Building Distributed Co-simulations using CoHLA

Thomas Nägele\textsuperscript{a}, Jozef Hooman\textsuperscript{a,b}, Jack Sleuters\textsuperscript{b}
\textsuperscript{a}Radboud University, \textsuperscript{b}ESI (TNO), The Netherlands

CoHLA: Configuring HLA

- Goal: support concurrent multi-disciplinary development of Cyber-Physical Systems (CPSs)
- High Level Architecture (HLA) standard is used for co-simulation execution
- Domain Specific Language (DSL) to specify co-simulation of different types of models on HLA
- Generates runnable co-simulation code for use with OpenRTI
- Supported model formats:
  - Functional Mock-up Interface standard (FMI)
  - Parallel Object-Oriented Specification Language (POOSL)
- Allows addition of loggers, actors and basic metric processors

Example: Smart lighting system

- Smart lighting system for an office building
- Building consists of different types of areas
- Three types of models
  - Lighting controllers: discrete-time, POOSL
  - Lights: continuous-time, FMU
  - Occupancy sensors: discrete-time, FMU

Implementation using CoHLA

```coala
confederation VerySmallBuilding {
  Instances |
  Instance r1Controller as BasicRoomController
  Instance r1Light as DimmableLight
  Instance r1Sensor as OccupancySensor
  Instance r2Controller as BasicRoomController
  Instance r2Light as DimmableLight
  Instance r2Sensor as OccupancySensor
  Instance corridorController as CorridorController
  Instance corridorLight1 as DimmableLight
  Instance corridorSensor1 as OccupancySensor
}

Connections |
Connection { r1Light.setValue <- r1Controller.setValue }
Connection { r2Light.setValue <- r2Controller.setValue }
Connection { corridorLight1.setValue <- corridorController.setValue }
Connection { r1Controller.occupied <- r1Sensor.occupied }
Connection { r2Controller.occupied <- r2Sensor.occupied }
Connection { corridorController.relatedActivity <- corridorSensor1.occupied }
Connection { corridorController.relatedActivity <- corridorSensor1.occupied }
```

Listing 1: Definition of a small building co-simulation in CoHLA.

- Scalability is difficult: buildings quickly become very large
  - CoHLA definition also becomes very large
  - Simulation time increases rapidly

Distributed simulation

- Options for distributed simulation execution added to CoHLA
  - Automatic, without configuration
  - Manual, simulations assigned to computation nodes
    - Generated by Lighting DSL: group connected simulations
- Distribution as supported by OpenRTI
- Sample building: 36 lights, 38 sensors, 14 controllers
- Distribute over up to 7 computation nodes in the cloud

Conclusion

- CoHLA allows fast construction of co-simulations
- Development of a separate DSL saves time
- Distributed simulation improves simulation speed