Jet Propulsion Laboratory
California Institute of Technology

Integration of SysML models with M&S tools for Behavior Modeling & Simulation

Bjorn Cole, Ph.D.

Systems Modeling, Analysis and Architecture Group

Overview

- Introduction
- The Ingredients for Execution
- Confidence and Traceability
- Our Version
- Summary

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Introduction

The key problem: smooth abstraction and concretization of a system with a large number of contributors

© 2016 California Institute of Technology. Government sponsorship acknowledged.

A reminder: Why do we build system models?

The system model is a tool for collecting, synchronizing, and refining a common architectural model out of the mental models of many skilled practitioners

A follow-on question: but how do we know what this combined model really means?

And when do we get to the math?

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Reuse of system models requires separation of scenario and system

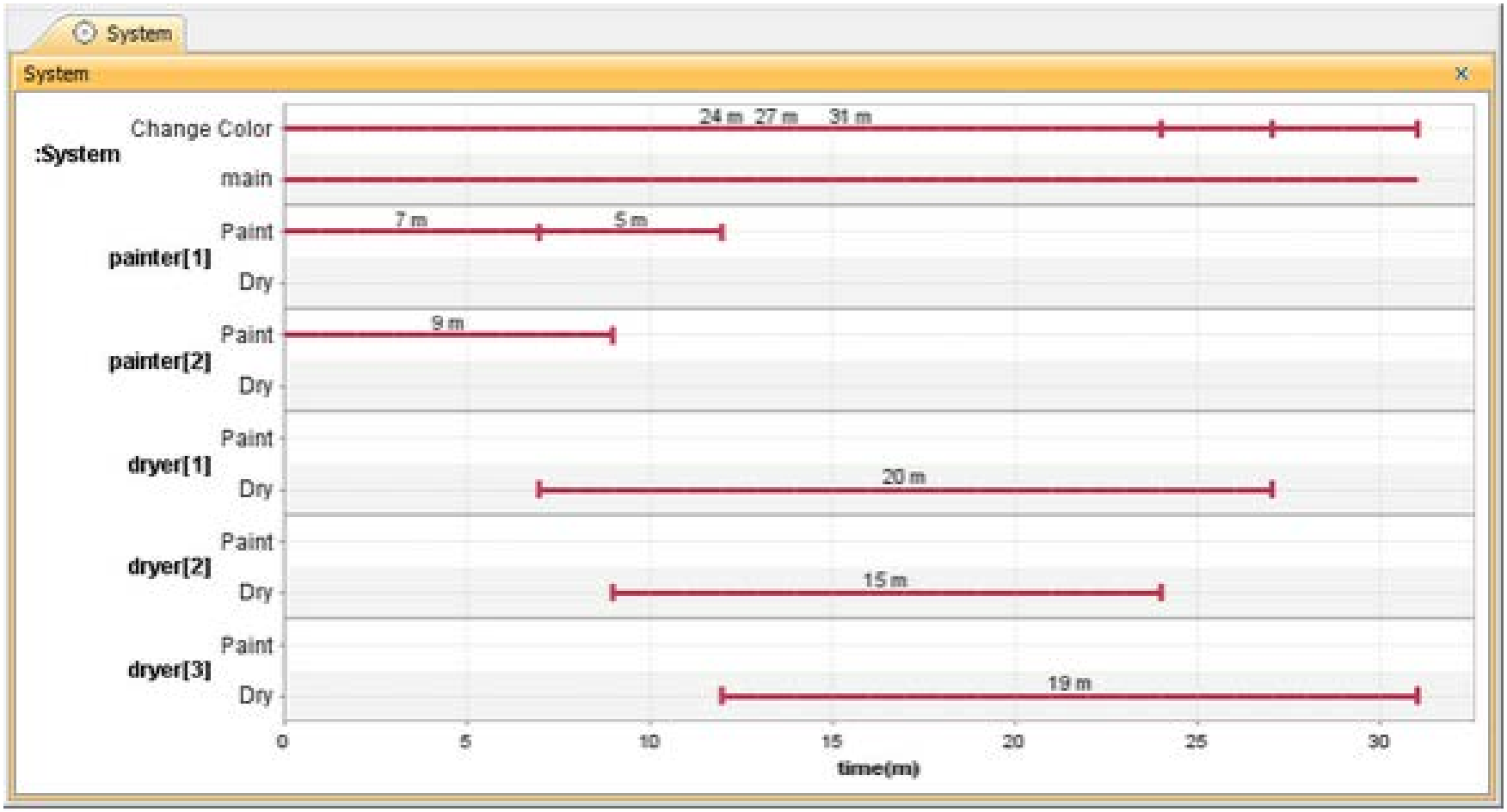
It is very easy to accidentally embed scenario assumptions within the simulation model

To have reusable assets, need to be able to drive the simulation model with multiple scenarios that drive parameters

But: Many user scenarios are couched not as parameters but as process, flow chart, sequences of events or activity

© 2016 California Institute of Technology. Government sponsorship acknowledged.

The Executable UML (xUML) vision lights a smooth path between specification and analysis



