Traceability in the Model-based Design of Cyber-Physical Systems

Christian König, Alachew Mengist, Carl Gamble, Jos Höll, Kenneth Lausdahl, Tom Bokhove, Etienne Brosse, Oliver Möller, Adrian Pop

Alachew Mengist

The 2nd American Modelica conference

September 22-24, 2020

LINKÖPING UNIVERSITY MODELIOSOFT AARHUS UNIVERSITET TWT GmbH Science & Innovation controllab Newcastle University Verified

Motivation

- Today's products design are getting more and more complex
- Different engineering disciplines need to work together
- Frequent changes in design requirements

- The different roles in the engineering process are using highly specialized tools
  - Different vendors
  - Open source projects
  - In-house development

Motivation

- **Current state of the art**
  - No whole-life-cycle tool support
  - Weak or no integration between tools
  - Minimal or no traceability between artifacts
- **Challenges for product design tools**
  - Difficult to manage change and variability of design
  - Difficult to link designs to other lifecycle artifacts
  - Difficult to trace and analyze the impact of design changes

Traceability

- Creation and the use of links (or connections)

- Makes it possible to ensure that the requirements are met
- Enables analysis on models such as coverage, impact analysis
- Facilitates documentation of system development (model evolution documentation)

Challenges of Traceability Management

- Development by distributed teams
- Artifacts and links undergo constant change
- Involve multiple stakeholders with different background
- Trace links often manually created or not created at all

- Existing traceability approaches are either limited to
  - A specific domain and problem, or
  - Lack proper specification of traceability link semantics

Key Contribution

- Developing an OSLC tool-chain data integration architecture to link artifacts from different lifecycle tools

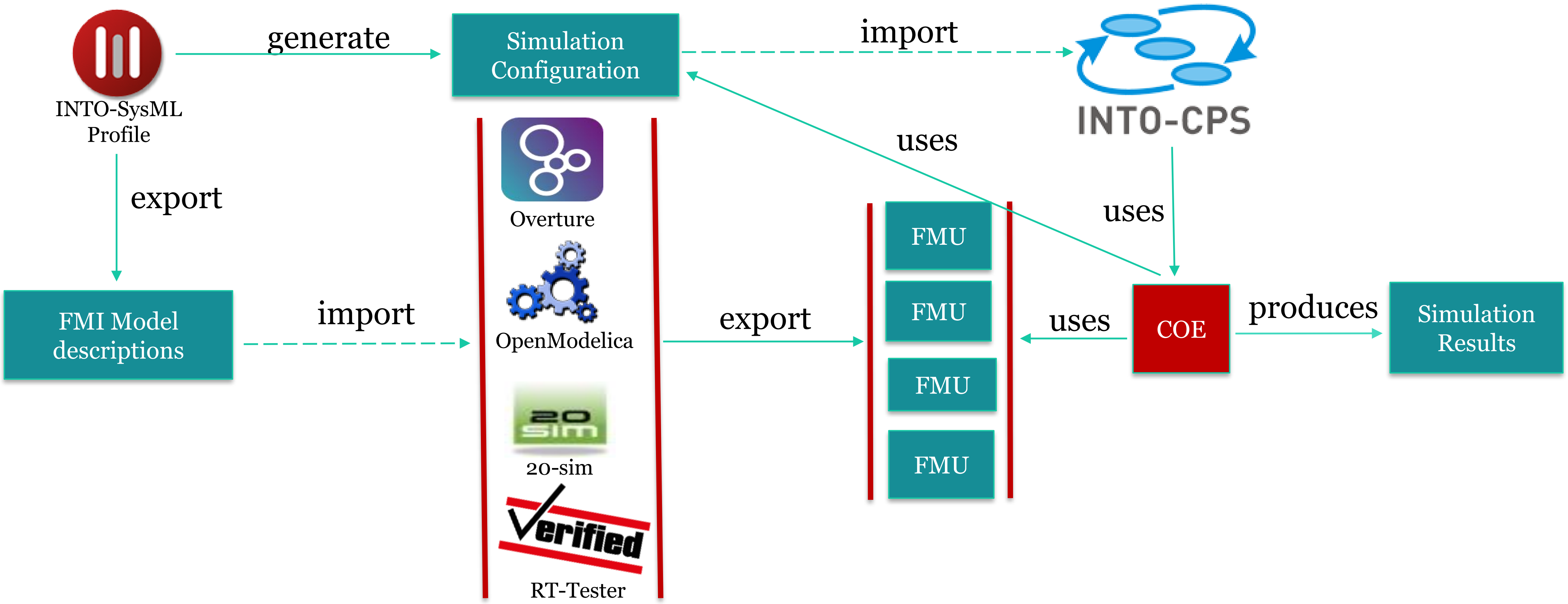
- Goal
  - Identify integration scenario for tool integration
  - Develop traceability information model (TIM)
  - Develop a prototype to support that integration scenario (Proof of Concept)
  - Test the prototype with cross domain lifecycle tools

