Distributed and Heterogeneous Event-based Monitoring

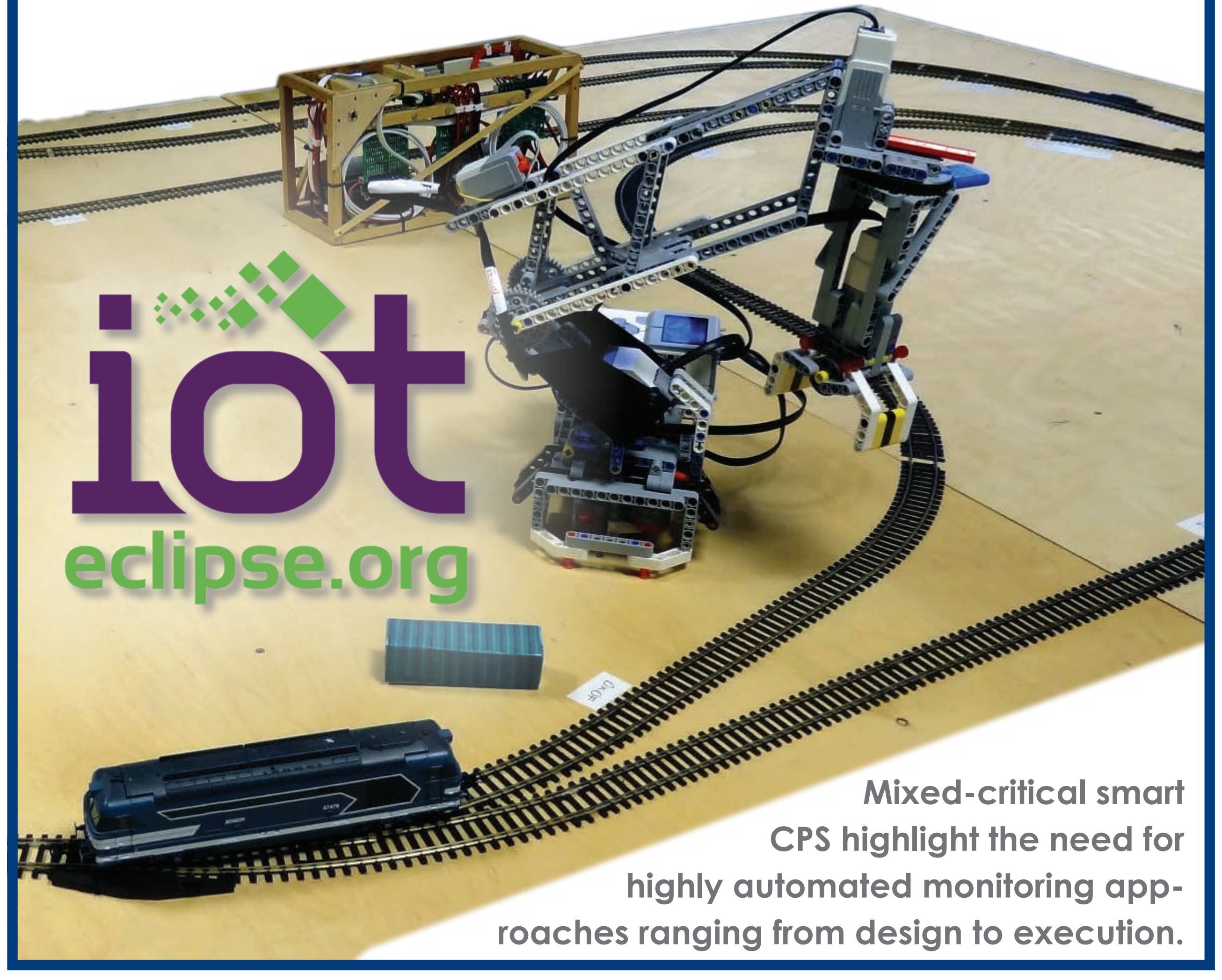
in Smart Cyber-Physical Systems

László Balogh[1], István Dávid[2], István Ráth[3], Dániel Varró[1,4], András Vörös[1,4]