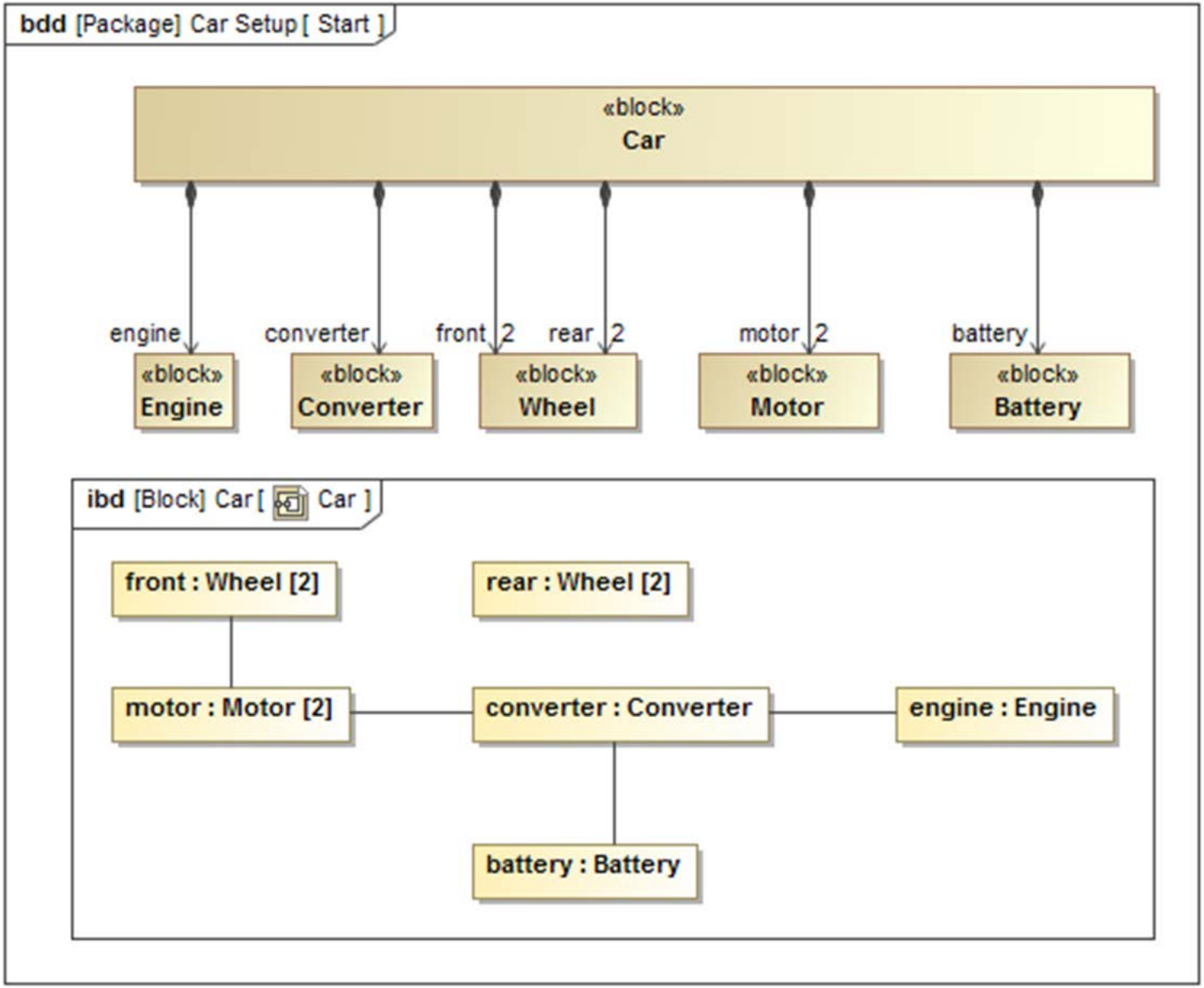
© 2016 California Institute of Technology. Government sponsorship acknowledged.

xUML is now covered by a combination of true standards and commercial anticipation

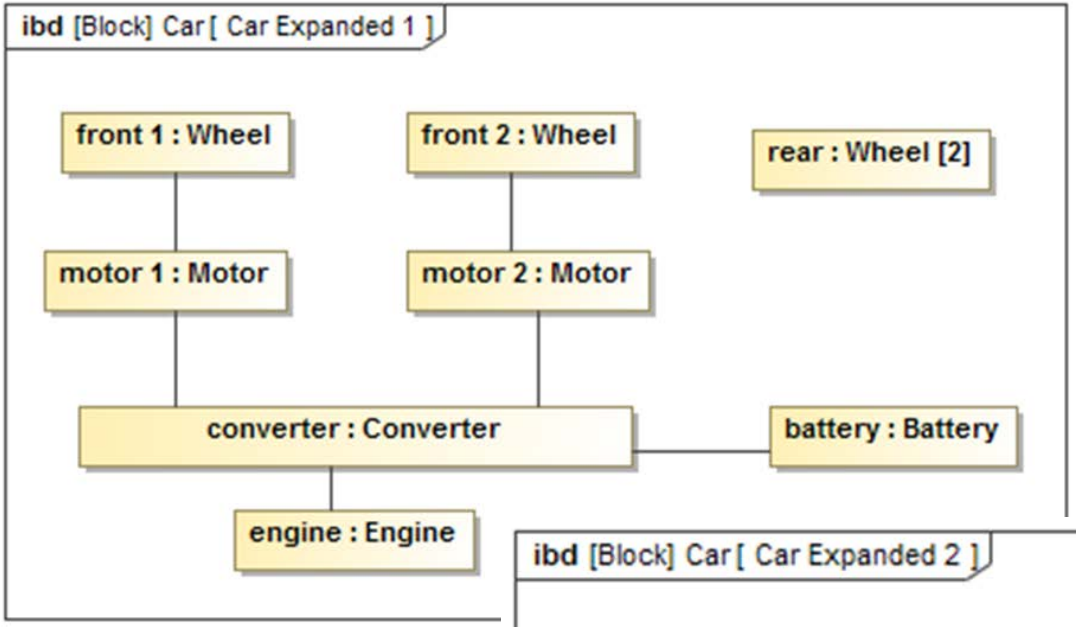
Behavior Type	OMG Standard	Other Standard	Other Implementation
Activities	fUML		
State Machines	*Precise State Machine Semantics*	SCXML	
Interactions	*Precise Interaction Semantics*		Cameo Simulation Toolkit (CST)
Time	*Precise Time Semantics*	FMI, Modelica	Cameo Model-Based Clock
Parametrics		FMI	ParaMagic, CST, MBSEPak
Structure	PSCS		

The Ingredients

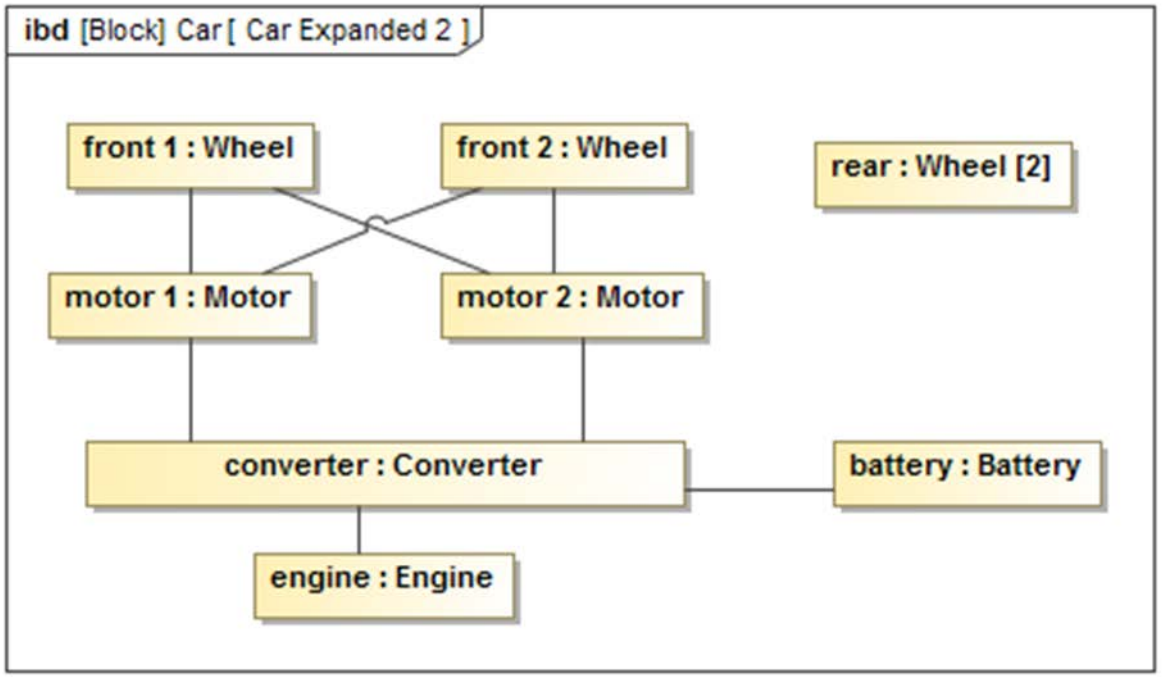
Precise Semantics for Composite Structures (PSCS) is key for starting simulations



Precise Semantics for Composite Structures (PSCS) is key for starting simulations



OR



?

