Using Environmental Contexts to Model Restrictions on Sensor Capabilities

Martin Richter    C. Jakobs    T. Werner    M. Werner

Chemnitz University of Technology (TUC)
martin.richter@informatik.tu-chemnitz.de
Short Resume

- Since March 2020 research assistant at the Operating Systems Group of the TUC
- 2016–2020 tutor for research and teaching
- 2013–2020 student of Applied Informatics and Automotive Software Engineering

Research Focus

Programming models for heterogeneous CPS to enable I/O virtualization
CPS

- Connect logical and physical world
- Multitude of heterogeneous sensors and actuators
- Observing through sensors
- Influencing through actuators

Digital World

Physical World

Sensor Measurements → Actuator Control

0100 0100 0101 0101
1001 1001 1011 1000
0110 1110 0010 1011
Challenges

- Distributed, heterogeneous, mobile, unreliable devices
- Different sensors/actuators may join/leave the system
- Currently: devices programmed individually
- Prone to error and complex
- We need **abstractions**!
Abstractions

- Systemic view, devices programmed as a whole
- Changing devices due to motion/failure
- Distribution, location, motion transparency on application level
- Require sensor and actuator virtualization
Virtualization is the utilization of a logical resource that is mapped onto possibly multiple physical resources at access/on demand.

- Virtual resources $v_i$ managed by runtime environment
- Transparent **exchange** of physical resources $p_j$
- Transparent **sharing** of physical resources for applications $a_k$

⇒ Detachment of applications and physical resources
Sensor Virtualization

- Focus on transparent exchange of physical sensors
- Programmer not directly involved in managing sensors
- Changing sets of devices due to unreliability or mobility transparently handled
- Developer specifies **what** should be measured
- RTE infers **how** it is measured
Programming Model: Physical Object Specifications

- Vector of object properties

\[ \vec{z} = \left[ (\tau_1, r_1), \ldots, (\tau_n, r_n) \right] \]

- \( \tau_j \) is type of property (i.e., domain of possible values)

- \( r_j \) is rule on values of type \( \tau_j \) (i.e., does value fit object’s property?)

Which sensors should be utilized?

```
Fire {
    Color: = red;
    Temp: > 300;
}
```

Runtime Environment

Physical Environment
Sensor Model

- Sensors measure physical quantity
- Transform measurement into digital signal via measurement process
- Grouped into classes based on measurand and measurement process
- Output may **not** directly map to object property types
Interpretation Methods

- Transform digital output of sensors into value of physical property type
- RTE chooses methods based on this type
- Each method requires output of sensors of certain classes
- These outputs have to be related

![Diagram showing the process of interpretation methods from physical environment to object property value.]

- CMOS Sensor, A/D-Converter
- Image Recognition
  - Color
  - Shape
- Physical Environment
  - $s_1$
  - $s_2$
  - ... 
  - $s_n$
- Interpretation Method
  - Object Property Value
Spatial Sensor Capabilities

- Sensor measurements are valid for region of space
- Depends on their location, surroundings and sensor-specific parameters

How to describe the influence of surroundings on sensors?
Environmental Context Model

- Restrict regions for which sensor measurements are valid
- Context $c$ is defined by:
  - Region of space (set of locations): $c.X$
  - Influenced physical quantity (e.g., electromagnetic radiation): $c.q$
  - Inward- and/or outward-blocking (e.g., tinted windows): $c.\neg out$, $c.\neg in$

\[
c.\neg out = 1 \\
c.\neg in = 0
\]
Using Environmental Contexts to Model Restrictions on Sensor Capabilities

Context Model

Influence of Contexts on Sensors

- Measurable region intersects with context’s region
- Similar physical quantity observed/influenced

\[
c.\text{out} = 1 \\
c.\text{in} = 0
\]

Choice of Sensors

Only the measurements of sensors with intersecting observed regions can be chosen as inputs for an interpretation method.
Recap Sensor and Context Models

- Programmer provides object property description $\vec{z}$
- RTE deduces possible interpretation methods $\vec{m}_{\tau_i}$ for each object property $z_i$ of type $\tau_i$
- Choice of sensors based on required classes and whether measurement regions overlap

$$\vec{z} = \begin{bmatrix} (\tau_1, r_1) \\ \vdots \\ (\tau_p, r_p) \end{bmatrix}$$

Diagram:

- Physical Environment
- $\vec{m}_{\tau_1}$, $\vec{m}_{\tau_2}$, ..., $\vec{m}_{\tau_p}$
- $s_1$, $s_2$, ..., $s_n$
Physical Object Identification

- Result of interpretation method is also valid for region of space
- Intersection of observed regions of input sensor measurements
- For checking whether all properties are present, regions of chosen interpretation methods have to overlap
- Object is present at locations where all rules are satisfied

\[
\begin{bmatrix}
  r_1(m_1) \\
  r_2(m_2) \\
  r_3(m_3)
\end{bmatrix}
= \begin{bmatrix}
  ? \\
  ? \\
  ?
\end{bmatrix}
\]
Conclusion

- Presented sensor and environmental context models
- Allow to describe restriction of sensor capabilities based on environmental contexts
- Enables RTE to continuously choose sufficient sets of sensors to observe physical object
- Sensor virtualization introduced
Using Environmental Contexts to Model Restrictions on Sensor Capabilities

Future Work

- Metrics for optimal choice of sensors for measuring object properties
- Efficient data structures for model implementation
- Similar models for actuators to enable actuator virtualization
Thank you!