Atego Artisan Studio Simulation

James Hummell Principal Engineer

Atego Artisan Studio integrations with Simulink

- Artisan Studio with parametrics and Simulink synchronizer
- Artisan Studio with parametrics, and state behavior, and code generation using Simulink synchronizer, Simulink code gen to C/C++ lib
- Artisan Studio using SySim use Simulink during model execution
- Artisan Studio with Atego Parasolver uses Simulink as a solution engine for parametrics, and returns values to Artisan
- Artisan Studio state behavior, Internal block definition and ports using ACE to tie into Simulink blocks and ports

Atego Solutions

ategoProcessDirector™
Process Definition, Deployment & Management

aonixPerc®
aonixPerc® Ultra
aonixPerc® Pico
aonixPerc® UltraSMP
aonixPerc® Raven
Java VM IDE

ategoExerpt™
Requirements Interchange

artisanStudio®
artisanStudio™ SySim
Artisan Studio Add-On
Modeling

Code Generation
artisanStudio™ Automatic Code Synchronizer

blueriverX32plus
C / C++ IDE

aonixObjectAda®
aonixObjectAda® Real-Time
aonixObjectAda® Raven
Ada IDE / Runtime

ategoAce™
Co-Simulation

ategoWorkbench™
Tool-Chain Collaboration & Hosting

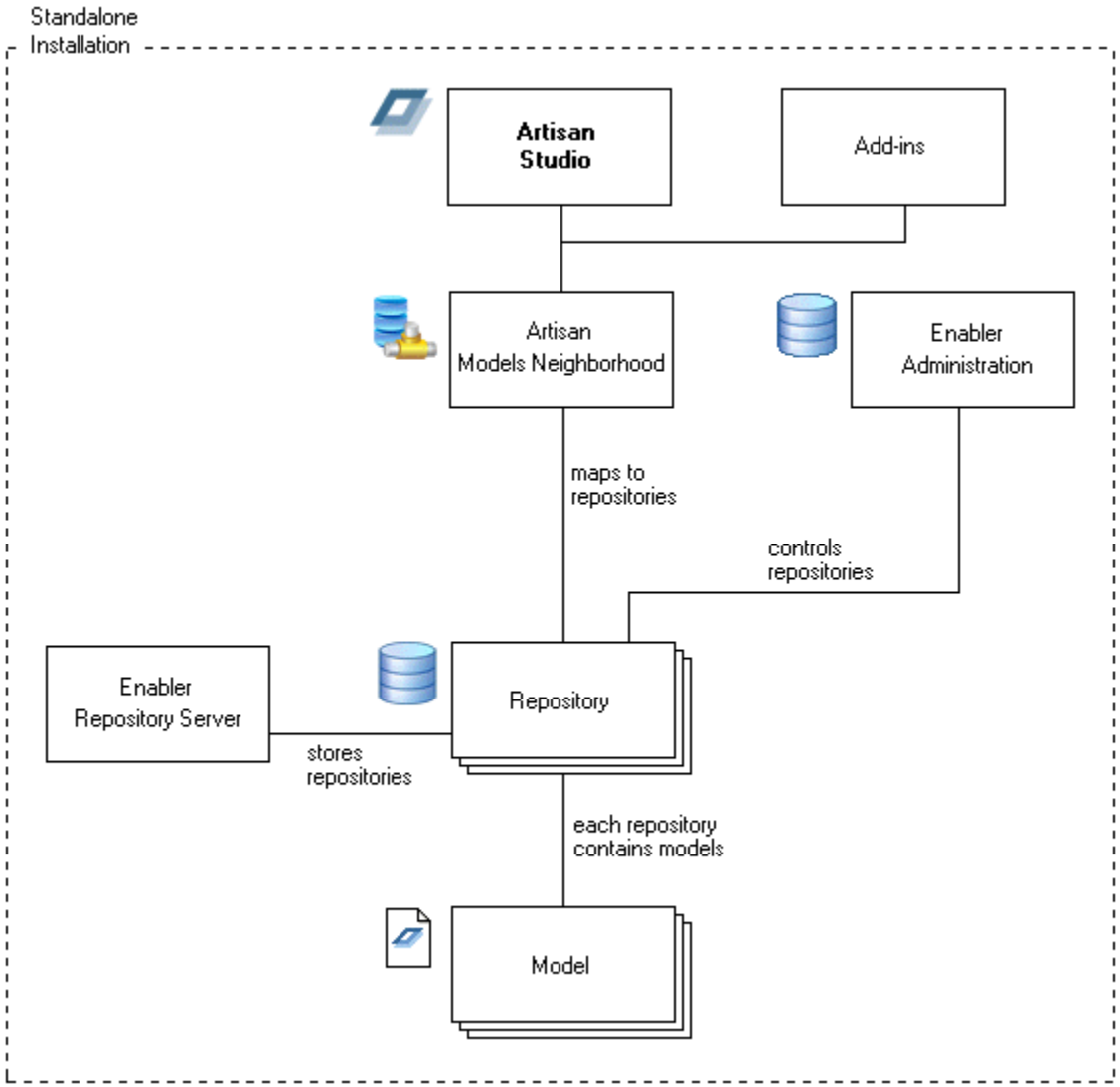
HighRelyCheck
HighRelyTrace

ategoGlobalServices™
Professional Services

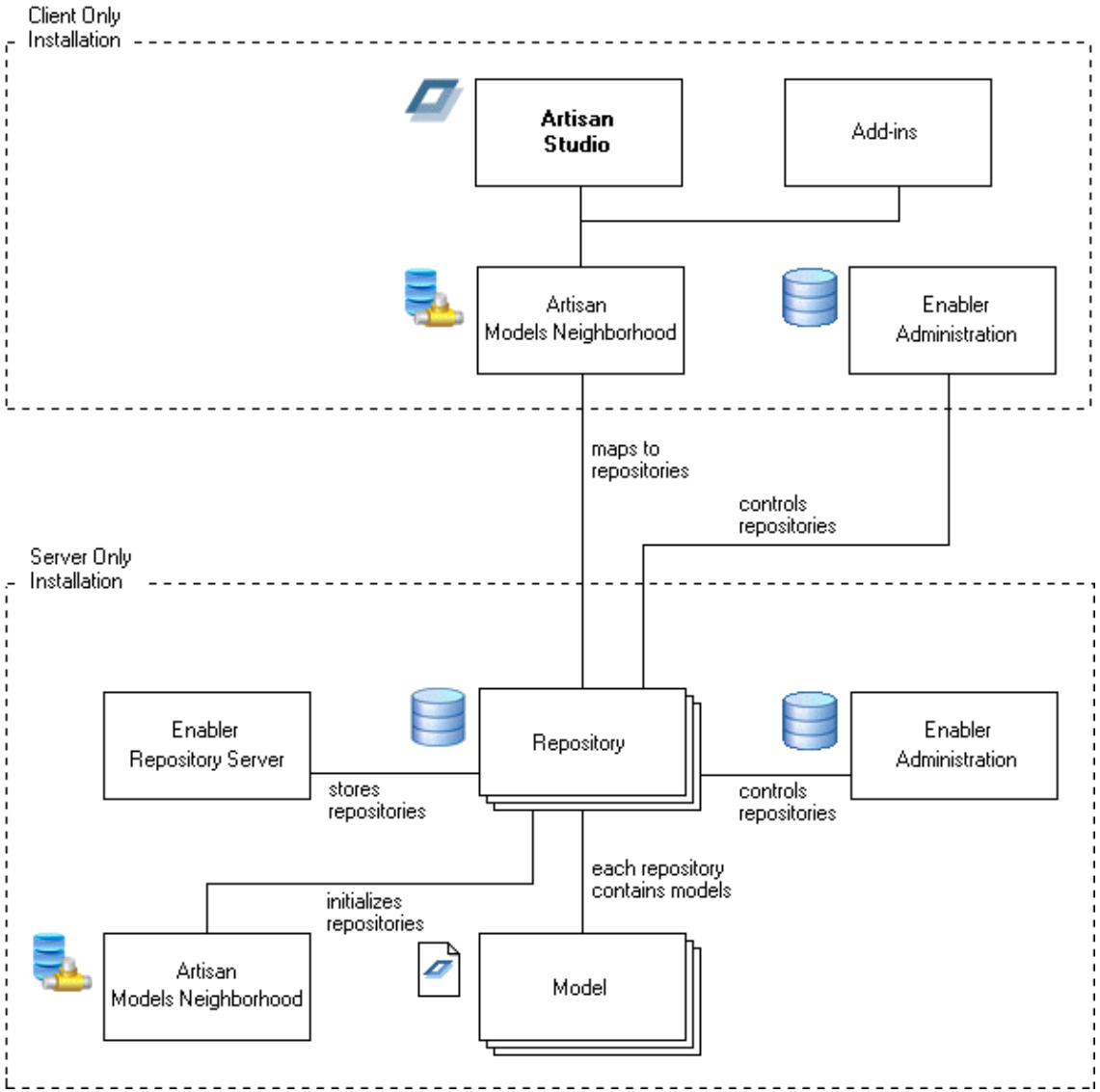
ategoHighRely™
Certification Solutions

atego™

Artisan Studio Architecture Standalone

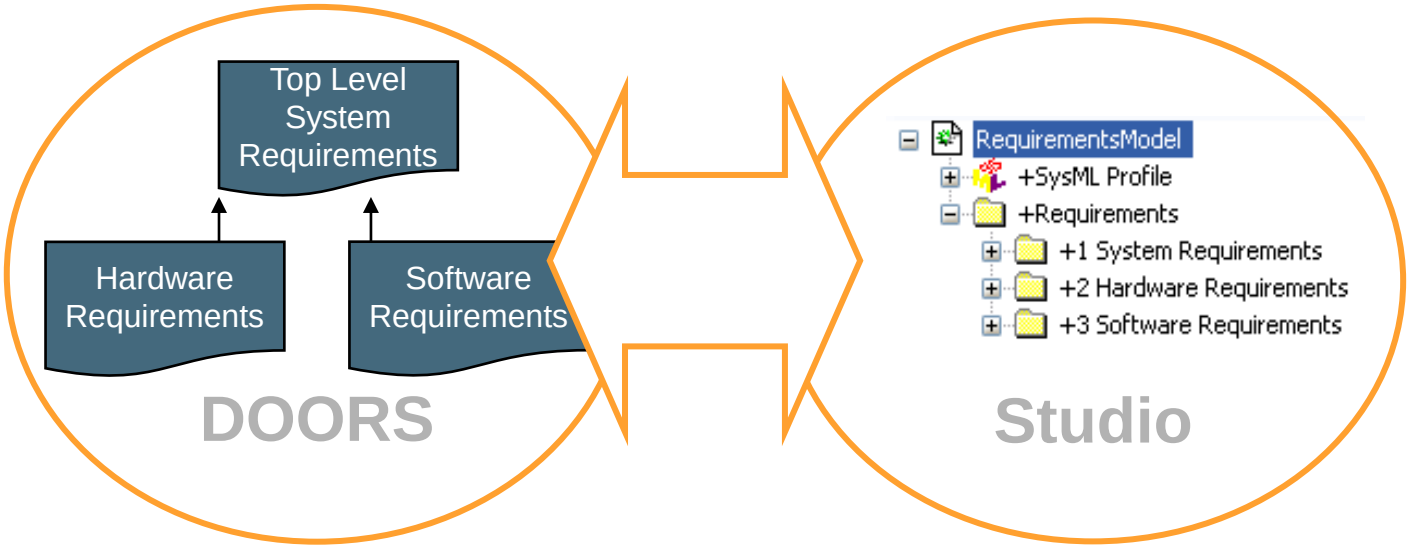


Artisan Studio Architecture Client/Server

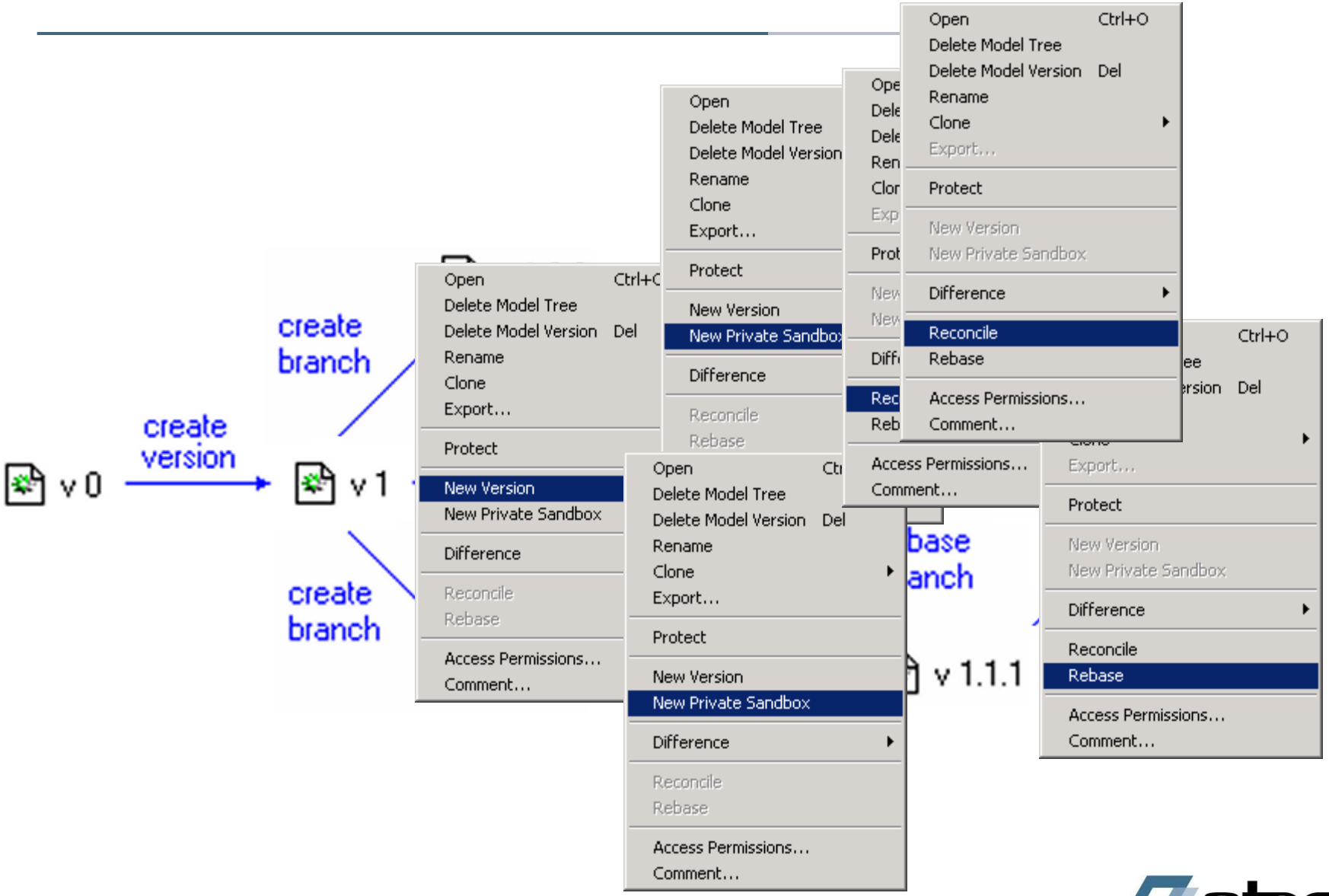


DOORS Synchronizer

- Multiple modules per Studio model
  - Able to bring requirements from multiple modules into a single Studio model
  - Create traceability in Studio
  - Export back to DOORS (including surrogates)