Open Services for Lifecycle Collaboration (OSLC)

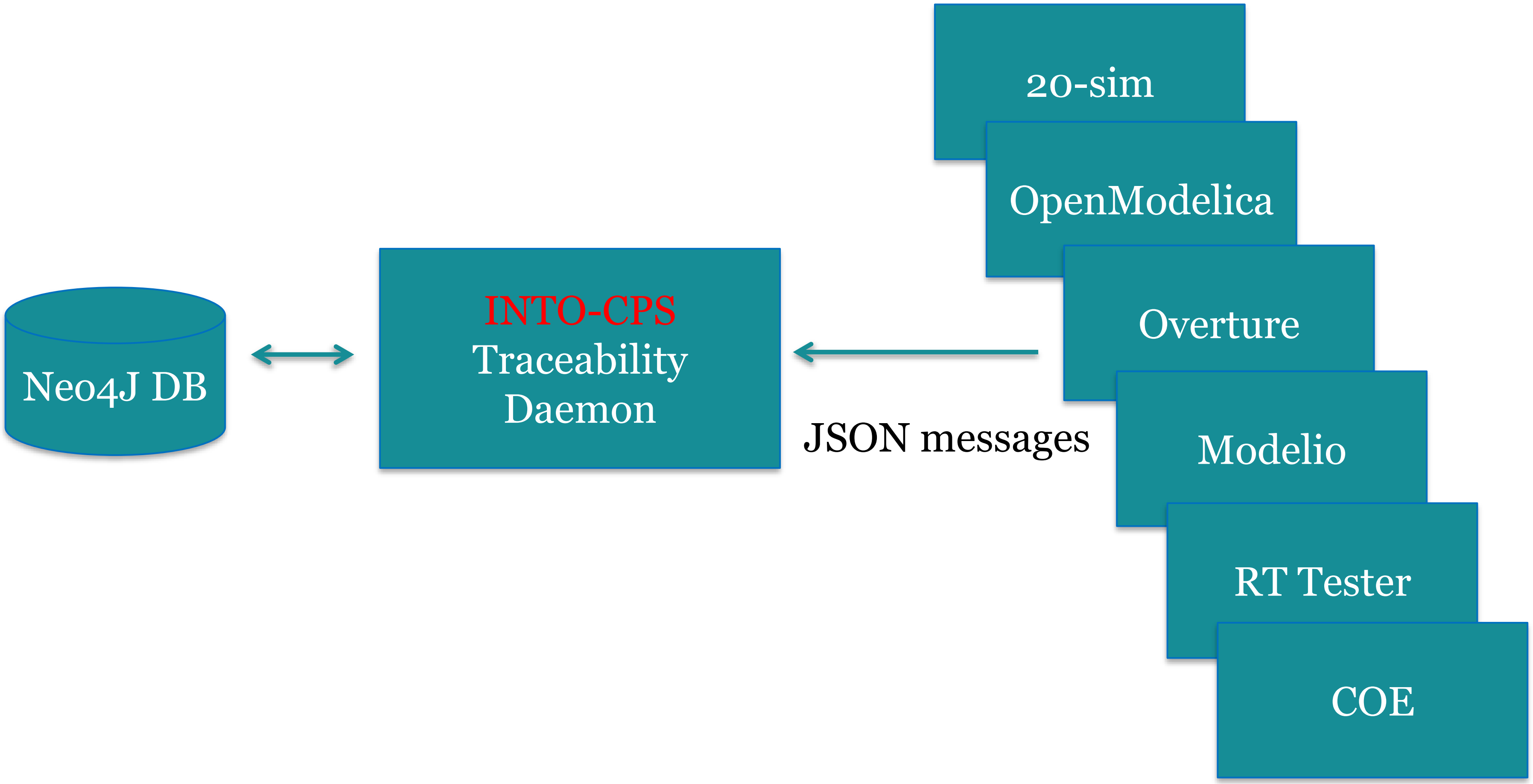
- Lifecycle integration based on Web standards
  - Linked Data and
  - RESTful Web Services

