Distributed and Heterogeneous Event-based Monitoring in Smart Cyber-Physical Systems

László Balogh¹, István Dávid², István Ráth³, Dániel Varró¹,⁴, András Vörös¹,⁴

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

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

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.

Challenges
Mixed-critical systems
- Safety-critical systems
- Internet-of-Things
- State-of-the-art
- Smart cities
- Traffic control
- Smart grid
- Fault tolerance
- Verification
- Constrained resources
- Low reliability
- Distributed
- Heterogeneous
- (Near) Real-time

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

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

Automated generation of deployment configuration

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)
  {HoldFor 10m}
}
rule on safetyAlert {
  system halt
  logErrorEvent
}