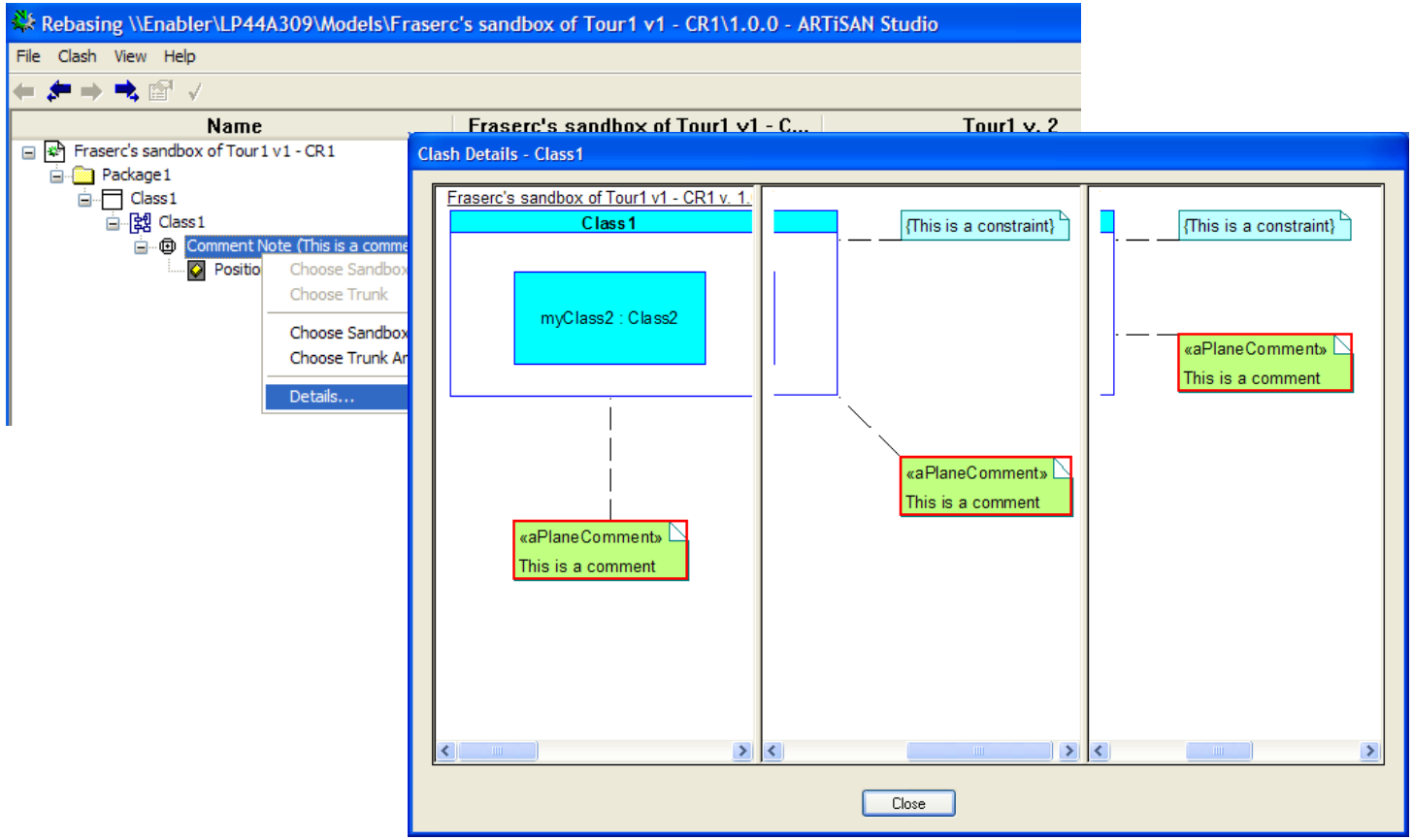


Repository-based CM – Using Private Sandboxes



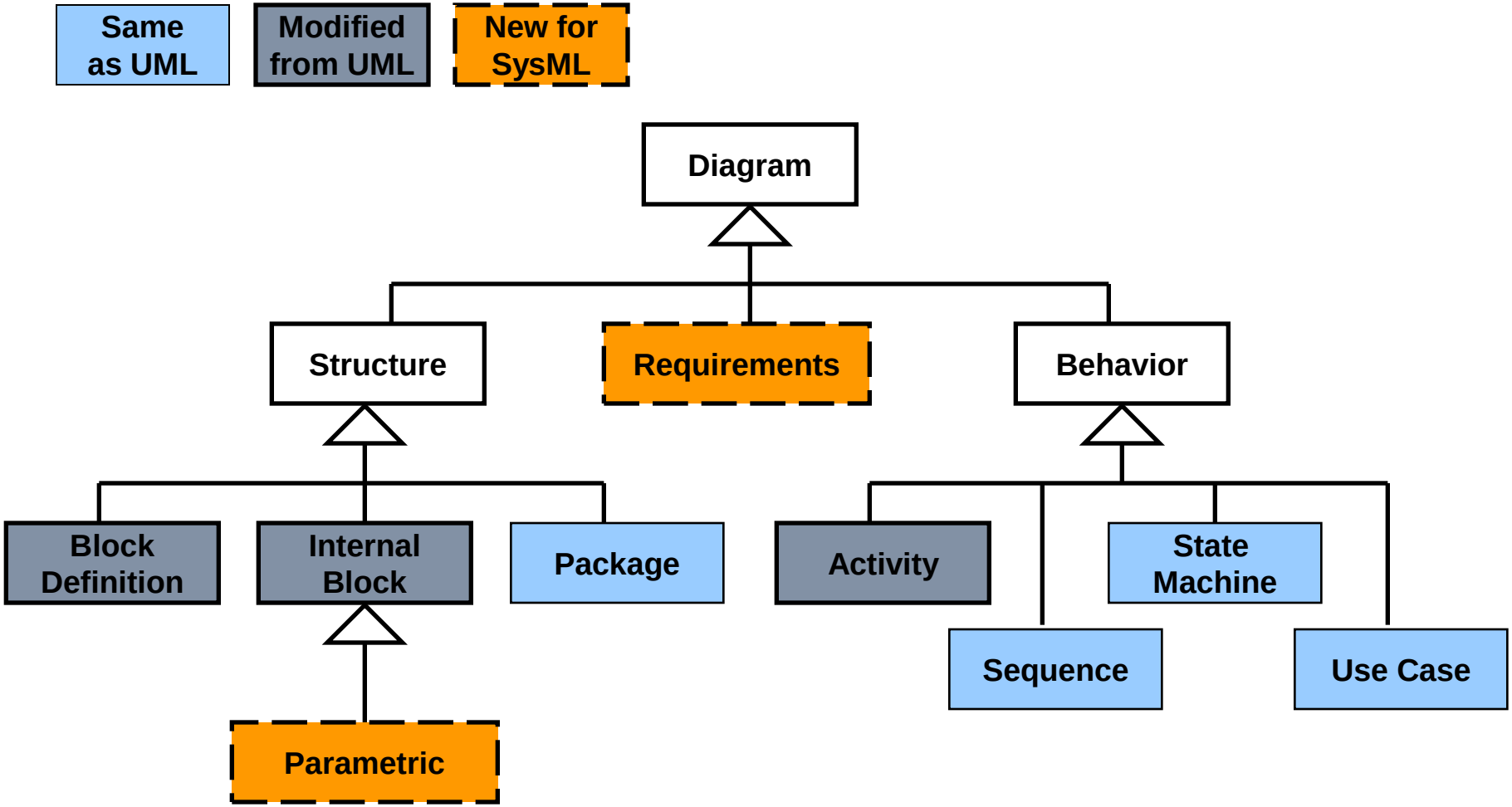
Repository-based CM – Using Private Sandboxes

- Artisan Studio includes a 3-way conflict resolution tool:
  - Including a diagram differencer for resolving conflicts



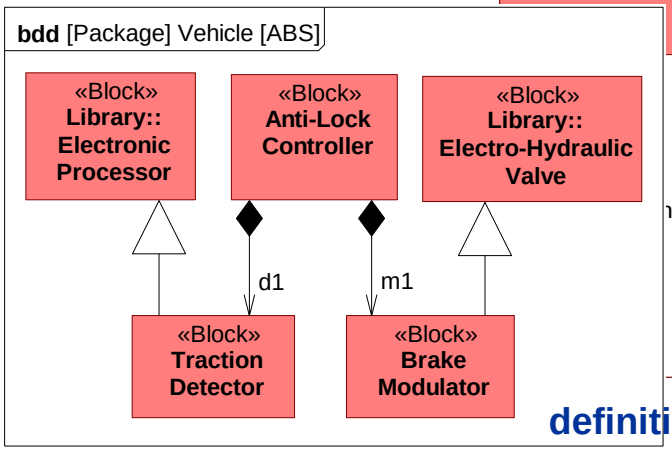
Modeling

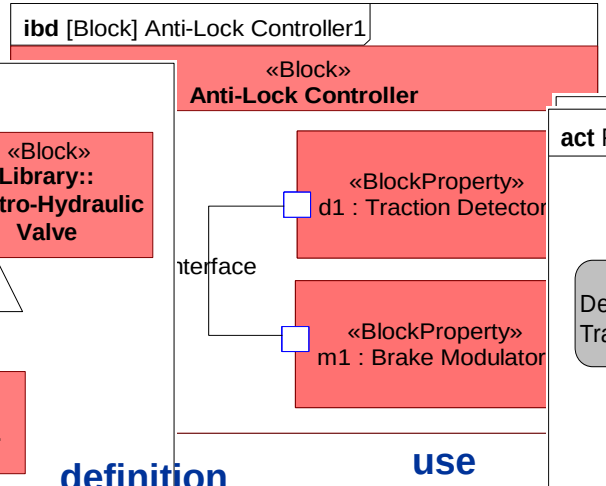
SysML (System Modeling Language)



