1 Application
3 Dialects
5+ Architectures

Platform independence with BridgePoint

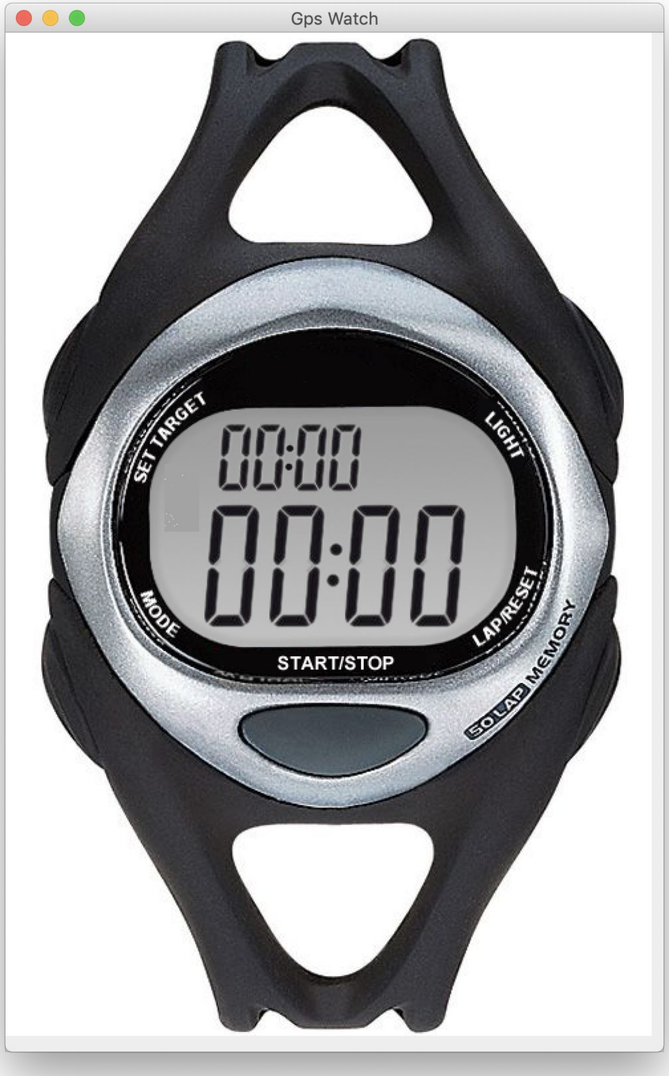
October 2018
Levi Starrett

Goals

- Demonstrate platform independent modeling using BridgePoint
- Provide background on the modeling dialects supported by BridgePoint
- Discuss some of the differences between the supported dialects and architectures

1 Application

- Simulation of a GPS running watch
- Keeps time and distance of a simulated "jog"
  - Maintains lap times; can clear and reset
  - Toggles through multiple display modes



3 Dialects

- 1. xtUML
  - Formal specification of the Shlaer-Mellor method of MDA
  - Specified in an xtUML meta-model
  - First introduced by Project Technology, then Mentor Graphics, now the open source community (xtuml.org)
  - Specifies graphical and semantic rules for models
  - Executable and translatable
  - Closely associated with the BridgePoint MDA tool