Precise Semantics for Composite Structures (PSCS) is key for starting simulations

Key questions answered about how to turn a SysML structural model into runtime instances:

- How to deal with connectors on parts with multiplicity
- How to expand parts with multiplicity
- How to deal with default and initial values
- How to connect structural aspects (parts, ports, connectors) to behavioral specifications in fUML

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Functional Subset for Executable UML Models (fUML) provides an execution engine

Key questions answered about how to execute a UML model:

- What to do with different actions?
  - Evaluate
  - Execute
  - Start new behaviors
- How to handle signals and events?

Does NOT handle constraints, however

© 2016 California Institute of Technology. Government sponsorship acknowledged.

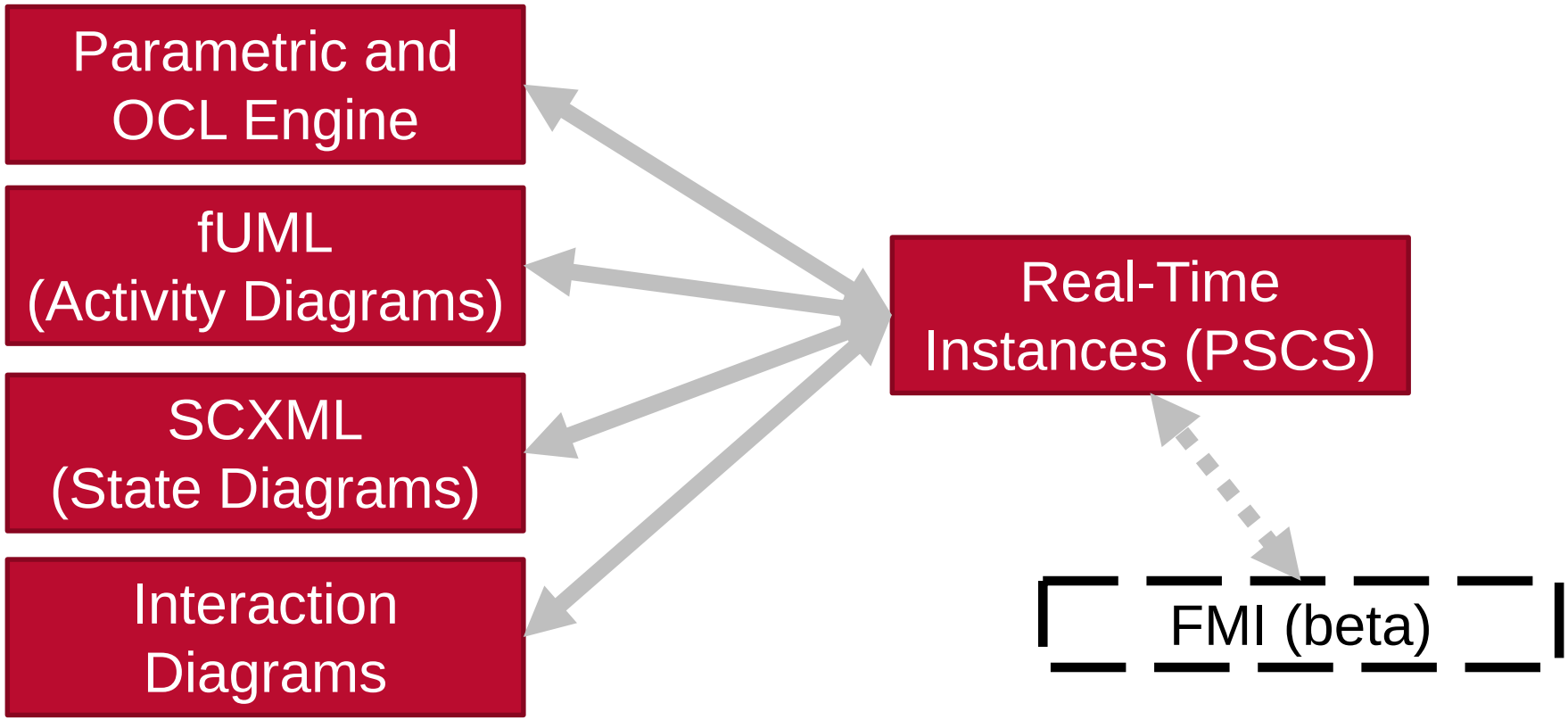
W3C State Chart XML (SCXML) provides standard state machine interpretation

Key questions answered about how to execute a state machine:

- How to absorb and raise events
- How to call additional behavior
- How to handle transitions