- Minimal web-based API allowing workflow integration
- Linked Data approach
  - Built on Resource Description Framework (RDF)
  - Use URIs (Uniform Resource Identifier) as identifiers
  - Use HTTP URIs so that people can discover including links to other URIs

Integration Scenario - INTO-CPS Tool Chain



Traceability Design and Architecture



Traceability Message Format and Ontology

- RDF triples (source, relations, target)
- Generate URIs for RDF Graphs
- JSON supported

→ "Simulation results (source) validates (relations) a requirement (target)"

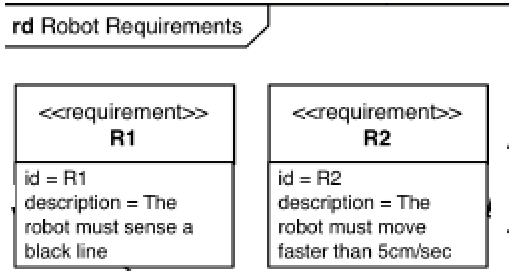
"A requirement satisfies model design "

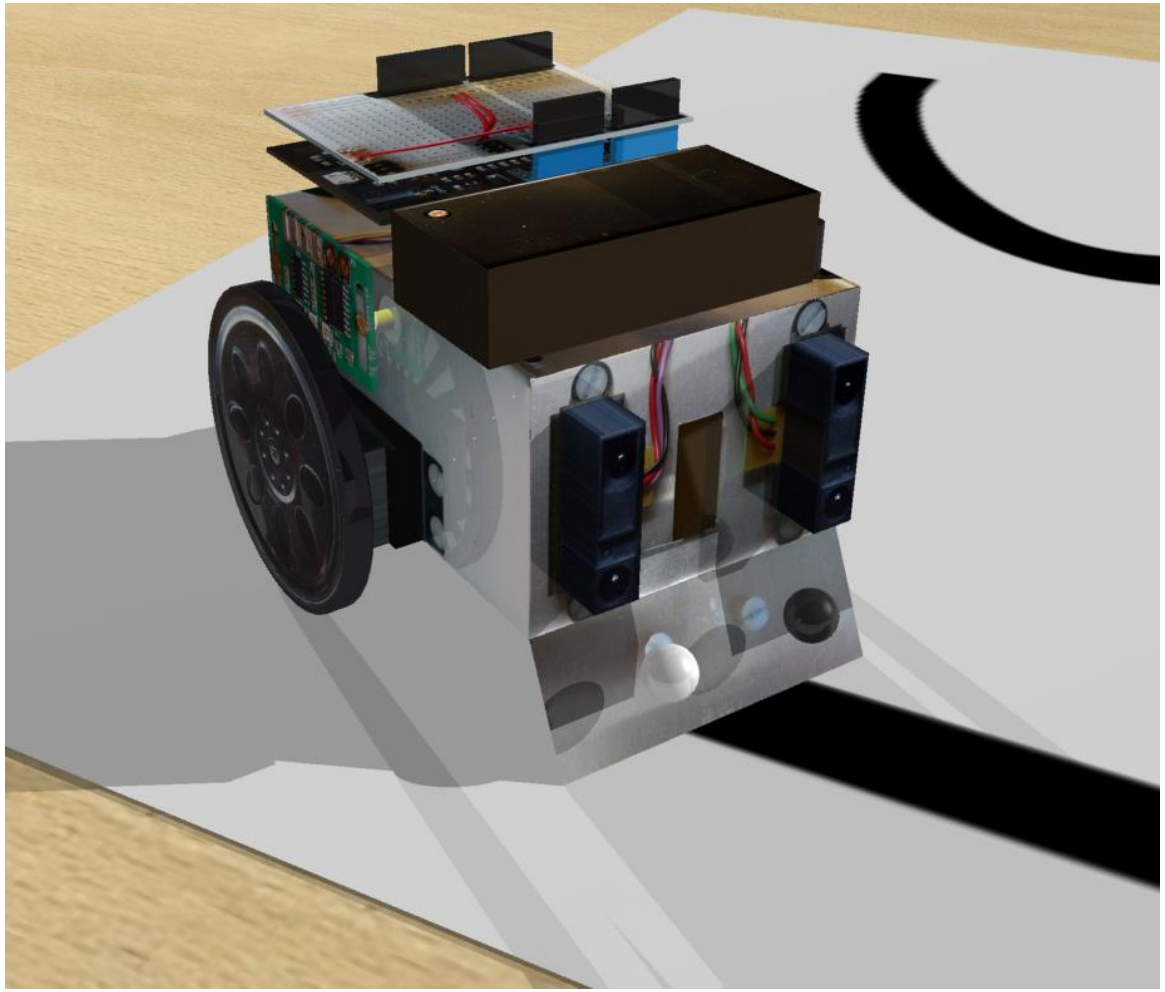
prov:used	one entity used another one
prov:wasAttributedTo	attribution of entities
prov:wasAssociatedWith	association of activities
prov:wasGeneratedBy	one entity is generated from another
prov:wasDerivedFrom	one entity is derived from another
prov:hadMember	one entity has one or more members
oslc:elaborates	an entity that elaborates on a requirement
oslc:satisfies	an entity that satisfies a requirement
oslc:verifies	an entity that verifies an assumption
into:doesNotVerify	an entity that does not verify an assumption
into:violates	an entity that violates an assumption