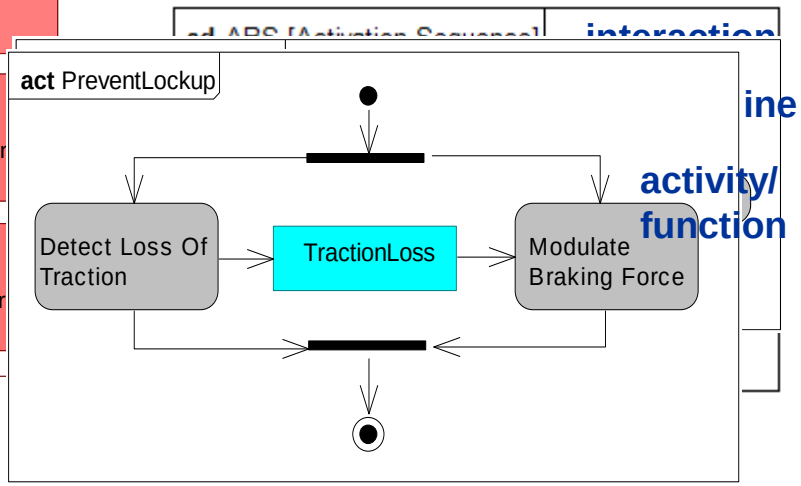
The Four Pillars of SysML

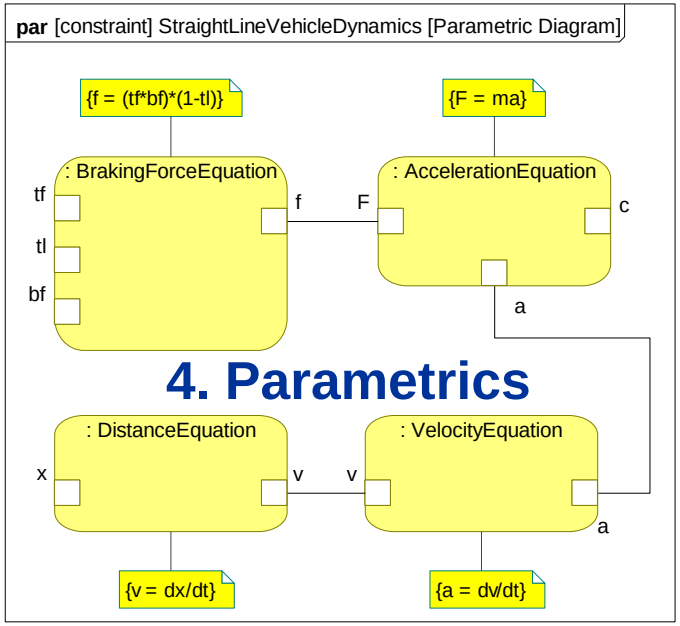
1. Structure





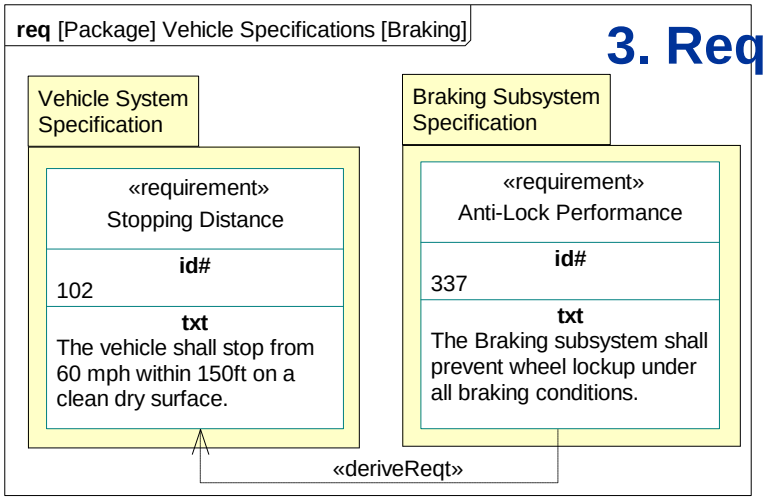
2. Behavior



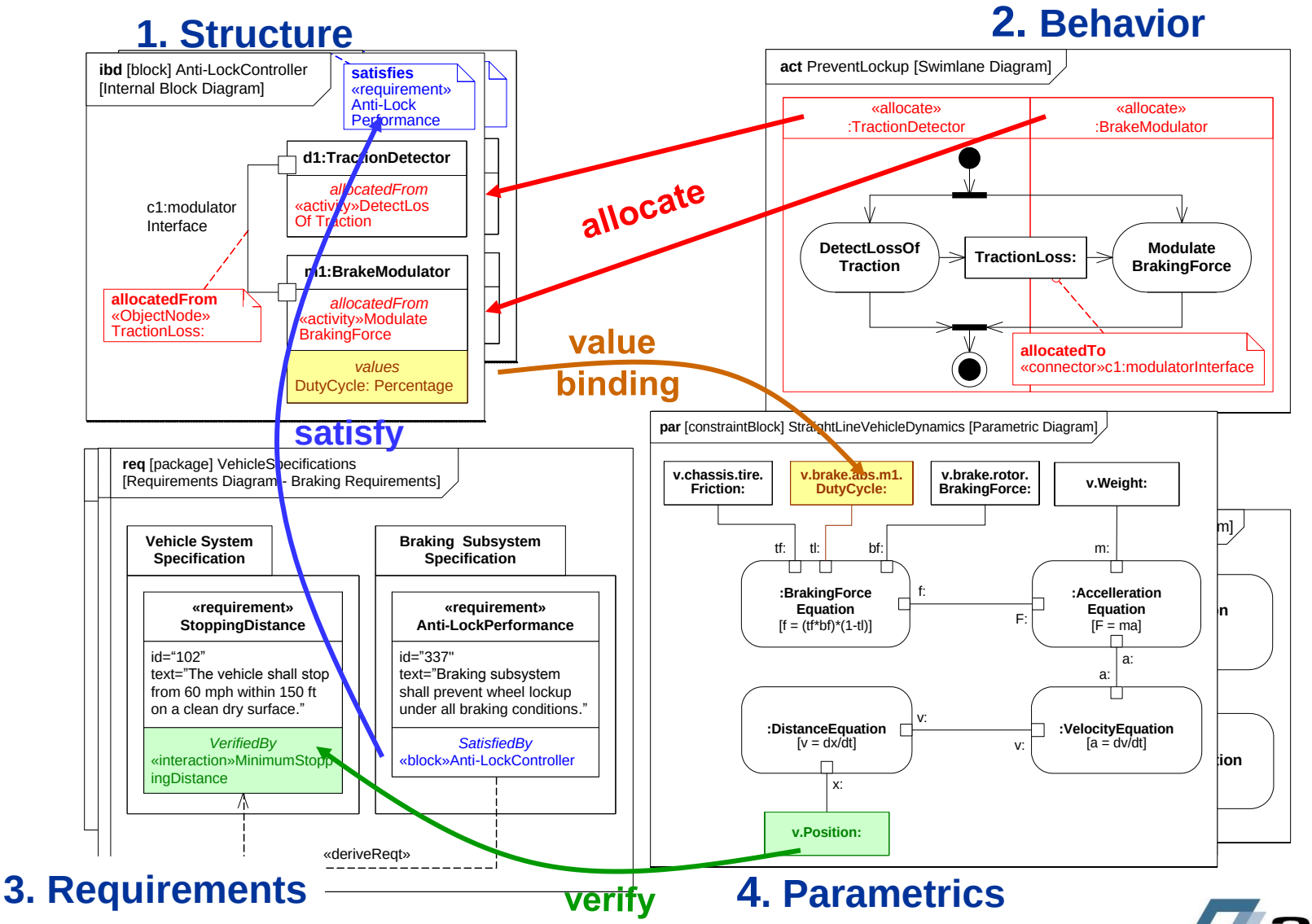