[1]Budapest University of Technology and Economics,
[2]University of Antwerp, [3]IncQuery Labs Ltd
[4]MTA-BME Lendület Cyber-Physical Systems Research Group

Motivation

Design-time verification of smart cyber-physical systems is often infeasible due to their complexity. Monitoring techniques offer a run-time alternative for verification. The architectural characteristics of smart CPS raise the need for **integrated** techniques for monitor **specification**, **deployment** and **execution** of the monitoring logic.



Mixed-critical smart CPS highlight the need for highly automated monitoring approaches ranging from design to execution.

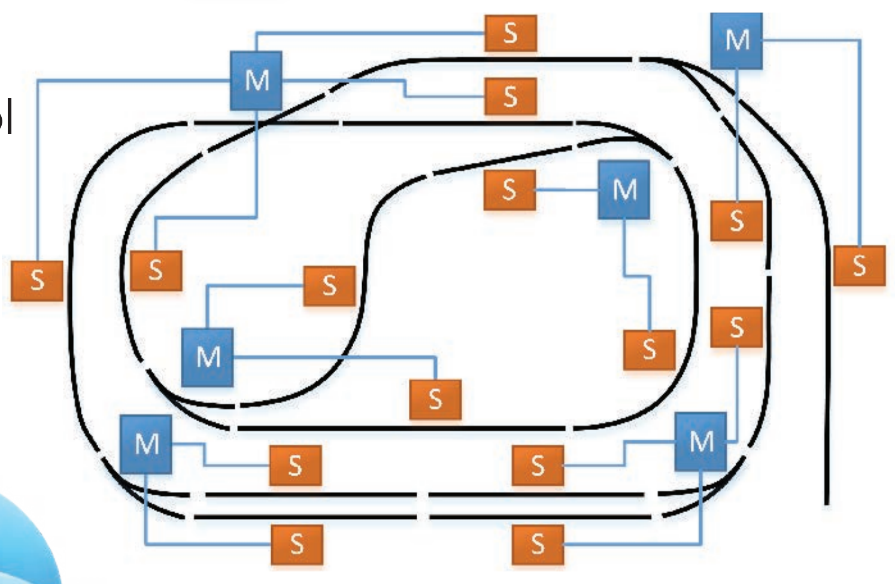
Challenges

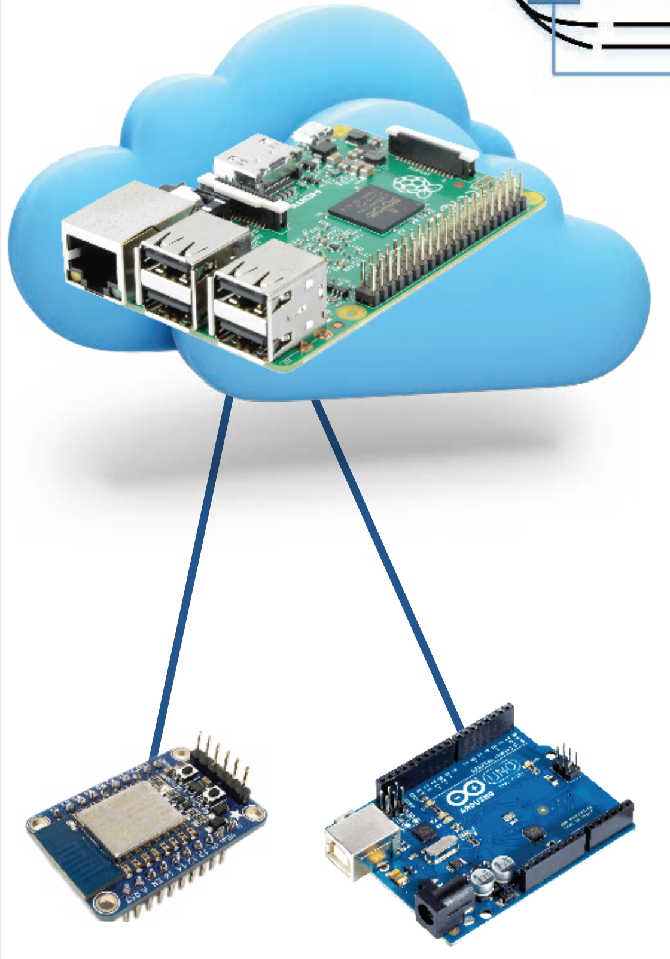
Mixed-critical systems

Safety-critical systems
- Fault tolerance
- Verification
- Constrained resources

Internet-of-Things
- State-of-the-art HW/SW
- Low reliability

- Smart cities
- Traffic control
- Smart grid

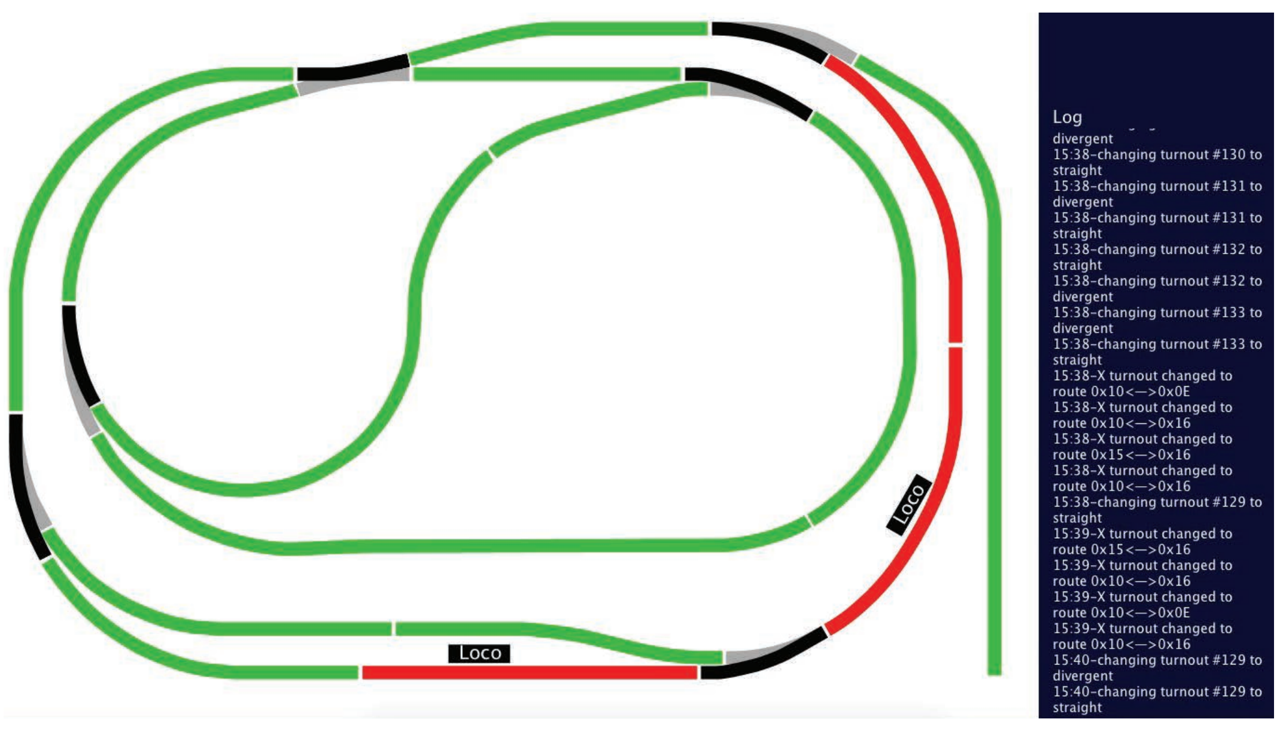




- Exploit IoT to optimize critical systems
- Enhanced control based on sensor data

