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

László Balogh, István Dávid, István Ráth, Dániel Varró and András Vörös

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group
Overview

- Smart cyber-physical systems
  - Motivation: the MoDeS3 case-study

- Complex-event processing with VIATRA-CEP

- Ongoing work
MODEL-BASED DEMONSTRATOR FOR SMART AND SAFE SYSTEMS
**Big Picture**

- **Traditional safety-critical systems:**
  - Model-based development
  - Validation & verification
  - Code generation
  - Safety requirements

- **Cyber-physical systems:**
  - Various information sources (sensors)
  - Heterogeneous: Embedded computers & cloud computing

---

**Combination of both worlds:**

Development techniques used for safety-critical systems with technologies from cyber-physical systems
Demonstrator
Demonstrator

Robot System:
- Moving/ removing objects
Demonstrator

Robot System:
- Moving/ removing objects

Railway System:
- Transportation
Demonstrator

Robot System:
- Moving/ removing objects

Railway System:
- Transportation

GOAL: safe (accident free) working of the system
<table>
<thead>
<tr>
<th>SW</th>
<th>Monitoring and Control System</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW</td>
<td>Robot system</td>
</tr>
<tr>
<td></td>
<td>Railway system with</td>
</tr>
<tr>
<td></td>
<td>- Sensors</td>
</tr>
<tr>
<td></td>
<td>- Actuators</td>
</tr>
<tr>
<td>SW</td>
<td>Distributed Safety Logic</td>
</tr>
</tbody>
</table>
### MoDeS3

<table>
<thead>
<tr>
<th><strong>HW</strong></th>
<th><strong>SW</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Railway system with - Sensors - Actuators</td>
<td>Monitoring and Control System</td>
</tr>
<tr>
<td><strong>SW</strong></td>
<td>Distributed Safety Logic</td>
</tr>
</tbody>
</table>
Distributed Safety Logic

15 sensors:
- Sensing the trains and estimating their locations
Distributed Safety Logic

6 embedded controllers:
- Actuators
Distributed Safety Logic

6 embedded controllers:
- Actuators
Distributed Safety Logic

- Distributed:
  - 6 controllers
  - Communication
- Safety: prevent accidents by stopping the trains
Distributed Safety Logic

Model-driven development
- Validation techniques
  - VIATRA Query
- Verification techniques
  - Model-transformation
  - VIATRA
- Code generation

YAKINDU Statechart Tools

Code generation
Distributed Safety Logic

Model-driven development
- Validation techniques
  - VIATRA Query
- Verification techniques
  - Model-transformation
  - VIATRA
- Code generation

YAKINDU Statechart Tools

Each track section is controlled by a dedicated BBB

Code generation
Distributed Safety Logic

- Model-driven development
  - Validation techniques
  - VIATRA Query
  - Verification techniques
  - Model-transformation
  - VIATRA
  - Code generation

- IoT technologies for communication
- MQTT
  - Eclipse Paho
  - Mosquitto

YAKINDU Statechart Tools

Code generation
Component Level Runtime Verification

- Formal specification language: statechart
  - Hierarchical
  - Timed
  - Parametric
- Runtime monitor generation
- Formal semantics
  - Analysis

Formal model

C++ Monitor
MoDeS3

<table>
<thead>
<tr>
<th>HW</th>
<th>Monitoring and Control System</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW</td>
<td>Robot system</td>
</tr>
<tr>
<td>SW</td>
<td>Railway system with</td>
</tr>
<tr>
<td></td>
<td>- Sensors</td>
</tr>
<tr>
<td></td>
<td>- Actuators</td>
</tr>
<tr>
<td>SW</td>
<td>Distributed Safety Logic</td>
</tr>
</tbody>
</table>
Additional level of safety – high level monitoring
Monitoring and Control System

- Additional level of safety – high level monitoring

Computer vision
Monitoring and Control System

- Additional level of safety – high level monitoring

![Diagram](image_url)

- Computer vision
- Communication
- Monitoring

Data stream
Monitoring and Control System

- Additional level of safety – high level monitoring

![OpenCV](OpenCV.png)

Computer vision

![MQTT](MQTT.png)

Communication

![Monitoring](Monitoring.png)

Monitoring

Complex Event Processing
### Monitoring and Control System

- **Additional level of safety – high level monitoring**

![Diagram showing computer vision, communication, and monitoring systems with VEPL and VIATRA-CEP programming examples]
Monitoring and Control System