Line Follower Robot Pilot Study

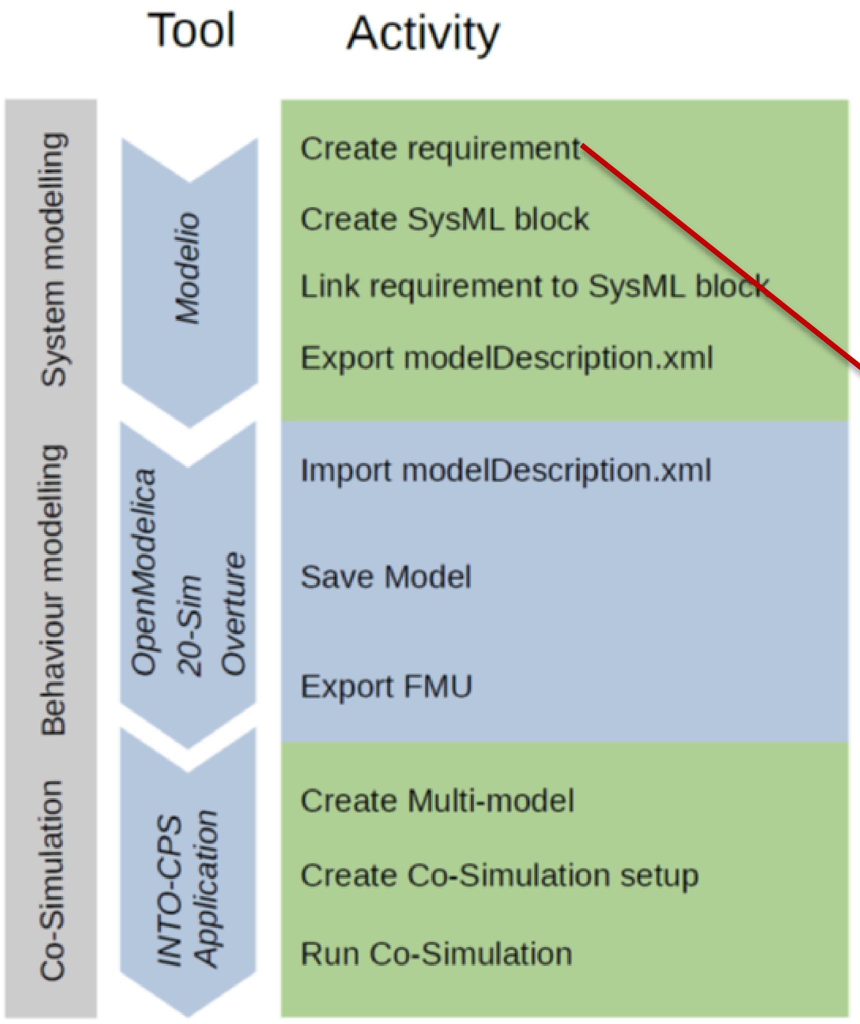
RE1: The robot must sense a black line