3 Dialects

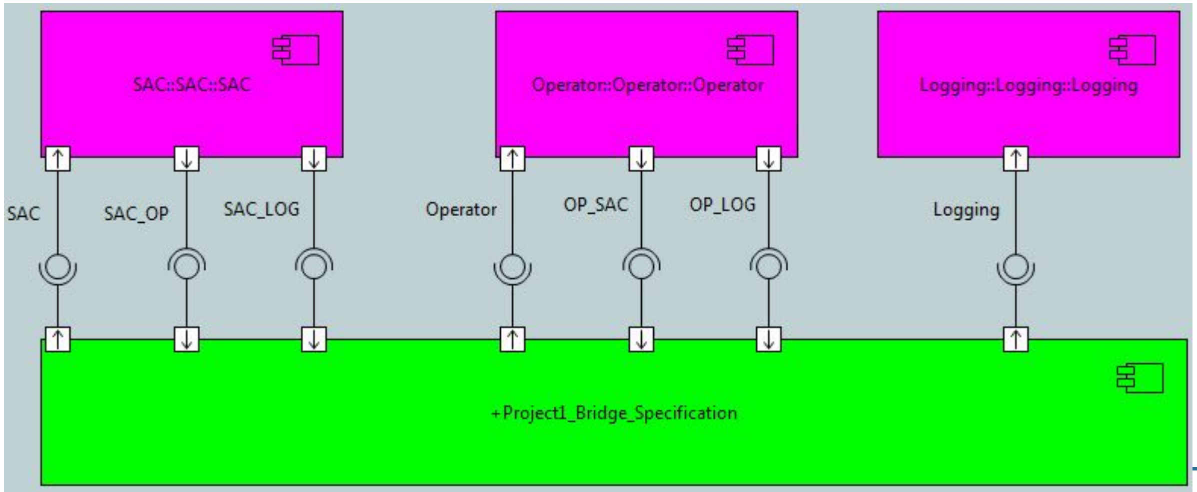
- 2. OAL
  - Object Action Language
  - Specifies rules for processing of data in xtUML models
  - Specified in the xtUML meta-model and a BNF grammar
  - First introduced by Project Technology, then Mentor Graphics, now the open source community (xtuml.org)
  - Default action language for xtUML
  - Interpretable by Verifier in BridgePoint

3 Dialects

- 3. MASL
  - Model and Action Specification Language
  - Formal specification of the Kennedy-Carter method of MDA (which has its roots in S-M)
  - Created with inspiration drawn from iUML, ASL, and Ada with added syntax for “structural” model elements
  - No graphics
  - Translatable by a parser and model compiler
  - Supported by BridgePoint in xtUML models (action language only)

xtUML/OAL and MASL compatibility

- Additional tool support
  - xtUML to MASL model compiler (exporter)
  - MASL to xtUML parser/model-to-model translator
- Mapping techniques
  - Idiomatic modeling constraints
  - Line by line action language mapping
  - Tool support for differences (e.g. polymorphic events)


SAC::SAC::SAC
Operator::Operator::Operator
Logging::Logging::Logging
SAC
SAC_OP
SAC_LOG
Operator
OP_SAC
OP_LOG
Logging
+Project1_Bridge_Specification

5+ Architectures

- 1. BridgePoint Verifier (interpreted simulation)
- 2. MC-3020 C binary on Windows
- 3. MC-3020 C binary on macOS
- 4. MC-3020 C binary on Linux
- 5. MASL C++ binary on Linux
- Bonus!
  - MC-3020 C binary on Arduino
  - MASL C++ binary on Raspberry Pi (Raspbian)