- Additional level of safety – high level monitoring

- Shut down the system in case of dangerous situation
Monitoring and Control System

- Additional level of safety – high level monitoring

Shut down the system in case of dangerous situation

Monitoring logic
Monitoring and Control System

- Additional level of safety – high level monitoring

Shut down the system in case of dangerous situation

VIATRA-CEP

VIPL

Computer vision

Communication

Monitoring

Execution

Monitoring logic

Shut down the system in case of dangerous situation
Monitoring and Control System

- Additional level of safety – high level monitoring

- OpenCV
  - Computer vision

- MQTT.org
  - Communication

- Monitoring
Monitoring and Control System

- Additional level of safety – high level monitoring

OpenCV

Computer vision

Communication

Monitoring

Node-RED

Cloud

docker

DigitalOcean

IBM Bluemix
MoDeS3

<table>
<thead>
<tr>
<th>SW</th>
<th>Monitoring and Control System</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW</td>
<td>Robot system</td>
</tr>
<tr>
<td></td>
<td>Railway system with</td>
</tr>
<tr>
<td></td>
<td>- Sensors</td>
</tr>
<tr>
<td></td>
<td>- Actuators</td>
</tr>
<tr>
<td>SW</td>
<td>Distributed Safety Logic</td>
</tr>
</tbody>
</table>
Robot System

- **Goal:** Moving/removing objects from the trains
  - Place onto other train/place onto the ground

- **Control logic**
- **Sensors**
- **Feedback**
- **Code generation**
- **Computer vision**
Heterogeneous platform

Cloud infrastructure

Cloud

Infinite resources

Personal computer

Application processor

Real-time unit

PC

Fast response
Technologies

- Node-RED
- Mosquitto
- VIATRA-CEP
- OpenCV
- OpenModelica
- Maven
- MQTT
- Xtend
- Xtend
- Debian
- Paho
- Docker
Summary

- Goal: case-study for smart CPS
- Combine various techniques from the domains of
  - Cyber-physical systems
  - Safety-critical systems
SYSTEM LEVEL MONITORING FRAMEWORK: VIATRA-CEP
System Level Runtime Verification

- VIATRA - CEP

Abstraction of the system → Events → Processing
System Level Runtime Verification

- VIATRA - CEP

Abstraction of the system → Events → Processing
System Level Runtime Verification

- **VIATRA - CEP**

EMF metamodel:
- Elements and possible relations/connections
System Level Runtime Verification

- VIATRA - CEP

Instance model → Events → Processing
System Level Runtime Verification

- VIATRA - CEP

Graph pattern matching

Events ➔ Processing
System Level Runtime Verification

- VIATRA - CEP

Events are generated when a specific graph pattern appears...
System Level Runtime Verification

- VIATRA - CEP

Events → Processing

Automaton „consumes“ the events
Investigation of languages

- Parametric Timed Regular Expression
- Parametric Timed Event Automaton
  - Based on Parametric Event Automaton
- Example:
  - Two trains should not enter the same section
  - \(\text{enter}(t_1,s) \rightarrow \text{NOT}(\text{exit}(t_1,s)){\ast} \rightarrow \text{enter}(t_2,s)\)
Investigation of languages

- Parametric Timed Regular Expression
- Parametric Timed Event Automaton
  - Based on Parametric Event Automaton
- Questions:
  - Timed-automaton determinization
    - Needed to run the monitor on embedded devices
Future Goals

Engineering languages

Common formal intermediate representation

Runtime verification components

Embedded (C++) monitor

Rule System

enter(t_1,s) -> NOT (exit(t_1,s))

-> enter(t_2,s)
Automated deployment

Platform model

Runtime specification (high level language)

Cloud

PC
Summary

- CPS demonstrator: MoDeS3

- VIATRA – CEP: ongoing developments
  - Development of the automaton formalism
  - Determinization
  - Automatic deployment/monitor synthesis
Acknowledgements

- **Application in research:**
  - Fault Tolerant Systems Research Group
  - MTA-BME Research Group on Cyber-Physical Systems
  - Reconfigurable ROS-based Resilient Reasoning Robotic Cooperating Systems (R5-COP)

- **Industrial sponsors:**
  - IncQuery Labs Ltd.
  - Quanopt Ltd.
  - Ericsson