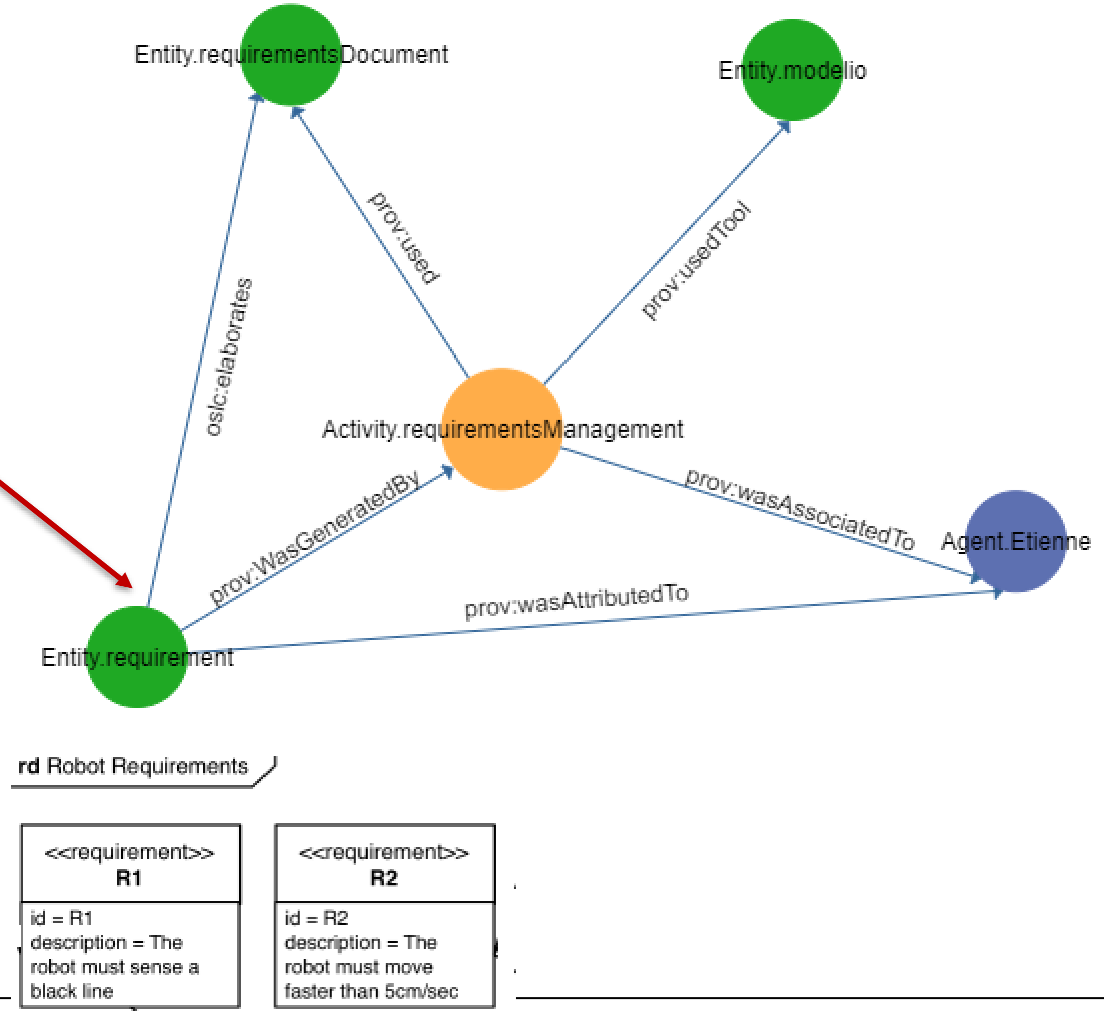
RE2: The robot must move faster than 5 cm/sec