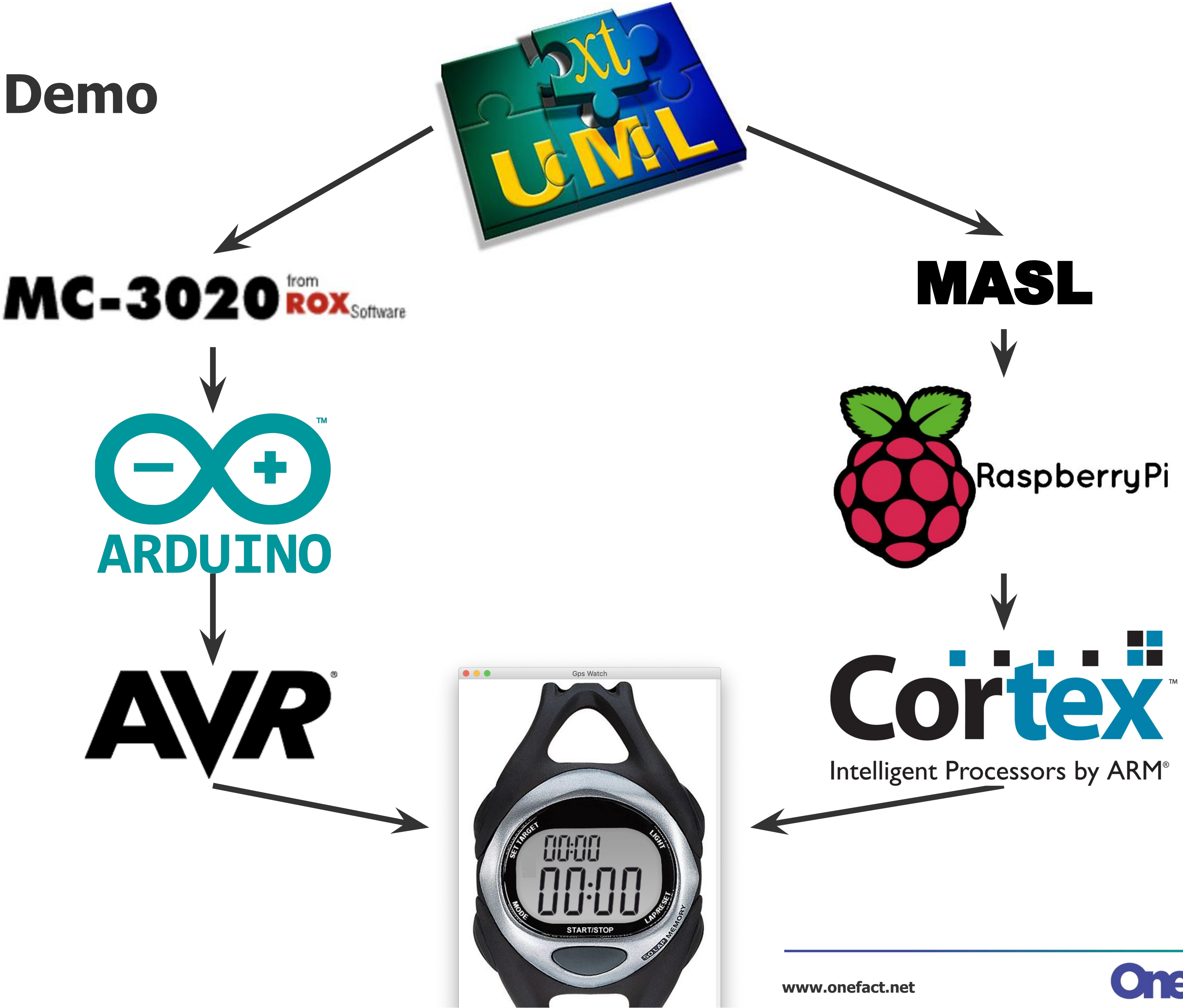
5+ Architectures

- Arduino Uno
  - ATmega328 microcontroller
  - 32 KB flash storage
  - 2048 bytes dynamic memory
  - 8-bit AVR architecture
- Usage by GPS Watch
  - 1593 bytes used for global data
  - 455 bytes available for the stack
- Handling memory constraints with MC-3020
  - Limit instance extent sizes
  - Identify singleton classes
  - Avoid using strings (or minimize string size)
  - Bit fields, type precision, more...

5+ Architectures

- Raspberry Pi 3 Model B v1.2
  - ARM Cortex-A7 CPU
  - 32 GB flash storage (microSD)
  - 1 GB dynamic memory
  - 32-bit ARM architecture
  - Runs Linux (Raspbian)
- Usage by GPS Watch
  - ~ 50KB peak memory used in one minute run
- MASL architecture design goals
  - Stability
  - Distributed systems
  - Dynamic loading and linking of independent modules
  - Persistent instance data