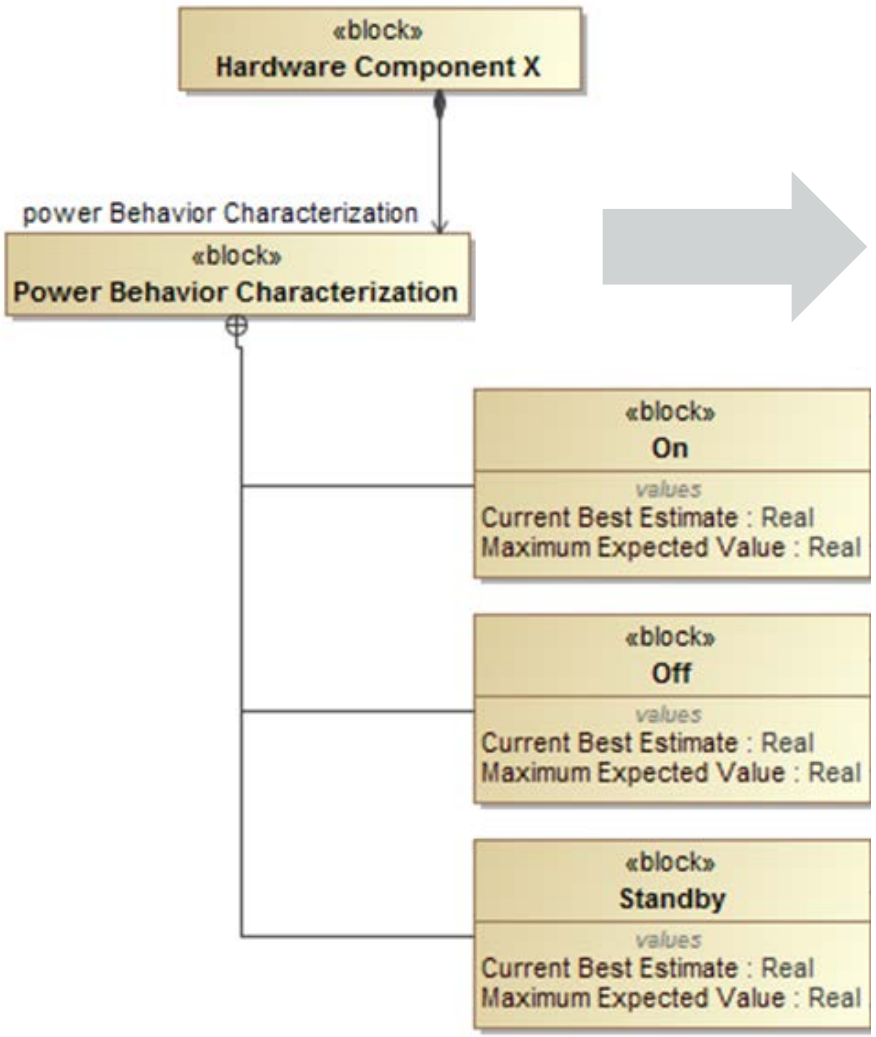
© 2016 California Institute of Technology. Government sponsorship acknowledged.

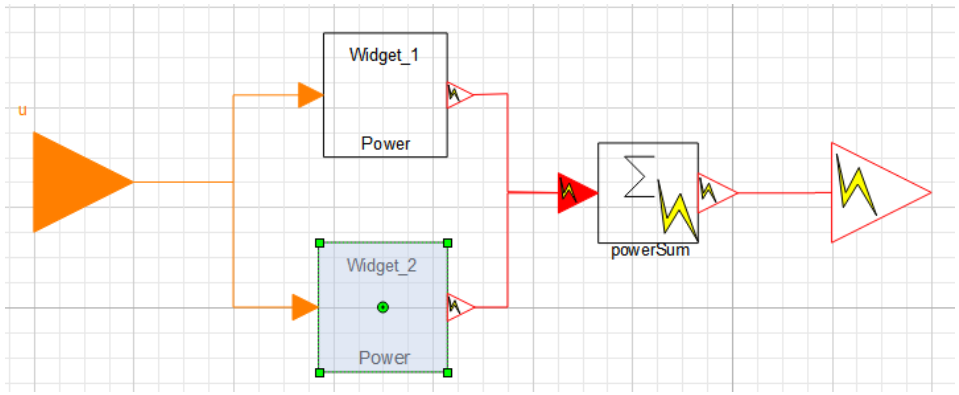
The Cameo Simulation Toolkit in MagicDraw connects standards with placeholder semantics to cover much of UML



© 2016 California Institute of Technology. Government sponsorship acknowledged.

Parameter export is not enough; structure is also required

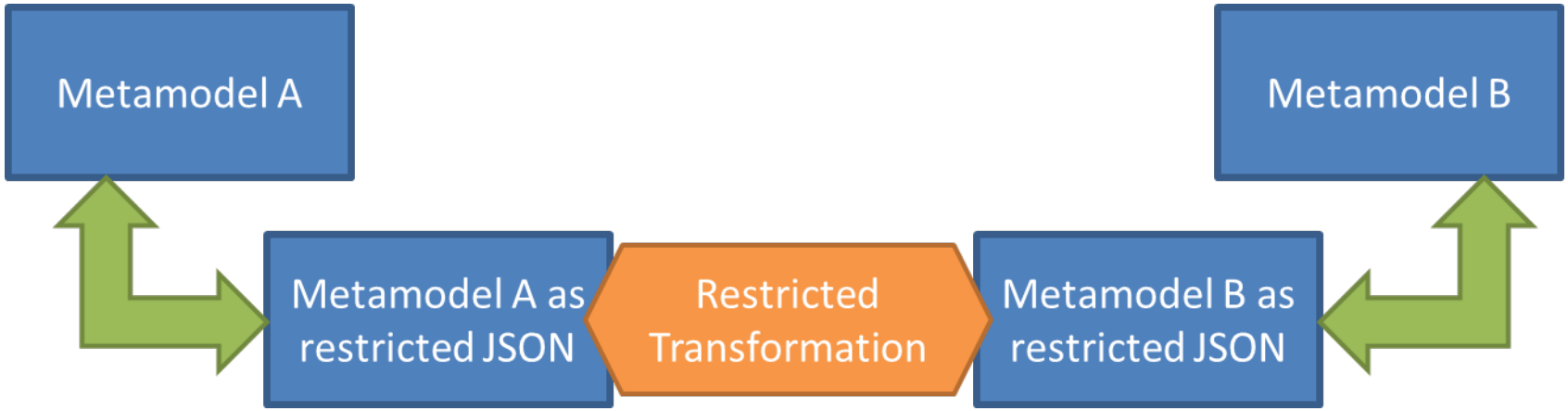




Object-oriented models benefit from structure to organize parameters and resemble their real-world counterparts

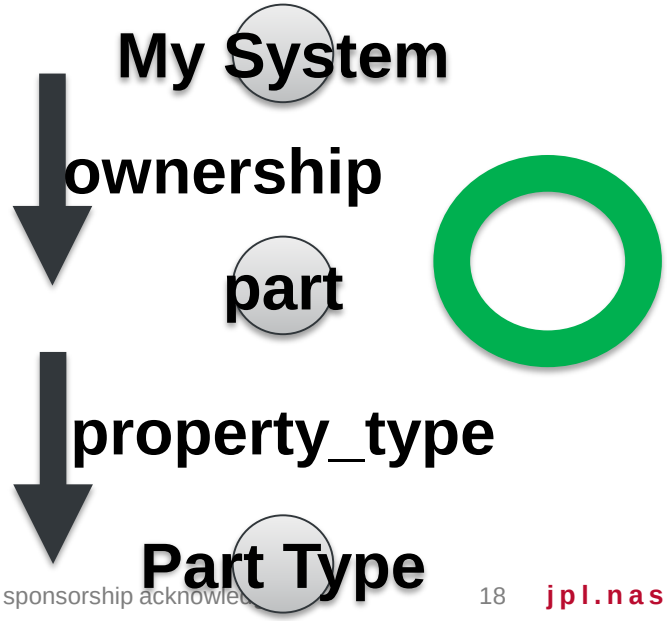
© 2016 California Institute of Technology. Government sponsorship acknowledged.