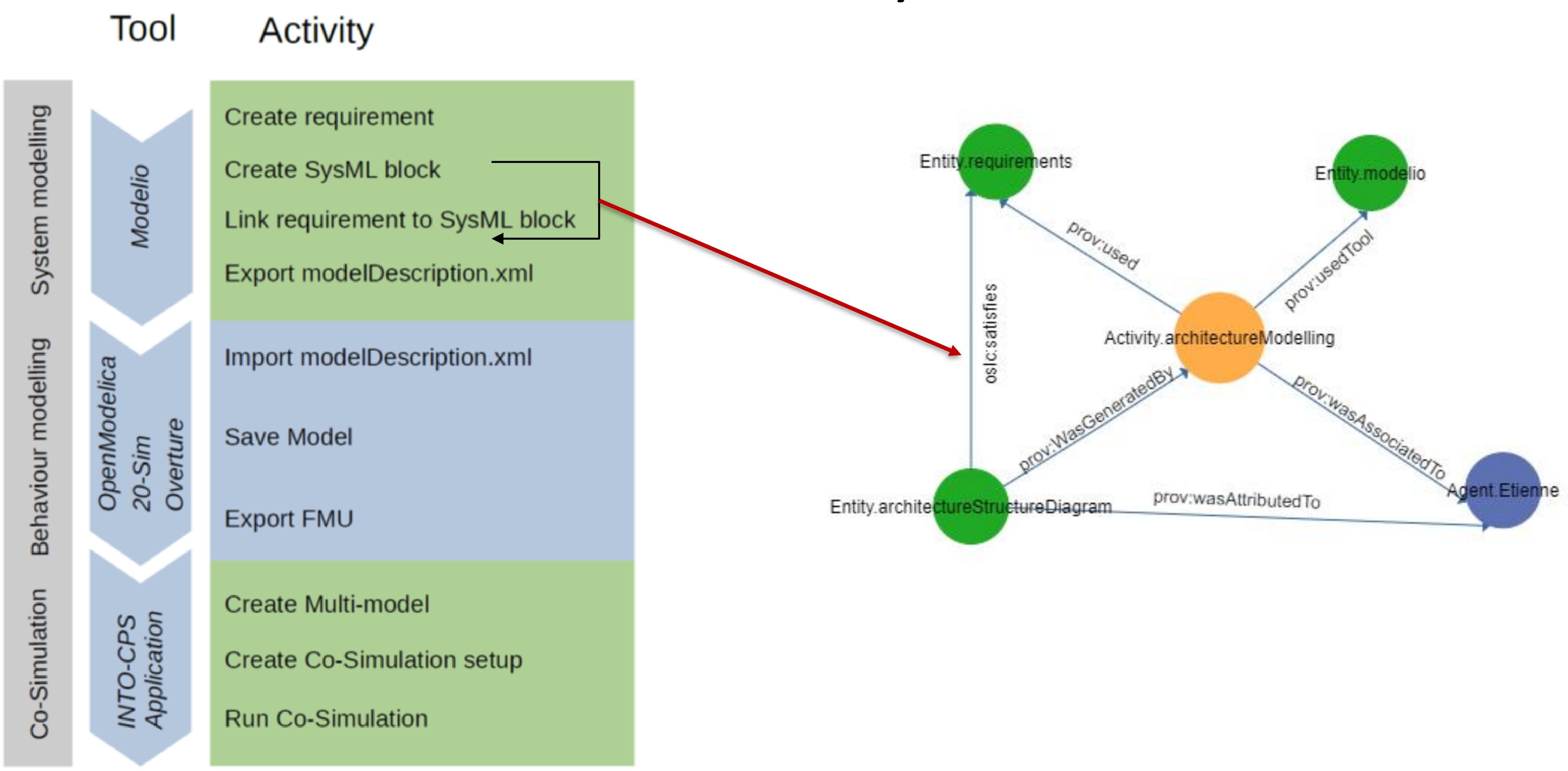


Line Follower Robot Pilot Study - Requirements Engineering

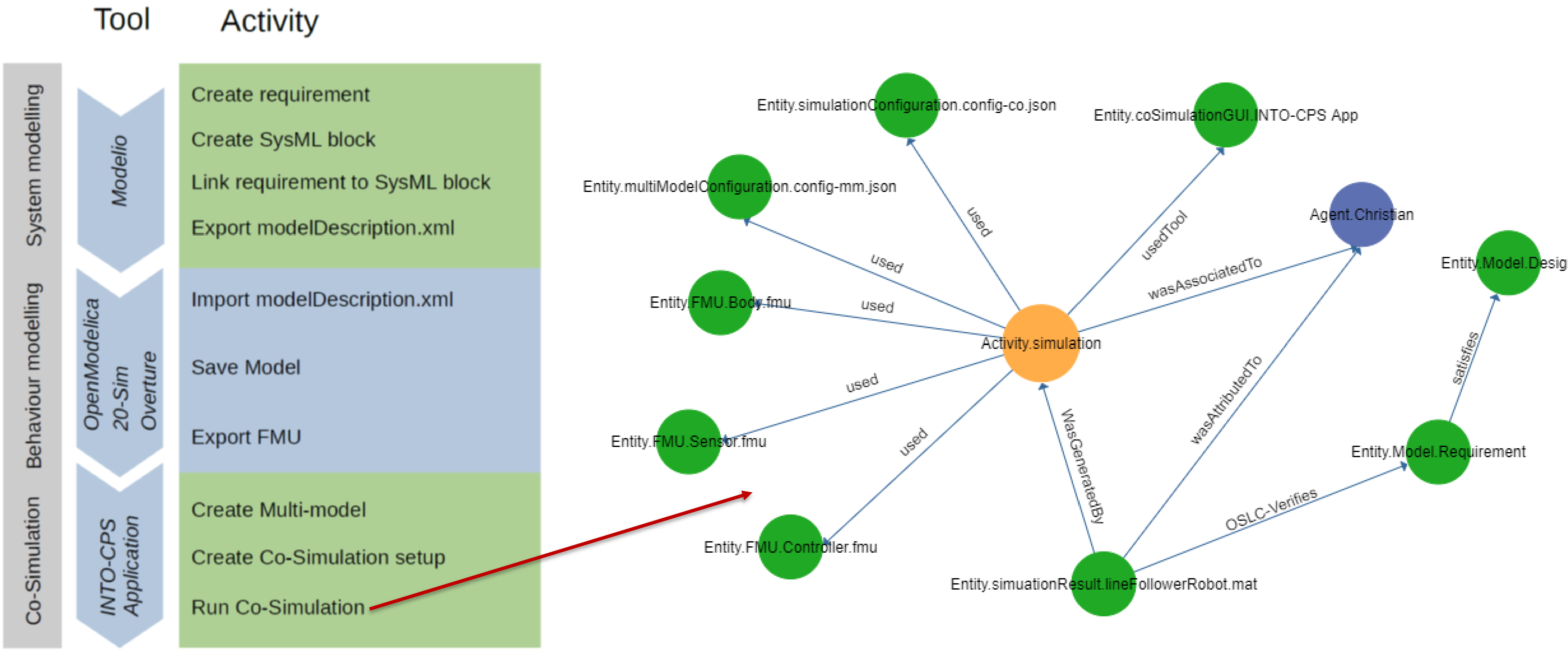




Line Follower Robot Pilot Study - Architectural Modeling

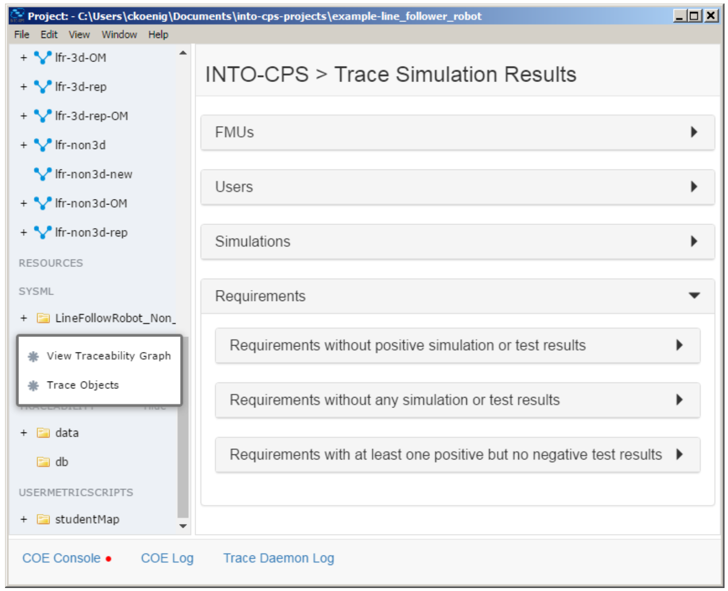


Line Follower Robot Pilot Study - Co-simulation

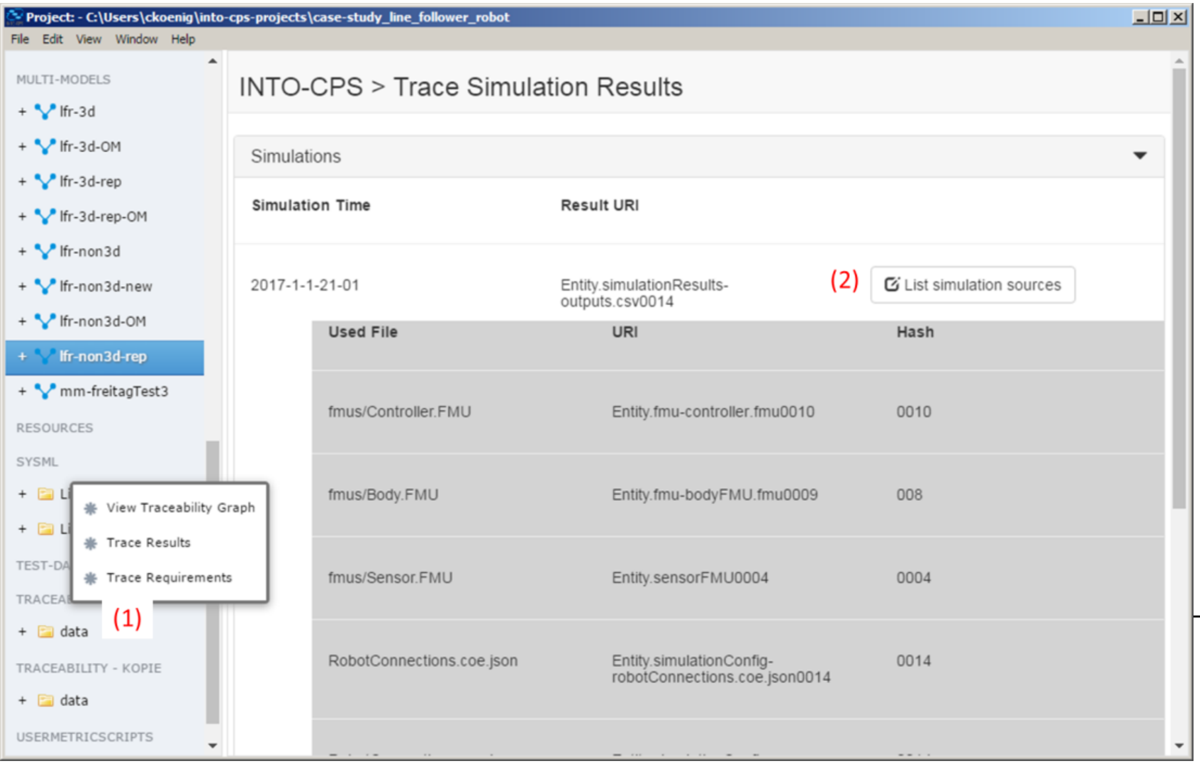


Alachew Mengist

Traceability – Query User Interface



- Search simulation results → FMUs and their versions
- Search FMUs → related Requirements
- Search users → related FMUs, Requirements, simulation results



Simulation results and files

match (n{type:'simulationResult'})-
[:Trace{name:"prov:wasGeneratedBy"}]->(m)
return n.uri, m.time, m.type

match({uri:'Entity.<Result_file>'})-
[:Trace{name:"prov:wasGeneratedBy"}]->
(simulation)-
[:Trace{name:"prov:used"}]-
(entity)
return entity.uri, entity.path, entity.hash

Conclusions

- Many benefits to integrating modeling tools with other lifecycle tooling
  - Collaborative development involving multiple users
  - Traceability and data consistency across the tool-chain, cross domain queries, enhanced impact analysis
  - Difficult integration problems (e.g. creating artifacts) are handled by the native tool
- OSLC integrations have one-time investment to support core protocols
  - Adding support for specific domains (Change, Quality, etc) becomes light weight
  - Loosely coupled tool integration

Conclusions

- Pros
  - Open Technologies (Using Open Web Standards JSON, RDF, OSLC etc...)
  - Reduce data integration costs, increase flexibility
  - Scalability

- Cons
  - Cypher query language demands expert users to manually enter queries to search the traceability database
    - Understanding of the underlying structure
  - No dedicated error handling if no connection to the traceability daemon is available.

Alachew Mengist

LINKÖPING UNIVERSITY