4. Parametrics

3. Requirements

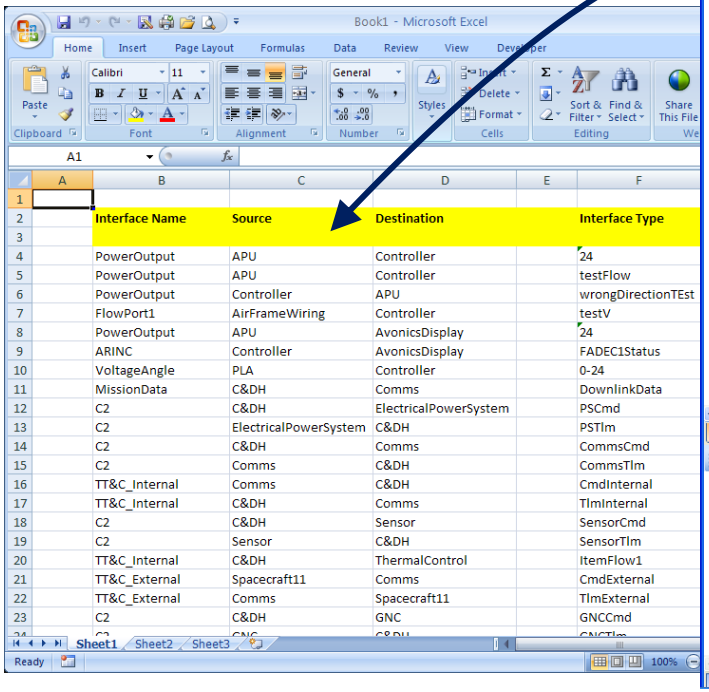


Cross Connecting Model Elements

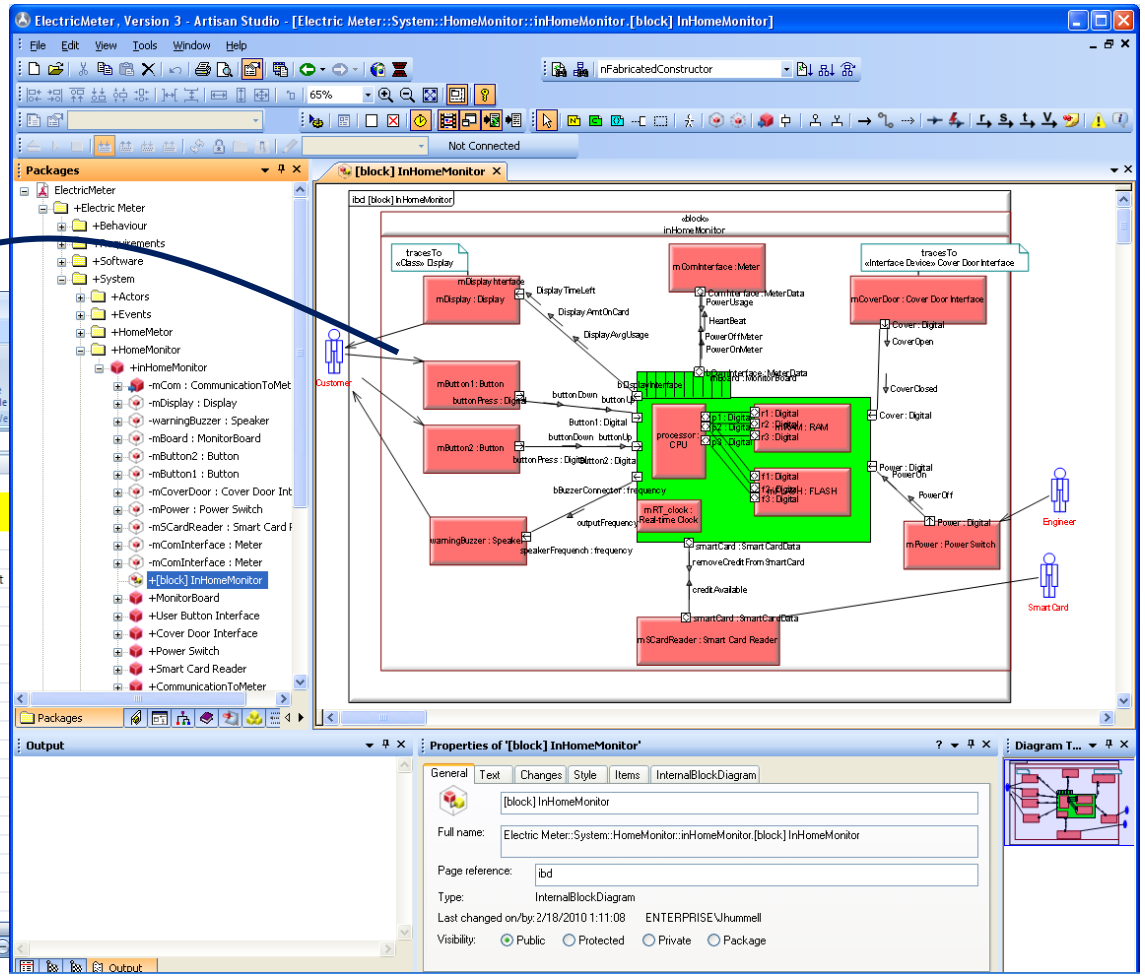


Data Mining/Data Analysis (ICD Generator)

- Generate traditional Systems documentation from modeling information

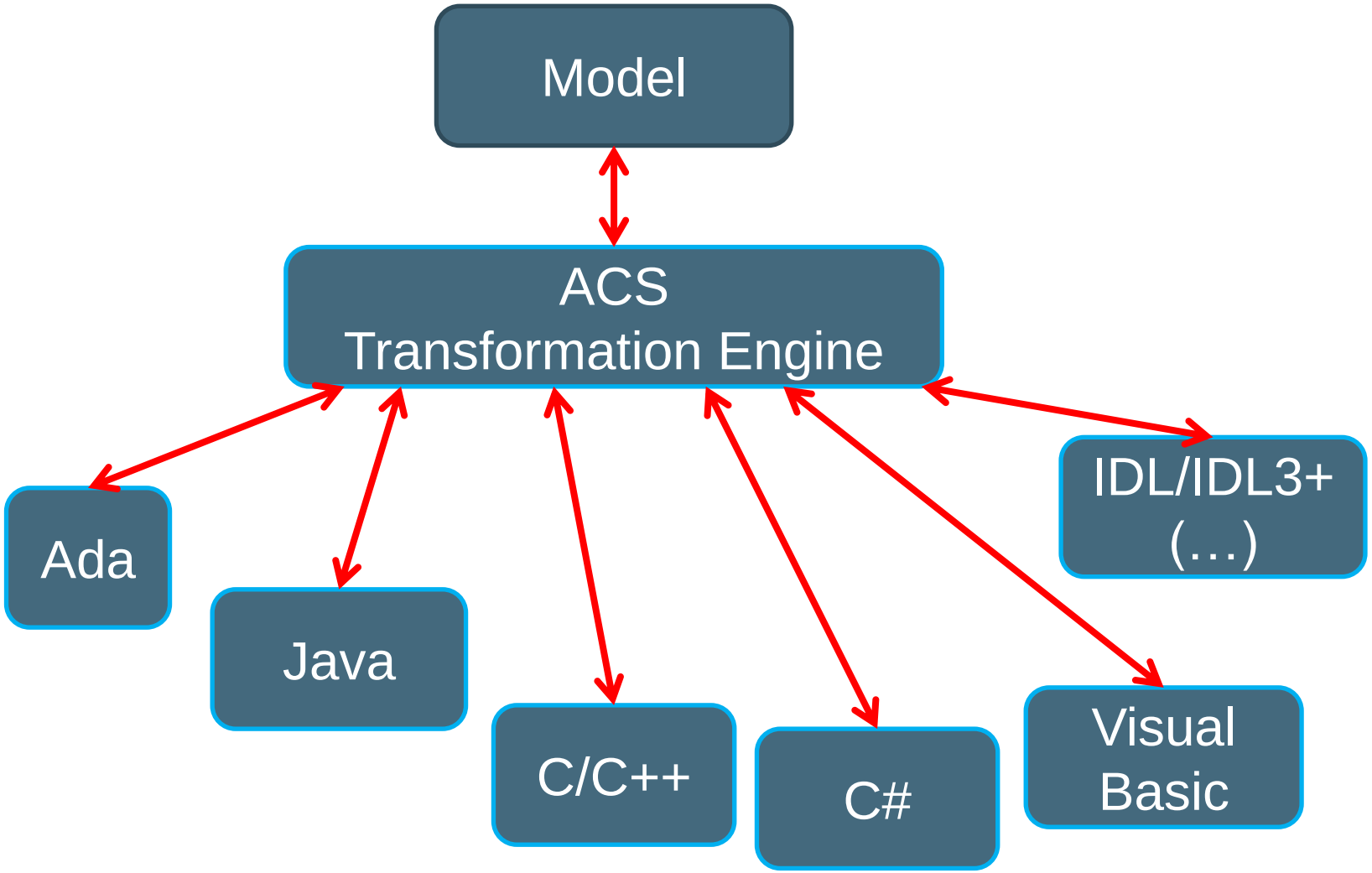


Interface Name	Source	Destination	Interface Type
PowerOutput	APU	Controller	24
PowerOutput	APU	Controller	testFlow
PowerOutput	Controller	APU	wrongDirectionTEst
FlowPort1	AirFrameWiring	Controller	testV
PowerOutput	APU	AvonicsDisplay	24
ARINC	Controller	AvonicsDisplay	FADEC1Status
VoltageAngle	PLA	Controller	0-24
MissionData	C&DH	Comms	DownlinkData
C2	C&DH	ElectricalPowerSystem	PSCmd
C2	ElectricalPowerSystem	C&DH	PSTlm
C2	C&DH	Comms	CommsCmd
C2	Comms	C&DH	CommsTlm
TT&C_Internal	Comms	C&DH	CmdInternal
TT&C_Internal	C&DH	Comms	TlmInternal
C2	C&DH	Sensor	SensorCmd
C2	Sensor	C&DH	SensorTlm
TT&C_Internal	C&DH	ThermalControl	ItemFlow1
TT&C_External	Spacecraft11	Comms	CmdExternal
TT&C_External	Comms	Spacecraft11	TlmExternal
C2	C&DH	GNC	GNCCmd