Structure export is achieved by embracing the graph form of SysML models (Cole, Dinkel 2016)




```
Block MySystem {
   part : PartType
}

Block PartType
```



© 2016 California Institute of Technology. Government sponsorship acknowled

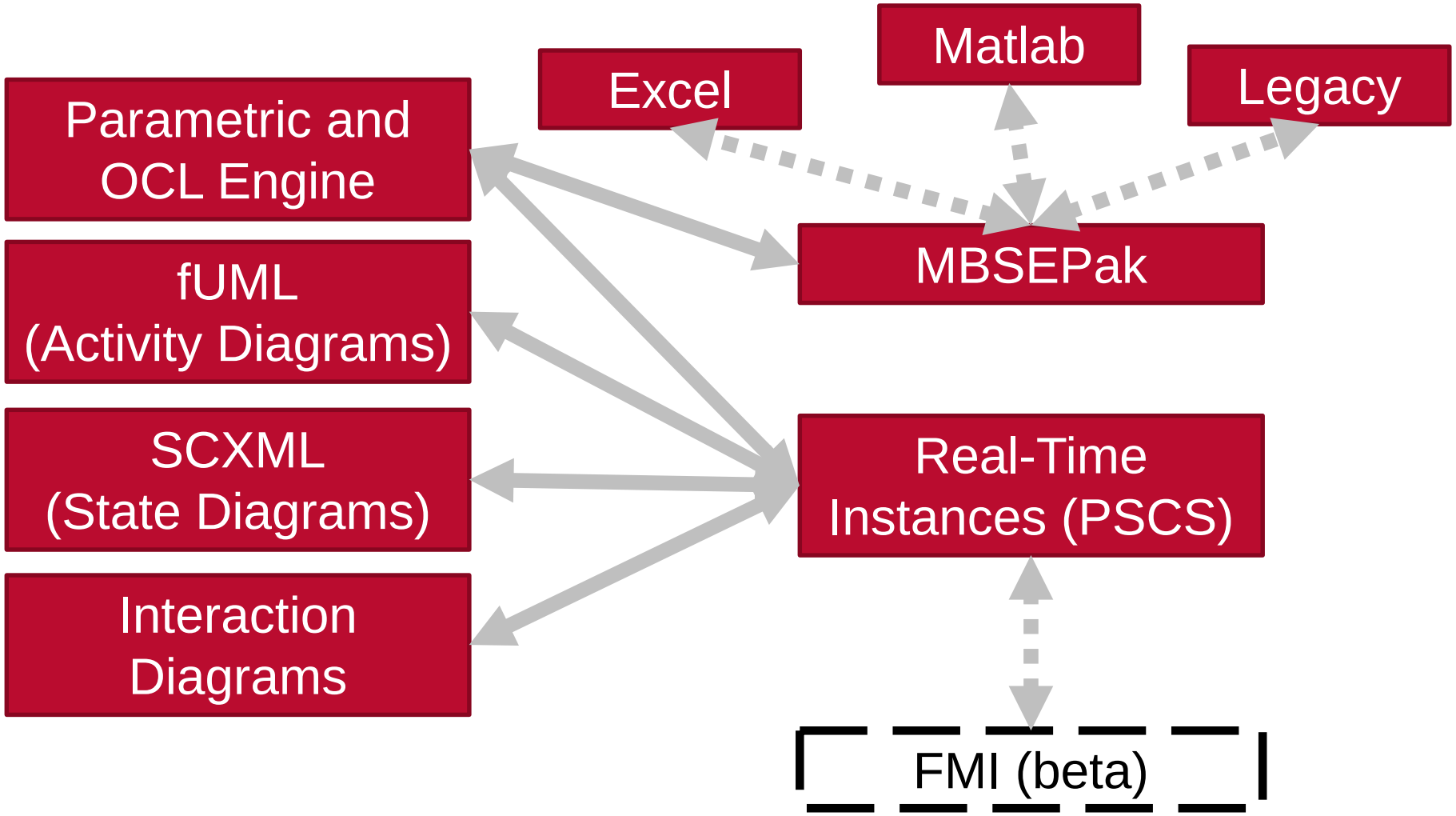
Phoenix Integration MBSE Pak provides a final piece of the execution puzzle

The MBSE Pak connects to Phoenix Integration's Analysis Server:

- A uniform API to arbitrary tools, with many connections already built
- Calls to wrapped models can be made along with other parametric evaluations in Cameo Simulation Toolkit
- This capability also provides a hook for time control (e.g., adaptive stepping)

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Phoenix Integration MBSE Pak provides a final piece of the execution puzzle; complexity below matches organizational complexity



© 2016 California Institute of Technology. Government sponsorship acknowledged.

Integration of models works better with multiple steps

The steps move from independent model build, to wrapped model, to integration with system model and/or to deployment to web services

Each incremental step provides chance for validation

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Confidence and Traceability

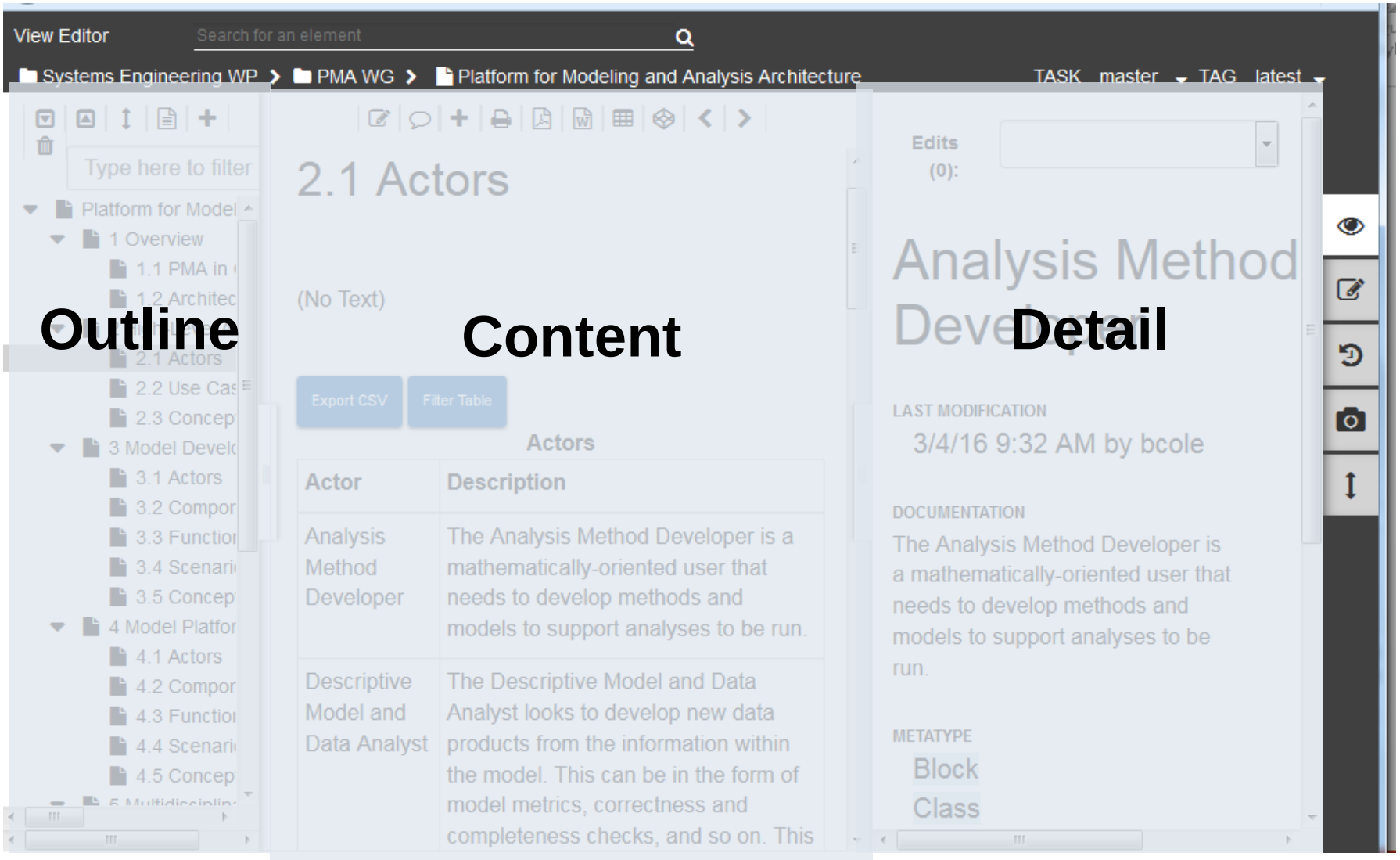
Analysis workflow includes two other key steps: data gathering and reporting