- Distributed
- Heterogeneous
- (Near) Real-time

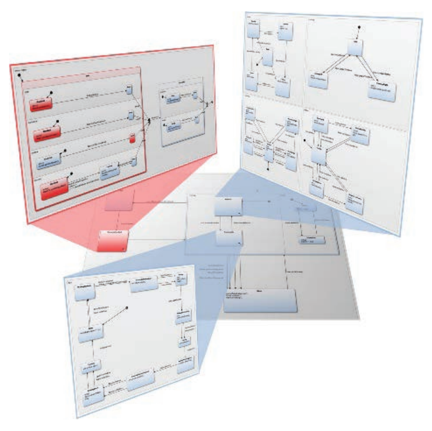
Monitoring



- Run-time verification
- Supplement design-time verification
- Computationally cheaper

High level specification languages

- Bridging the semantic gap between design-time and run-time models
- Coordination
- Abstraction
- Multi-formalism
- Semantic integration



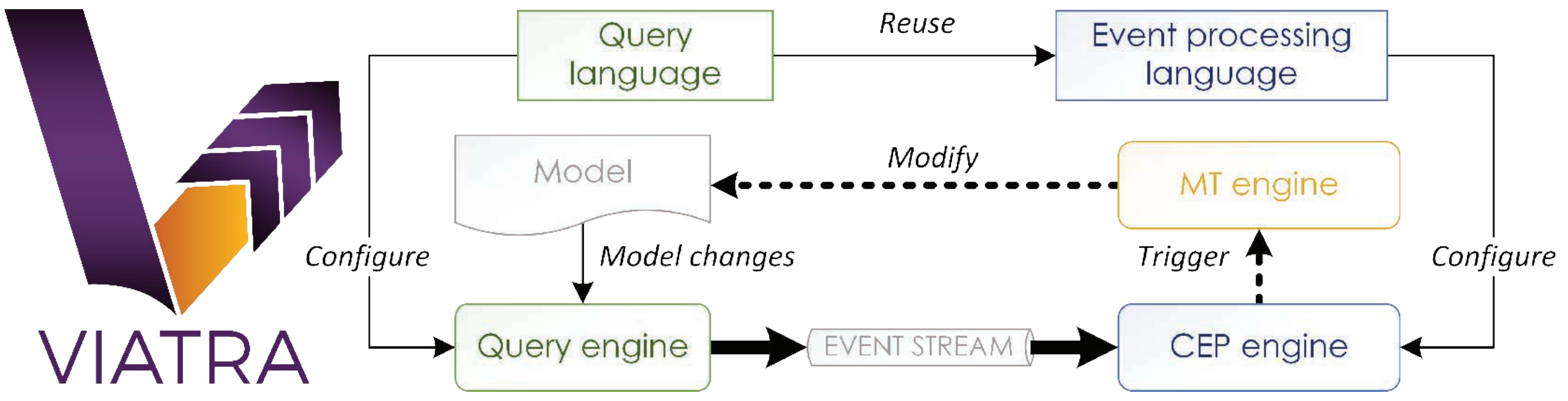
Proposed solution

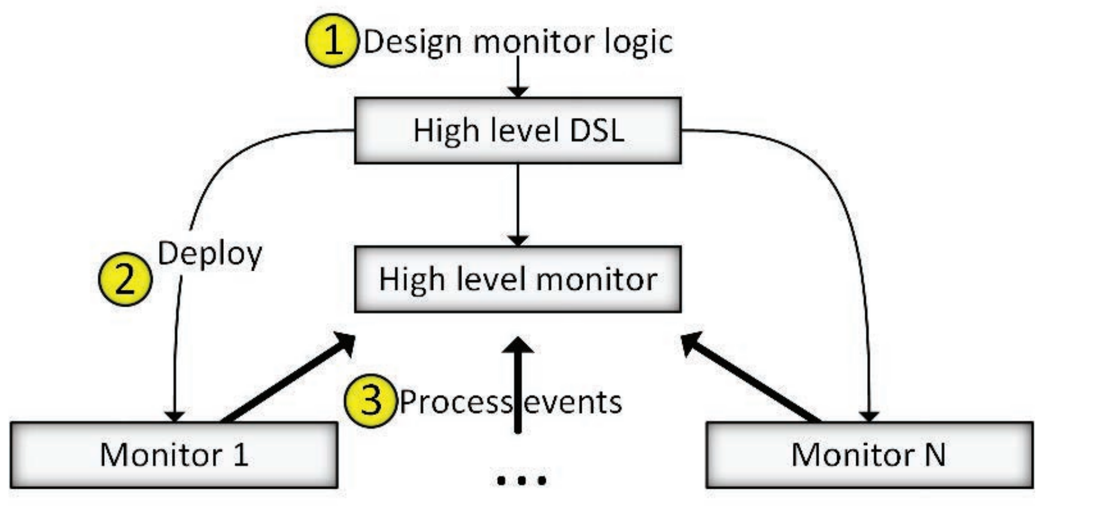
Event based semantics

- Events capture interactions rather than internal state
- Notion of causal partial order that reflects physical reality for events separated in space

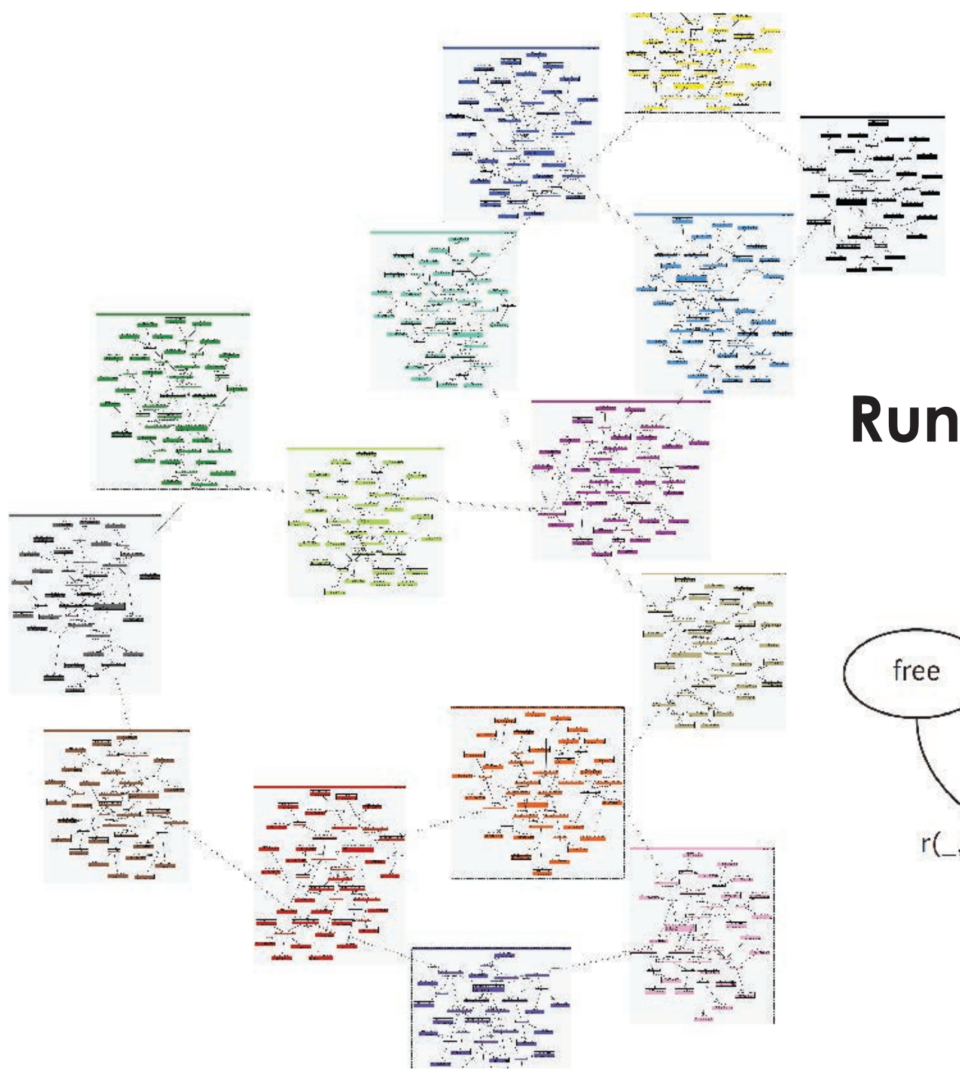
Complex Event Processing (CEP)

