Using Co-Simulation for Model Reuse and Experiment Reproducibility
A co-simulation is a simulation with multiple subsystems that are executed simultaneously and exchange data at runtime.

Co-Simulation of a Distribution Grid

Certain commercial products are identified to adequately explain the concept; this does not imply endorsement or recommendation by NIST, nor does it imply that such products are necessarily the best available for the purpose.
Applications of Co-Simulation

- Human-in-the-loop training
- Model reuse across simulators
- Shared or unique physical resources
- Simulation of complex systems (CPS/IoT)
  - Often safety-critical systems
  - Often cannot run experiments on the live systems
  - Often require a combination of expertise to understand
Approaches to Co-Simulation

**ad hoc**
- MATLAB S-Functions, TCP/IP Sockets, ...

**frameworks**
- HELICS, mosaik, ...

**standards**
- High Level Architecture (HLA)
- Functional Mock-up Interface (FMI)
- ...

The IEEE **High Level Architecture (HLA)** is a co-simulation standard defining the services a set of **federates** can use in a **federation**.
A wrapper is software that defines the method of *time synchronization* and *data exchange* used for a simulator in a federation.

A simulator with a wrapper can be *re-used* in multiple federations:
Federate Interoperability

**Connectivity**
- Able to share data (protocols)

**Semantic Interoperability**
- Able to *understand* shared data (data models)

‘Functional’ Interoperability
- Able to *effectively use* shared data
Distributed as a virtual machine
Contains a graphical experiment and federate design environment
Uses code generation to turn models into executable code

available at https://github.com/usnistgov/ucef
In UCEF a graphical language is used to design federates:

The federates modeled in this language can be transformed into executable code / simulation models using code generation.
UCEF: Portable HLA Development Kit

mix & match → WebGME → generates HLA RTI

UCEF Virtual Machine

Grid → power
Auction → price
Building → temperature
Federate Interoperability

Connectivity
- Able to share data (protocols)

Semantic Interoperability
- Able to *understand* shared data (data models)

‘Functional’ Interoperability
- Able to *effectively use* shared data
Federation Object Model (FOM)

Simulation Object Model (SOM)

Federation

Federate → Federate → Federate

Runtime Infrastructure (RTI)

• Examples:
  • SISO-STD-001.1-2015
    Real-time Platform Reference Federation Object Model (RPR FOM)
  • SISO-STD-018-00-2020
    Space Reference Federation Object Model (SpaceFOM)
AV Functions include:

- Traffic Modeling
- Operating Envelope Specification (OES)
- Perception and V2X Sensor Modeling
- Localization Modeling
- Cameras and Computer Vision
- ADS (Brain)
- Vehicle Physics Engine Modeling

Automated Driving System (ADS) Enabled Dynamic Driving Task (DDT) Execution

Testbed Functions

- Hi-Res Map
- Path Plan (RINX, drive cycle)
- Trip Manager (initial route, object/event plan)
- Map Sets
- Path Plans
- Trip Mgr.

Operating Envelope Specification (OES)

Vehicle Functions

- Phys Eng
- Time Locn
- Sensors
- Comms
Connectivity
- Able to share data (protocols)

Semantic Interoperability
- Able to understand shared data (data models)

‘Functional’ Interoperability
- Able to effectively use shared data
Cyber-Physical Systems (CPS)
CPS Framework - Property Trees

- Trustworthiness
  - Privacy
  - Reliability
  - Resilience
  - Safety
  - Security
    - Confidentiality
    - Integrity
    - Availability

\[ P_1 \sim \ldots \sim P_n \]
Generation of Property Trees

CPS Stakeholders (Business, Societal & Technical) → Raw CPS concerns → Aspects and Concerns → CPS Framework

- Conceptualization Facet: Model of CPS
- Realization Facet: Instance of CPS
- Assurance Facet: CPS Assurance

Assured CPS

What things should be and what things are supposed to do

How things should be made and be operated

How to prove things actually work the way they should
Composition of CPS

Will these systems work together?

CPS_1.Trustworthiness.Security.Confidentiality.P1
+ CPS_2.Trustworthiness.Security.Confidentiality.P1
= VALID SYSTEM
CPS Descriptor
Federate Metadata

Simulation Object Model (SOM)
<SOM>
  <CPSFramework>
  ...
</SOM>

Federate

Simulation Object Model (SOM)
<SOM>
  <CPSFramework>
  ...
</SOM>

Federate

Federation Object Model (FOM)
<FOM>
  <CPSFramework>
  <Shared Object Classes>
  <Shared Interaction Classes>
  ...
</FOM>

Runtime Infrastructure (RTI)

Federate Library
Questions

Thomas Roth
NIST Communications Technology Laboratory
IoT Devices and Infrastructures Group
thomas.roth@nist.gov