Good data and model versioning is required to understand the basis of results

Often the first question after a new case is run is “so what changed?”

© 2016 California Institute of Technology. Government sponsorship acknowledged.

OpenMBEE provides a platform for data control and reporting with the model as a direct connection



OpenMBEE provides a platform for data control and reporting with the model as a direct connection

mass

/opencae-architecture/OpenCAE/OpenCAE/ESD CAE Effort/Work Packages/Applicatio
Packages/Applications and Services/Views/View Editor/View Editor User's Guide/Model
/A/mass

DOCUMENTATION

This field is empty, but you can still click here to cross-reference a placeholder.

METATYPE

Property

VALUE

2.3

Values can be searched and directly cross-referenced in documents. Document updates when value does.

RELATED DOCUMENTS

View Editor User's Guide > Grouped Horizontal Bar Charts

View Editor User's Guide > Parallel Axis

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Model-based documentation of analysis maintains a live record that is updated; static versions can be saved for archive

Element History

Type here to filter history

--- Latest ---

--- Latest ---
bcole - 3/4/16 9:32 AM
bcole - 3/4/16 9:32 AM
bcole - 3/4/16 9:32 AM
bcole - 3/4/16 9:25 AM
mpiette - 12/7/15 9:45 AM
mpiette - 12/7/15 9:44 AM

LAST MODIFICATION

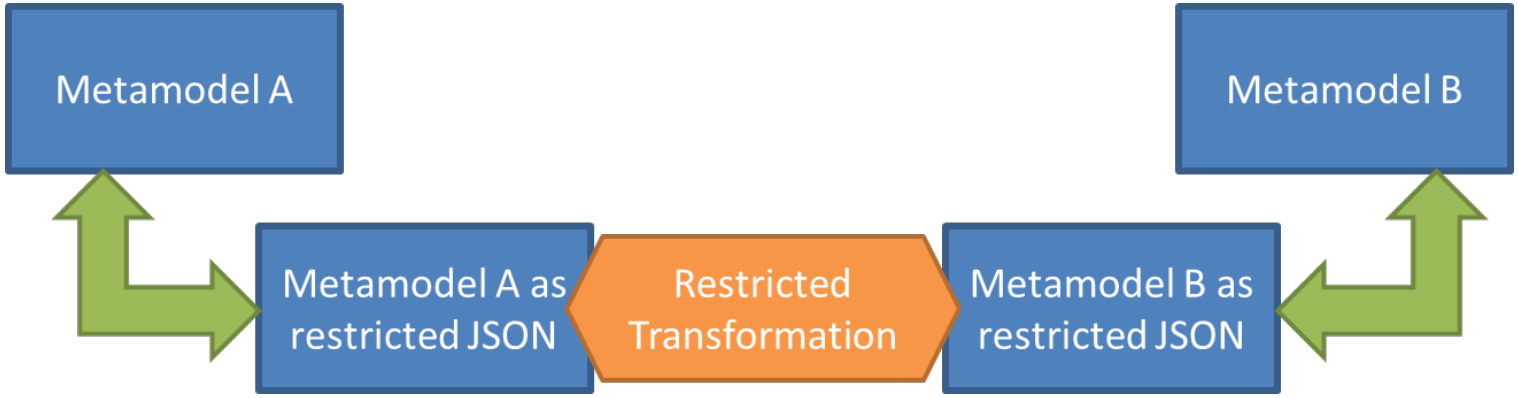
3/4/16 9:32 AM by bcole

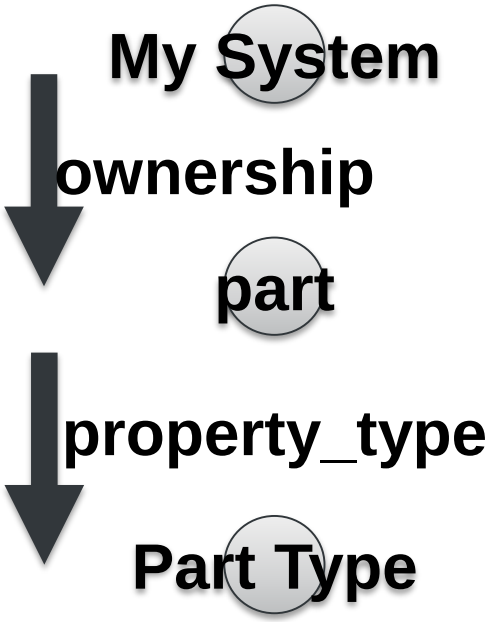
DOCUMENTATION

The Analysis Method Developer is a mathematically-oriented user that needs to develop methods and models to support analyses to be run.