- Predict high-level events likely to result from specific sets of low-level factors
- Identifies and analyzes cause-and-effect relationships among events in real time
- Allow to proactively take effective actions in response to specific scenarios

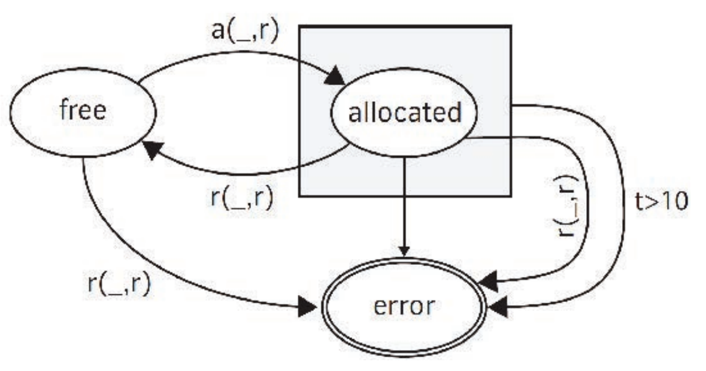




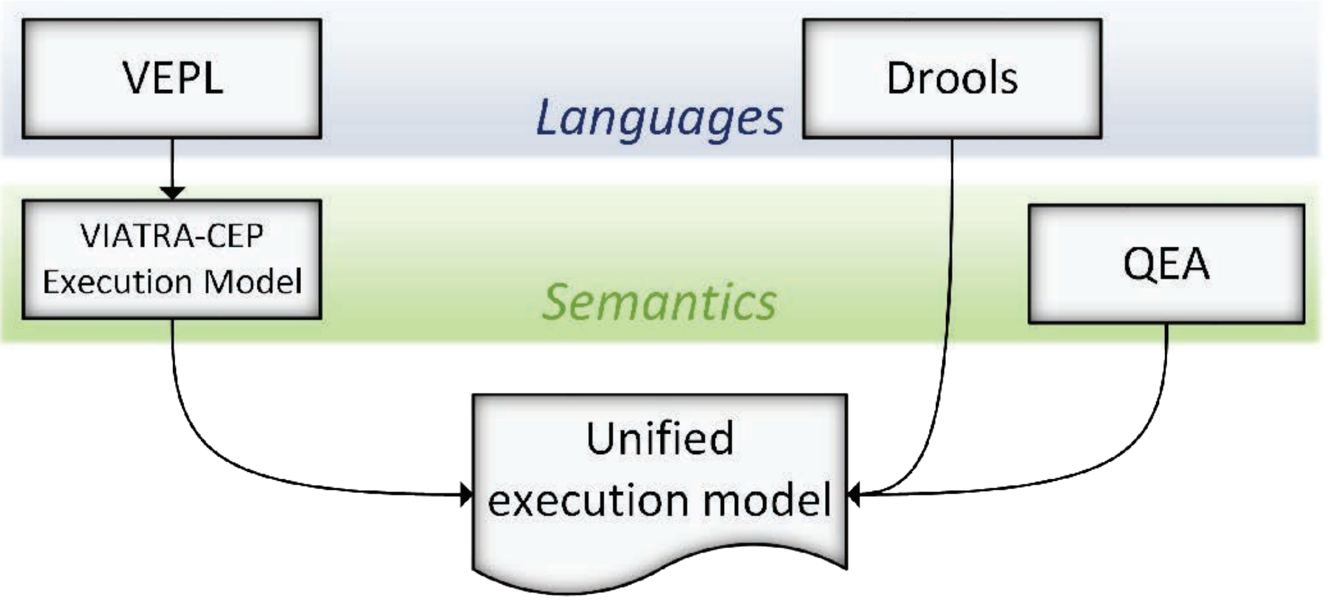
```
pattern trainReachUnsafe(t: Train){
  Train.nextGroup(t, ng);
  Group.safe(ng, safe);
  check(safe == false);
}or{
  Train.currentlyOn(t, co);
  Group.regions(g, co);

  Train.nextGroup(t, ng);
  PowerableGroup(ng);
  find regionNeighbor(ng, nng);
```