Demo



xtUML

MC-3020 from ROX Software

ARDUINO™

AVR®

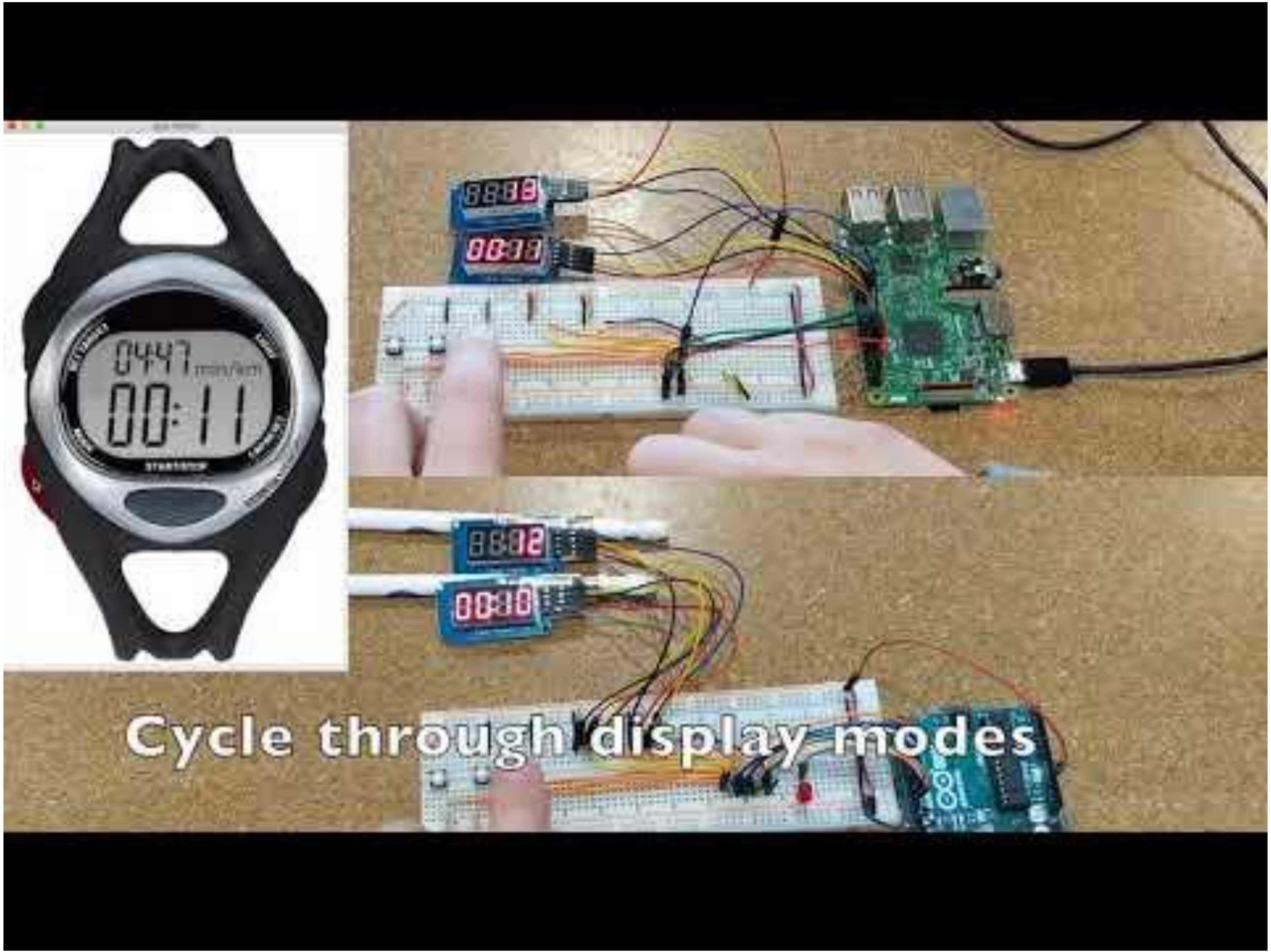
MASL

RaspberryPi

Cortex™

Intelligent Processors by ARM®

Demo



Questions?

OneFact

ONE FACT, INC.

onefact.net