Our Version

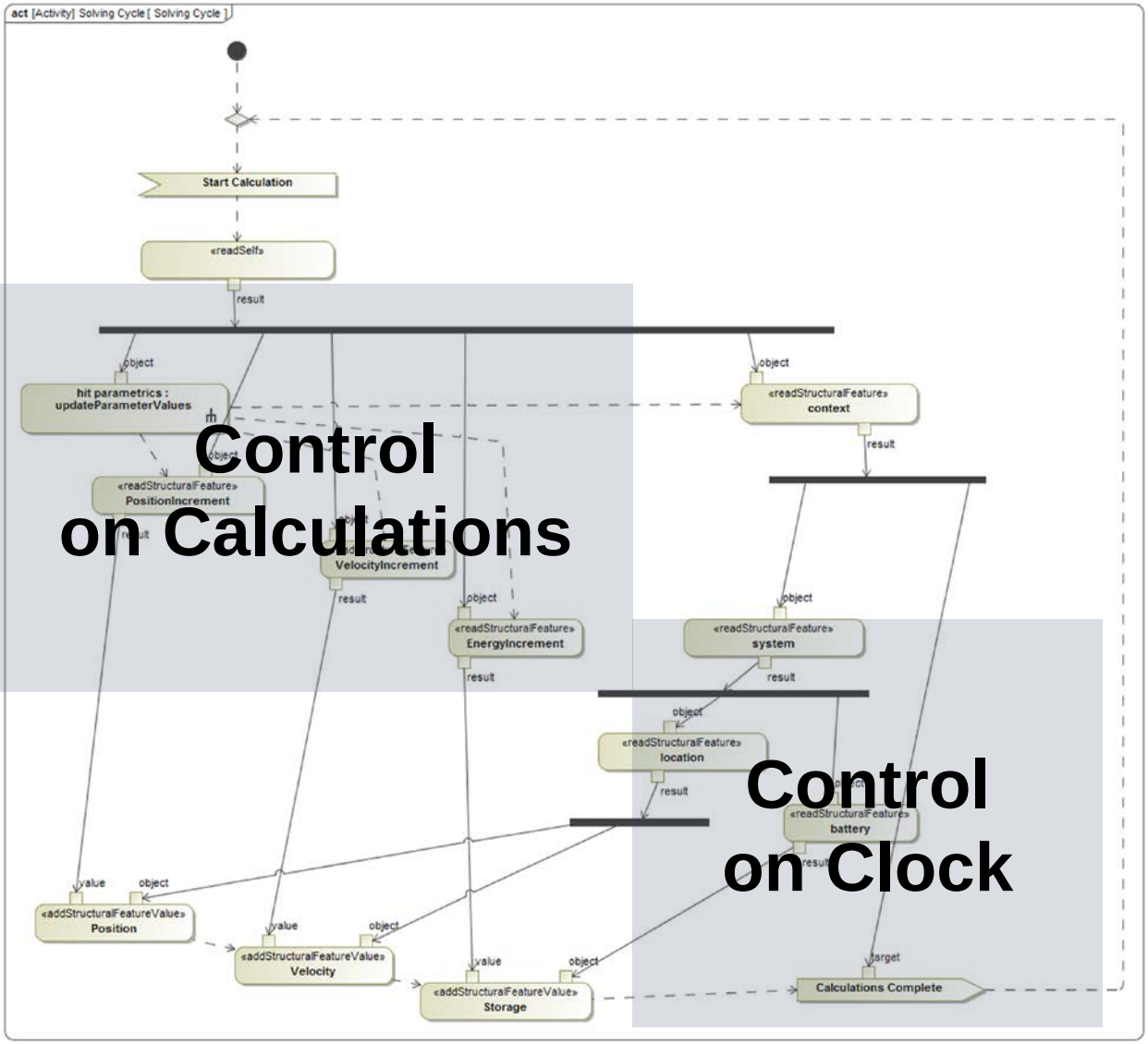
Some of the structure of the simulation comes from the system model





© 2016 California Institute of Technology. Government sponsorship acknowledged.

Careful time control provides for an “over executive” to move things around



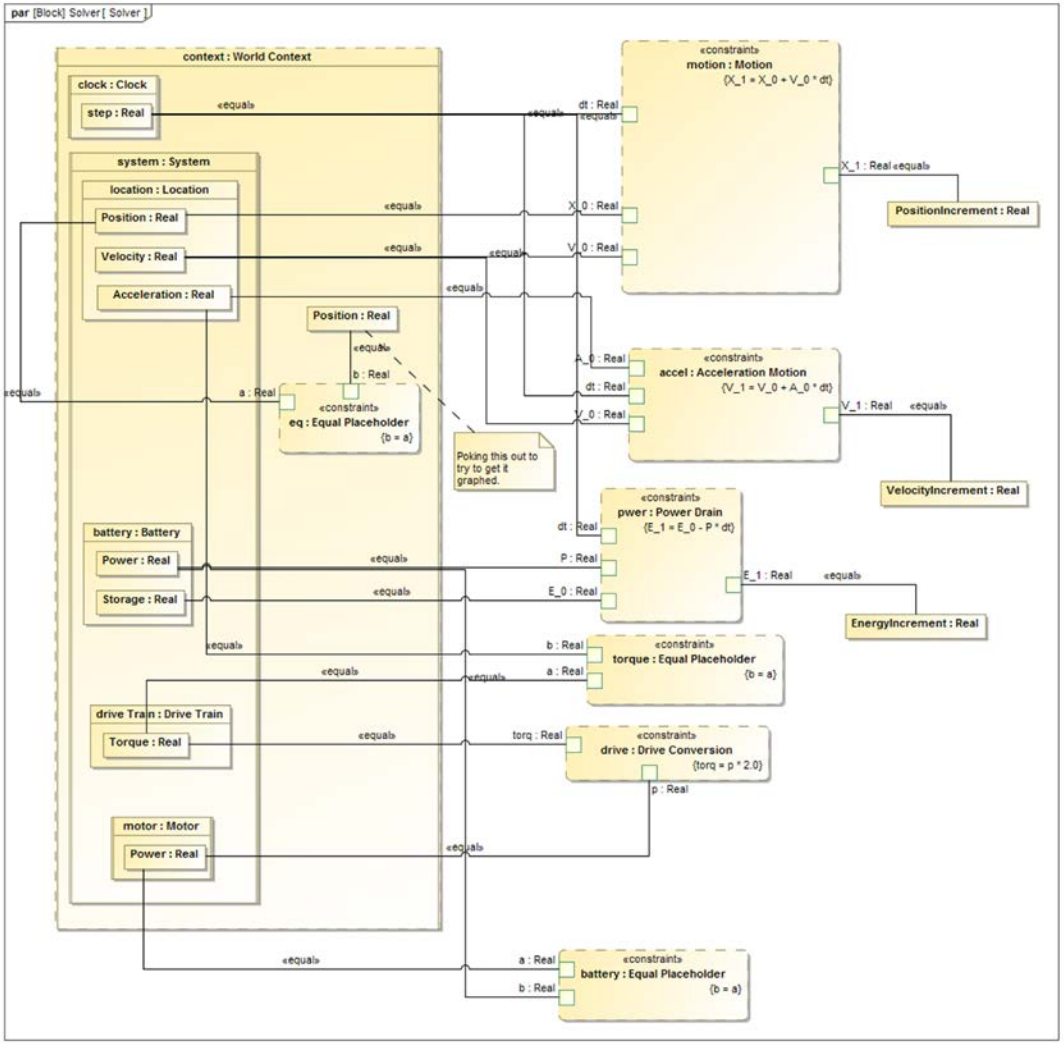
© 2016 California Institute of Technology. Government sponsorship acknowledged.

