Composing Learning Internet-of-Things Applications with Accessors

Armin Wasicek, Ilge Akkaya, Edward Lee, Chris Shaver
UC Berkeley
Ptolemy Minicon
10/12/2015
Machine learning: adding impact to IoT

Making Sense of IoT Data With Machine Learning Technologies
Added Value

• **Information excellence**
  – Optimize processes and asset utilization.

• **Solution leadership**
  – Without the ML service, no product

• **Data fusion**
  – Aggregates data

• **Accelerated innovation**
  – Open innovation
Ingredients of Learning IoT apps

• Physical components
  – Engine, wheels, etc.

• Smart components
  – Increase the value
  – Collect, organize, and interpret information

• Connectivity
  – Enables aggregations like ‘swarms of things’
Challenges

How to connect these different paradigms?

• Accessors enable interoperability
• Wrap sensors, actuators, and services
• Export an actor interface

▶ Uniform interface specification language
Accessors and Actors

Horizontal and vertical design contracts

- Actor composition, same level of abstraction
- Wrapping connects abstraction levels

► Modeling environments support designers to get to correct and stable solutions.

16/10/15
Armin Wasicek
How’s my driving?

Driver -> Actual Behavior

Driver exhibits Actual Behavior

Recognized Behavior -> Online Learning

Recognized Behavior reports

Online Learning classifies

Online Learning monitors Actual Behavior
Elements of the Solution

• Discrete Event (DE) director
  – baseline Model of Computation (MoC)

• PILOT, a machine learning toolkit
  – designed to work with streaming interfaces.

• Lattice-based ontology package
  – reducing the need for explicit type annotation
Driving Behavior Swarmlet
Machine Learning Pipeline
Contract Analysis

- Using LatticeOntologySolver to verify horizontal contract

Constraint: CsvToJson.json $\geq$ Speed
Constraint2: DriverBehaviorClassification.data $\geq$ Speed
Conclusion

• Learning will be part of most IoT applications

• Learning IoT applications are a combination of data sources and processing systems.

• Pulling together Accessors and actors from the ML package makes building smart systems very accessible

• **Accessors reduce complexity** through hiding implementation details

• Analyzing horizontal and vertical design contract support designers
Thanks for your attention!