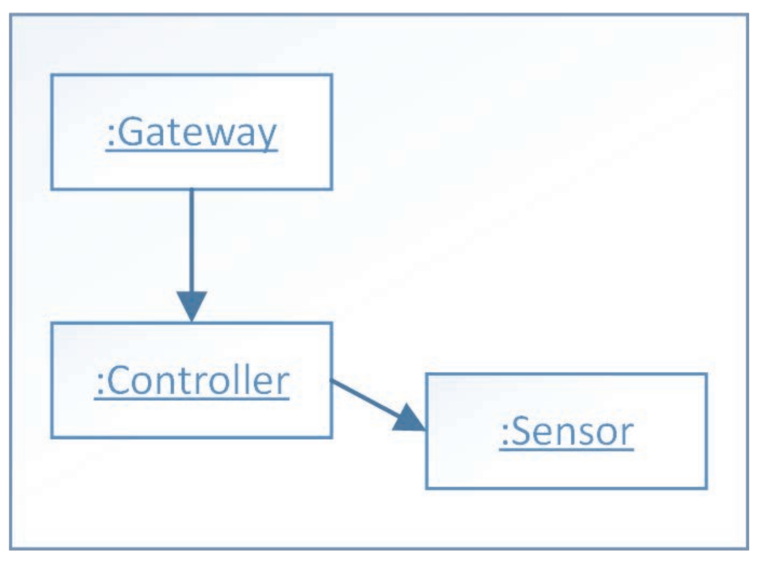
Run-time models



Semantic integration of distributed monitor behavior



Architecture/Platform modeling

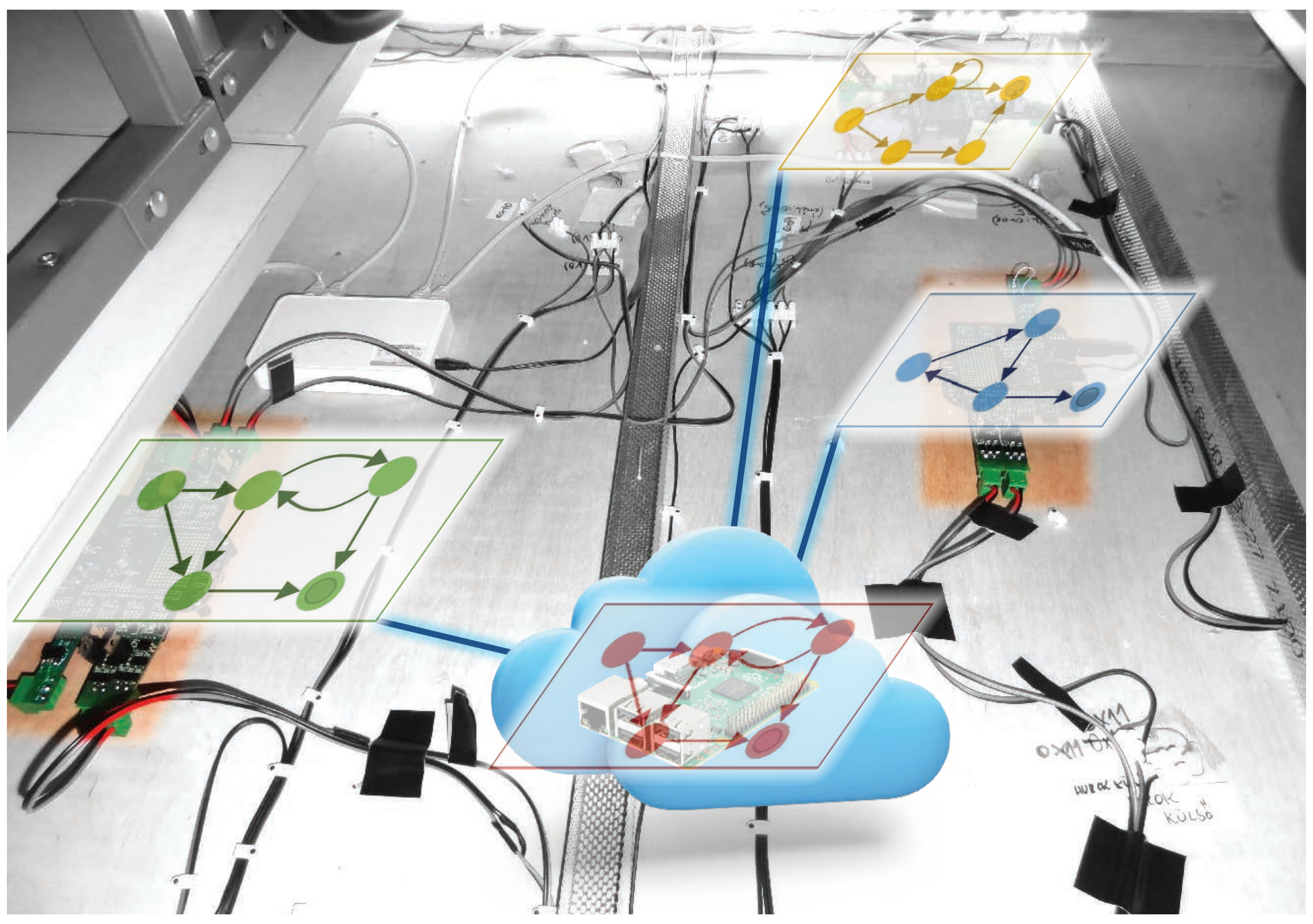


Monitor rule specification

```
@Deploy(controller1)
atomicEvent pressureHigh {
  as sensor1.pressure > 100
}

atomicEvent voltageHigh {
  as sensor2.voltage > 5.0
}
```

```
complexEvent safetyAlert {
  as(pressureHigh AND voltageHigh)
    [holdsFor 1500]
}
rule on safetyAlert {
  system.halt
  logErrorEvent
}
```

Automated generation of deployment configuration



Dávid, Ráth, Varró: **Foundations for Streaming Model Transformations by Complex Event Processing**, International Journal on Software and Systems Modeling, 2016
Ráth, Horváth: **IoT supercharged: complex event processing for MQTT with Eclipse technologies**, EclipseCon Europe 2015
Bergmann, Dávid, Hegedüs, Horváth, Ráth, Ujhelyi, Varró: **VIATRA3: A Reactive Model Transformation Platform**, Theory and Practice of Model Transformations, 2015

eclipse.org/viatra
wiki.eclipse.org/VIATRA/CEP

Universiteit Antwerpen
IncQueryLabs Ltd