So far the implementation is much more about sequential sim than co-sim

- Driving connections are through parametrics between analysis tools
- Simulation Toolkit is source of events and signals moving around the system
- Piece that needs development is an appropriate hybrid solver for the framework
  - CST is starting toward an event calendar in 18.3
  - Need to easily gather key states
  - Need to predict / react to event times and maintain event stack
  - Need to adapt time steps
  - *Can* be done in current suite, but not much fun

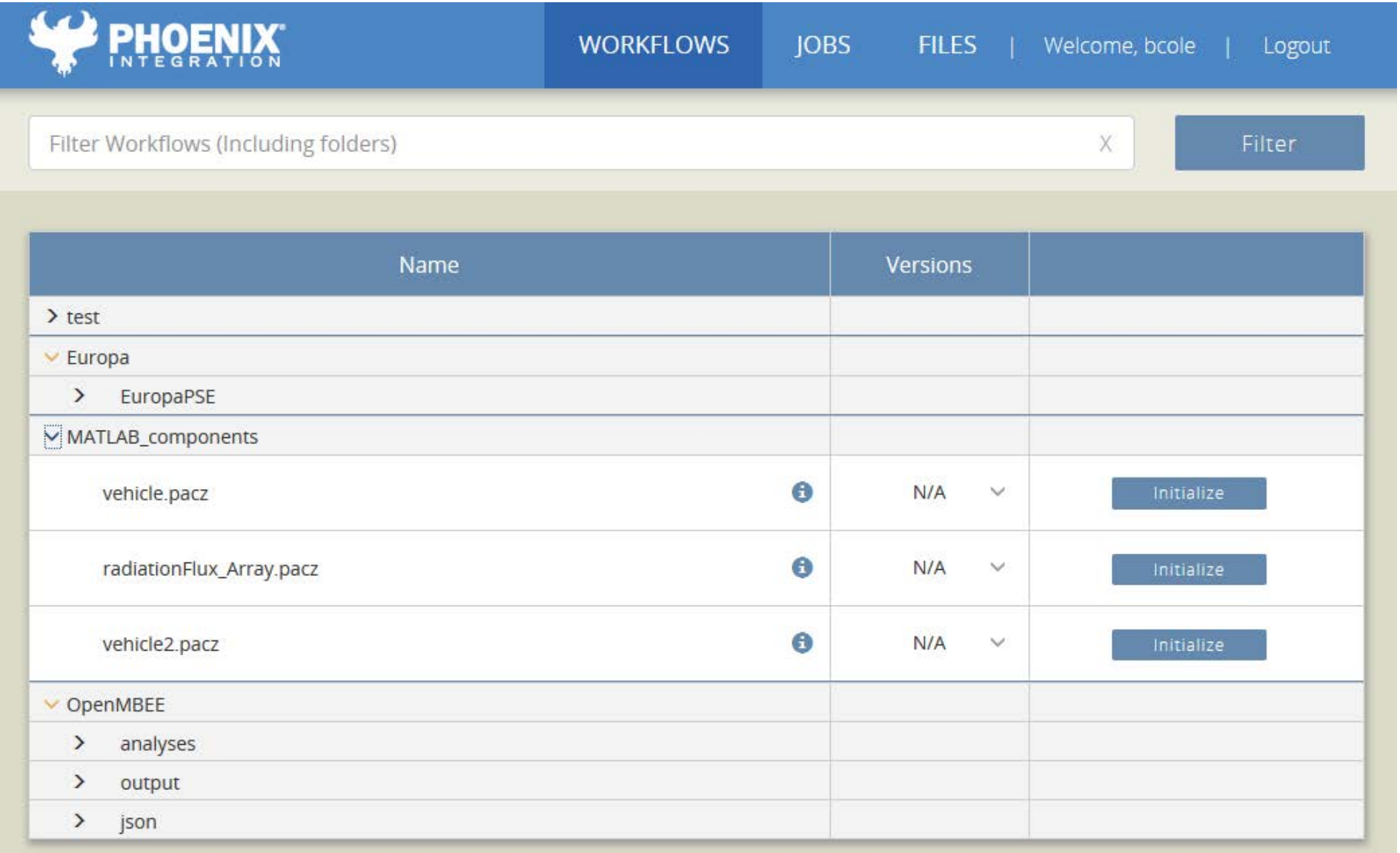
© 2016 California Institute of Technology. Government sponsorship acknowledged.

The MBSEPak enhanced parametric diagram allows you to work out analysis flow and connect it directly to the system



© 2016 California Institute of Technology. Government sponsorship acknowledged.

Deployment to ModelCenter Cloud “publishes” analyses to a larger team



Deployment to ModelCenter Cloud “publishes” analyses to a larger team

Ethan Model w=0.1

Completed

100.00% completed

X

Run ID	width	height	length	YM	Ix	deflection
0 ✓ LOG	0.1	0.03	1	200000000000	2.25e-007	0.0037037037037037

« ‹ 1 › »

Export CSV

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Storing inputs and results with OpenMBEE provides data control

- Much discussion went into instance specifications v. specialized blocks v. normal blocks for storing results
- Instance specifications slowly emerged as preferred approach
  - Tool builders favoring this approach
  - Most aligned with OMG metamodeling approach
  - Also easy to make copies for new scenarios; separated from the model structure

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Summary

Recent experience and survey of the field showing increasing maturity but we are currently in the land of first versions

- Commercial tools to connect to system models are clearly available
- Vision is in place, standards on the way
- Many of these tools are still in version 1 or 2 – lots feedback still going to vendors on usability, use cases, and approach
- Scale is increasing but not yet here
- Need to keep an eye on the lessons of our mechanical and electrical friends in development of their CAD tools and analysis platforms

© 2016 California Institute of Technology. Government sponsorship acknowledged.

The High Points:

- Executability of systems models is moving fast – a lot of it is already here!
- Increasing availability of standards; we should push for more and demand interoperability over strict compliance
  - Use of interim solutions by vendors to provide functionality has served as a good proving ground for putative standards
- Commercial tools are growing but still early
- At JPL, we've turned this crank a time or two but lots for us to learn as we broaden the user base as well

© 2016 California Institute of Technology. Government sponsorship acknowledged.

After the talk – let's engage!

- Professional societies
  - (INCOSE, AIAA, IEEE)
- Institution to institution collaboration

bjorn.cole@jpl.nasa.gov

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov