIFICATION

Output Streams

TRIGGER

TRIGGER

RUNTIME MONITOR
input lat, lon: Float64 // from GPS
input accel_x: Float64 // from accelerometer
input slow_down_cmd: Bool

The GPS module operates with at least 5Hz.
input lat, lon: Float64 // from GPS
input accel_x: Float64 // from accelerometer
input slow_down_cmd: Bool

output gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_samples < 5 “GPS frequency less than 5 Hz.”
input lat, lon: Float64 // from GPS
input accel_x: Float64 // from accelerometer
input slow_down_cmd: Bool

output gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_samples < 5 “GPS frequency less than 5 Hz.”

output accel_velo @1Hz := accel_x.aggregate(over: 5s, using: \int)
output gps_velo @1Hz := lon.aggregate(over: 5s, using: \nabla)

trigger abs(accel_velo - gps_velo) > 0.1 “Conflicting measurements for velocity.”
output fast := accel_velo > 700
output slow_down := fast. offset (by: -1)
defaults (to: false) ∧ ¬fast

trigger ¬slow_down_cmd. aggregate (over: 5s, using: \exists) ∧ slow_down.
hold ()
defaults (to: false)

“Spurious Slow-Down.”

Accelerometer and GPS readings coincide.
\textbf{RTL}ola \textbf{IN a NUTSHELL}

\textbf{input} lat, lon: \texttt{Float64} // from GPS  
\textbf{input} accel\_x: \texttt{Float64} // from accelerometer  
\textbf{input} slow\_down\_cmd: \texttt{Bool}

\textbf{output} gps\_samples @1Hz := lat.aggregate(over\_exactly: 1s, using: count)  
\textbf{trigger} gps\_samples < 5 “GPS frequency less than 5 Hz.”

\textbf{output} accel\_velo @1Hz := accel\_x.aggregate(over: 5s, using: \( \int \))  
\textbf{output} gps\_velo @1Hz := lon.aggregate(over: 5s, using: \( \nabla \))  
\textbf{trigger} abs(accel\_velo - gps\_velo) > 0.1 “Conflicting measurements for velocity.”
input lat, lon: Float64 // from GPS
input accel_x: Float64 // from accelerometer
input slow_down_cmd: Bool

output gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_samples < 5 “GPS frequency less than 5 Hz.”

output accel_velo @1Hz := accel_x.aggregate(over: 5s, using: \int)
output gps_velo @1Hz := lon.aggregate(over: 5s, using: \nabla)
trigger abs(accel_velo - gps_velo) > 0.1 “Conflicting measurements for velocity.”

A slow-down is preceded by the respective command.
**RTLola in a Nutshell**

```
input lat, lon: Float64 // from GPS
input accel_x: Float64 // from accelerometer
input slow_down_cmd: Bool

output gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_samples < 5 “GPS frequency less than 5 Hz.”

output accel_velo @1Hz := accel_x.aggregate(over: 5s, using: ∫)
output gps_velo @1Hz := lon.aggregate(over: 5s, using: ∇)
trigger abs(accel_velo - gps_velo) > 0.1 “Conflicting measurements for velocity.”

output fast := accel_velo > 700
output slow_down := fast.offset(by: -1).defaults(to: false) ∧ ¬fast
trigger @1Hz ¬slow_down_cmd.aggregate(over: 5s, using: ∃)
∧ slow_down.hold().defaults(to: false) “Spurious Slow-Down.”
```
1. Never injure humans.
2. Obey orders.
3. Protect yourself.

Our Take on Runtime Verification
**Strong Type System**

**Input**
- `lat, lon: Float64` // from GPS
- `accel_x: Float64` // from accelerometer
- `slow_down_cmd: Bool`

**Output**
- `gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)`
- Trigger `gps_samples < 5 “GPS frequency less than 5 Hz.”`

- `accel_velo @1Hz := accel_x.aggregate(over: 5s, using: ∫)`
- `gps_velo @1Hz := lon.aggregate(over: 5s, using: ∇)`
- Trigger `abs(accel_velo - gps_velo) > 0.1 “Conflicting measurements for velocity.”`

- `fast := accel_velo > 700`
- `slow_down := fast.offset(by: -1).defaults(to: false) ∧ ¬fast`
- Trigger `@0.5Hz ¬slow_down_cmd.aggregate(over: 5s, using: ∃) ∧ slow_down.hold().defaults(to: false) “Spurious Slow-Down.”`
**Strong Type System**

**Input**
- lat, lon: `Float64` // from GPS
- accel_x: `Float64` // from accelerometer
- slow_down_cmd: `Bool`

**Output**
- gps_samples @ 1Hz := lat.aggregate(over_exactly: 1s, using: count)
- trigger gps_samples < 5 “GPS frequency less than 5 Hz.”
- accel_velo @ 1Hz := accel_x.aggregate(over: 5s, using: $\int$)
- gps_velo @ 1Hz := lon.aggregate(over: 5s, using: $\nabla$)
- trigger abs(accel_velo - gps_velo) > 0.1 “Conflicting measurements for velocity.”
- fast := accel_velo > 700
- slow_down := fast.

**Trigger**
- @0.5Hz ¬slow_down_cmd.aggregate(over: 5s, using: $\exists$) ∧ slow_down.hold().defaults(to: false) “Spurious Slow Down.”
**Strong Type System**

input lat, lon: Float64 // from GPS
input accel_x: Float64 // from accelerometer
input slow_down_cmd: Bool

output gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_samples < 5 “GPS frequency less than 5 Hz.”

output accel_velo @1Hz := accel_x.aggregate(over: 5s, using: \(\int\))
output gps_velo @1Hz := lon.aggregate(over: 5s, using: \(\nabla\))
trigger abs(accel_velo - gps_velo) > 0.1 “Conflicting measurements for velocity.”

output fast := accel_velo > 700
output slow_down := fast.offset(by: -1).defaults(to: false) \& \neg fast
trigger @0.5Hz \neg slow_down_cmd.aggregate(over: 5s, using: \(\exists\))
\& slow_down.hold().defaults(to: false) “Spurious Slow-Down.”
**SPECIFICATION**

**input** lat, lon: Float64 // from GPS  
**input** accel_x: Float64 // from accelerometer  
**input** slow_down_cmd: Bool

**output** gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)  
**trigger** gps_samples < 5 “GPS frequency less than 5 Hz.”

**output** accel_velo @1Hz := accel_x.aggregate(over: 5s, using: ∫)  
**output** gps_velo @1Hz := lon.aggregate(over: 5s, using: ∇)  
**trigger** abs(accel_velo - gps_velo) > 0.1  
“Conflicting measurements for velocity.”

**output** fast := accel_velo > 700  
**output** slow_down := fast.offset(by: -1).defaults(to: false) ∧ ¬fast  
**trigger** @1Hz ¬slow_down_cmd.aggregate(over: 5s, using: ∃)  
∧ slow_down.hold().defaults(to: false) “Spurious Slow-Down.”

**DEPENDENCY GRAPH**

```
lat -- gps_smpl -- trig_1  
lon -- g_velo -- trig_2  
acc -- a_velo -- fast  

trig_1 -> slow_down  
trig_2 -> slow_down

fast -- slow_down
```
output $h \ := \ s.\text{aggr}(\text{over: } 1.5p, \text{using: } \gamma)$
output $h$ := s.aggr(over: 1.5p, using: $\gamma$)
output $h := s.aggr(over: 1.5p, using: \gamma)$
**Sliding Windows**

**output h** := s.aggr(over: 1.5p, using: γ)
\textbf{List Homomorphisms}

\[
\text{output } h := \text{s.aggr(over: 1.5p, using: } \gamma) \]

\(\gamma : A^* \rightarrow B\)

\(\text{map}_\gamma : A \rightarrow T \quad \text{fin}_\gamma : T \rightarrow B \quad \circ_{\gamma} : T \times T \rightarrow T\)

\(\gamma(v_1, \ldots, v_n) = \text{fin}_\gamma(\text{map}_\gamma(v_1) \circ_{\gamma} \cdots \circ_{\gamma} \text{map}_\gamma(v_n))\)

\(\triangleright\text{Meertens: } \text{“Algorithmics: Towards programming as a mathematical activity”, 1986}\)
output $h := s \text{aggr}(\text{over}: 1.5p, \text{using}: \gamma)$
output $h \ @ p^{-1} Hz := s.aggr(over: 1.5p, using: \gamma)$
output $h \@ p^{-1} Hz := s.aggr(over: 1.5p, using: \gamma)$
output $h \, @p^{-1}Hz := s.aggr(over: 1.5p, using: \gamma)$

\textbf{Sliding Windows}

Li et al.: “No Pane, No Gain: Efficient Evaluation of Sliding-window Aggregates over Data Streams”, SIGMOD Rec. 2005

Our Take on Runtime Verification

1. Never injure humans.
2. Obey orders.
3. Protect yourself.

System

Monitor

Health

Analysis

Spec

Never injure humans.
2. Obey orders.
3. Protect yourself.
1. Never injure humans.
2. Obey orders.
3. Protect yourself.

RTLOLA
SPECIFICATION

ANNOTATED DG
INTERMEDIATE REP.

BACKEND
RUST INTERPRETATION

Graphical UI developed by Sanny schmitt

1. Never injure humans.
2. Obey orders.
3. Protect yourself.

RT Lola Specification

**Rust Interpreter**

**Specification:**
- GPS frequency validation
- GPS/IMU jump detection
- Hover phase detection

**Results:**
- 433,000 events
- 1,545ns per event @ 146%
- Stack size < 1kB, no heap

Thanks to Sanny Schmitt for designing the interface!
1. Never injure humans.
2. Obey orders.
3. Protect yourself.
VHDL/FPGA Compilation

**HLC**

- **PERIODIC**
  - Sys Clk
  - Scheduler
  - HLQ I/F

- **EVENT-BASED**
  - Ext I/F
  - Event Prep

**LLC**

**COORDINATOR**

- Idle
  - ~empty
  - done \(\wedge\) ~empty

- Pop
  - T
  - enable

- Eval
  - done

**EVALCONTROLLER**

- Idle
  - enable

- Prep

- Layer\(_n\)
  - \(\ldots\)

- Layer\(_0\)

**Queue**

- empty
- pop
- data, affected streams, time

---


Baumeister, Finkbeiner, Schwenger, Torfah, “On the Similarities of Aircraft and Humans”, CyberCardia@ESWeek2019
**SPECIFICATION**

**input** lat, lon: Float64 // from GPS

**input** accel_x: Float64 // from accelerometer

**input** slow_down_cmd: Bool

**output** gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)

**trigger** gps_samples < 5 “GPS frequency less than 5 Hz.”

**output** accel_velo @1Hz := accel_x.aggregate(over: 5s, using: ∫)

**output** gps_velo @1Hz := lon.aggregate(over: 5s, using: ∇)

**trigger** abs(accel_velo - gps_velo) > 0.1

“Conflicting measurements for velocity.”

**output** fast := accel_velo > 700

**output** slow_down := fast.offset(by: -1).defaults(to: false) ∧ ¬fast

**trigger** @1Hz ¬slow_down_cmd.aggregate(over: 5s, using: ∃)

∧ slow_down.hold().defaults(to: false) “Spurious Slow-Down.”

**DEPENDENCY GRAPH**

[Diagram showing dependencies between variables: lat, lon, acc, cmd, gps_smpl, g_velo, a_velo, fast, slow_dn, trig_1, trig_2, trig_3.]
**SPECIFICATION**

**Input**
- `lat`, `lon`: `Float64` // from GPS
- `accel_x`: `Float64` // from accelerometer
- `slow_down_cmd`: `Bool`

**Output**
- `gps_samples @1Hz := lat.aggregate(over_exactly: 1s, using: count)`
  
- `trigger gps_samples < 5 “GPS frequency less than 5 Hz.”`

**Output**
- `accel_velo @1Hz := accel_x.aggregate(over: 5s, using: \(\int\))`
  
- `output gps_velo @1Hz := lon.aggregate(over: 5s, using: \(\nabla\))`
  
- `trigger abs(accel_velo - gps_velo) > 0.1`
  
  “Conflicting measurements for velocity.”

**Output**
- `fast := accel_velo > 700`
  
- `output slow_down := fast.offset(by: -1).defaults(to: false) \& \neg fast`

**Trigger**
- `@1Hz \neg slow_down_cmd.aggregate(over: 5s, using: \(\exists\))`
  
  \& slow_down.hold().defaults(to: false) “Spurious Slow-Down.”
<table>
<thead>
<tr>
<th></th>
<th>FF</th>
<th>LUT</th>
<th>MUX</th>
<th>CA</th>
<th>MULT</th>
<th>Pwr [W]</th>
<th>Time [μs]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Drone</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>3036</td>
<td>3685</td>
<td>26</td>
<td>656</td>
<td>10</td>
<td>1.620</td>
<td>4.28</td>
</tr>
<tr>
<td>HLC</td>
<td>901</td>
<td>156</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>543</td>
<td>442</td>
<td>0</td>
<td>43</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLC</td>
<td>1281</td>
<td>2820</td>
<td>0</td>
<td>576</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>1905</td>
<td>1533</td>
<td>23</td>
<td>226</td>
<td>23</td>
<td>1.570</td>
<td>3.20</td>
</tr>
<tr>
<td>HLC</td>
<td>550</td>
<td>161</td>
<td>0</td>
<td>37</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>330</td>
<td>342</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLC</td>
<td>895</td>
<td>927</td>
<td>0</td>
<td>161</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cmd-Resp</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Parallel</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>6379</td>
<td>13794</td>
<td>0</td>
<td>849</td>
<td>0</td>
<td>1.582</td>
<td>3.77</td>
</tr>
<tr>
<td>HLC</td>
<td>936</td>
<td>232</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>540</td>
<td>326</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLC</td>
<td>4903</td>
<td>13236</td>
<td>0</td>
<td>971</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sequential</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>6909</td>
<td>14768</td>
<td>0</td>
<td>851</td>
<td>0</td>
<td>1.581</td>
<td>43.83</td>
</tr>
<tr>
<td>HLC</td>
<td>936</td>
<td>232</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>534</td>
<td>326</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLC</td>
<td>5433</td>
<td>14210</td>
<td>0</td>
<td>973</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Never injure humans.
2. Obey orders.
3. Protect yourself.
FUTURE DIRECTIONS
**input** lat, lon: `Float64` // from GPS
**input** accel_y: `Float64` // from accelerometer
**input** slow_down_cmd: `Bool`

**output** imu_velo @1Hz := accel_y.aggregate(over: 1s, using: \( \int \)) + imu_velo.offset(by: -1)
**output** imu_pos @1Hz := imu_velo.aggregate(over: 1s, using: \( \int \)) + imu_pos.offset(by: -1)
**trigger** abs(imu_pos - lat) > 0.1 "Conflicting measurements for position estimation."

\[
v(t) = \int_{0}^{t} a(\tau)d\tau = \sum_{i=0}^{t-1} \left( \int_{i}^{i+1} a(\tau)d\tau \right)
\]
import integration // aggregations, functions (sqrt), macros (indef. integration, haversine…)

input lat, lon: Float64 // from GPS
input accel_y: Float64 // from accelerometer
input slow_down_cmd: Bool

output imu_velo @1Hz := accel_y.aggregate(over: ∞, using: ∫)
output imu_pos @1Hz := indef_integral(imu_velo)
trigger abs(imu_pos - lat) > 0.1 “Conflicting measurements for position estimation.”

\[ v(t) = \int_0^t a(\tau) d\tau = \sum_{i=0}^{t-1} \left( \int_i^{i+1} a(\tau) d\tau \right) \]
Find $C$ s.t.

$$\forall \sigma \in \text{runs}(A \parallel S \parallel C): \sigma \models \varphi$$
Find $C$ s.t.

$$\forall \sigma \in \text{runs}(A \parallel S \parallel C): \sigma \models \varphi$$

1. Never injure humans.
2. Obey orders.
3. Protect yourself.
Outlook
Outlook

Check out StreamLAB: stream-lab.eu  Contact: schwenger@react.uni-saarland.de
APPENDIX

Better Have It and Not Need It....
Heck et al., “Two-year seizure reduction in adults with medically intractable partial onset epilepsy treated with responsive neurostimulation: Final results of the RNS System Pivotal trial”, Epilepsia 2014

Sun et al., “Responsive Cortical Stimulation for the Treatment of Epilepsy”, Neurotherapeutics 2008

Heck et al., "Two-year seizure reduction in adults with medically intractable partial onset epilepsy treated with responsive neurostimulation: Final results of the RNS System Pivotal trial", Epilepsia 2014

Kossoff et al., "Effect of an External Responsive Neurostimulator on Seizures and Electrographic Discharges during Subdural Electrode Monitoring", Epilepsia 2004

Sun et al., "Responsive Cortical Stimulation for the Treatment of Epilepsy", Neurotherapeutics 2008
Huge thanks to Leander, Marvin, and Malte!
**RTLola: Medical Domain**

**input** CLS: Float64
**input** rec, stim: Bool

**output** jerk := abs(derive(3, CLS))

**output** avg_long @100mHz := jerk.aggr(over: 2000s, using: avg)
**output** avg_short @1kHz := jerk.aggr(over: 2ms, using: avg)

**output** spike @1kHz := avg_short > avg_long.hold() + ε

**trigger** spike ∧ ¬rec.aggregate(over: 2ms, using: any) "Seizure not recognized"
**trigger** @1kHz rec.aggregate(over: 5ms, using: any) ∧ ¬stim.aggr(over: 3ms, using: any) "Stimulation not triggered"
Through the Zoo of RV Approaches*

**Logic**
- \( \text{LTL}_3 [1] \)
- RV-LTL [2]
- rLTL [3]
- STL/MTL [4]

**Data-Driven**
- RV-LTL [2]
- rLTL [3]
- MTL + Aggr [5]

**SW Tie-Ins**

* rather a tiny fraction thereof

---

THROUGH THE ZOO OF RV APPROACHES*

Logics

- LTL3 [1]
- RV-LTL [2]
- rLTL [3]
- STL/MTL [4]
- MTL + Aggr [5]
- Lustre [6]
- (RTOS) Striver [7]

Data-Driven

- RT
- St
- FS
- Em
- RT

SW Tie-Ins

- RT
- St
- Lustre [6]
- Lola [9] -2.0 [10];
- RT- [11]; FPGA- [12]

* rather a tiny fraction thereof

Through the Zoo of RV Approaches*

Logics

- LTL
- LTL₃
- RV-LTL
- rLTL
- MTL
- STL/MTL

Data-Driven

- Em
- RT
- Fs
- (RT)

Tie-Ins

- STL/MTL
- MTL + Aggr
- Lustre
- Lola
- JavaMOP
- Aspects
- DTrace
- RT-2.0
- Striver
- FPGA

Software

- RT
- St
- Fs
- Em

* rather a tiny fraction thereof

---