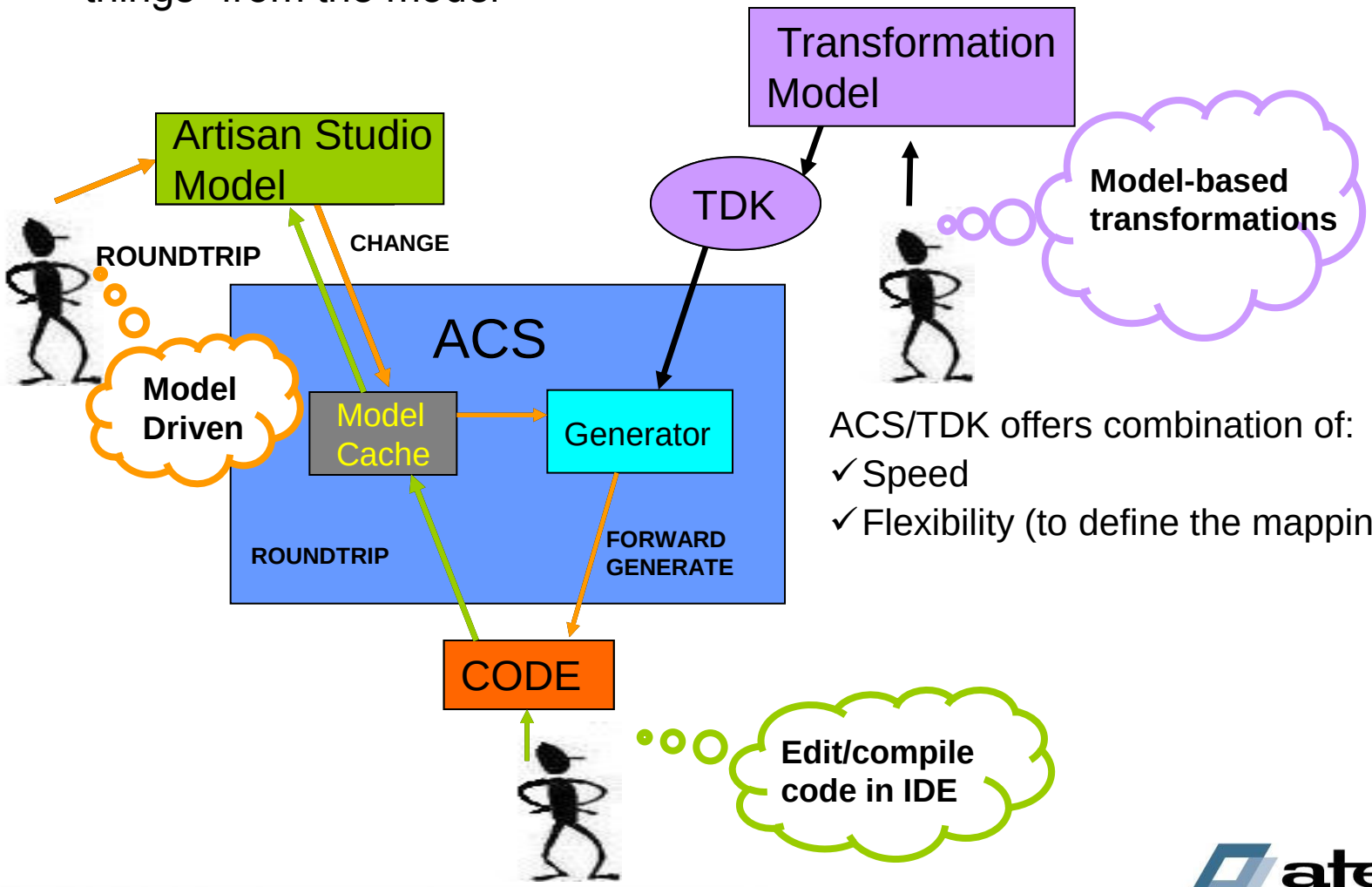
Model Execution and Code Generation

ACS Process



Why use modeling tools?

- An altogether different need is to support the automatic generation of "things" from the model

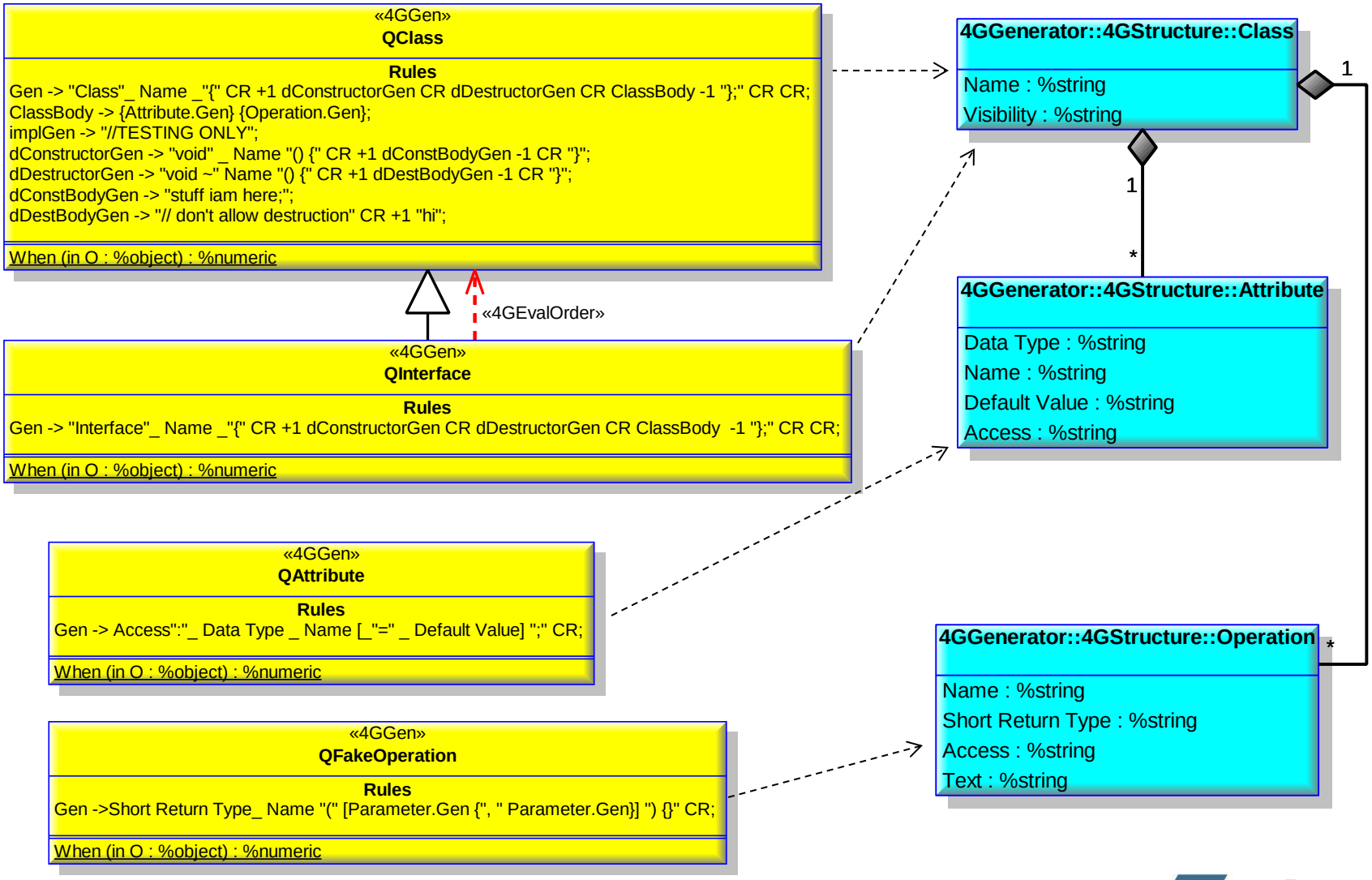


ACS/TDK offers combination of:

✓ Speed

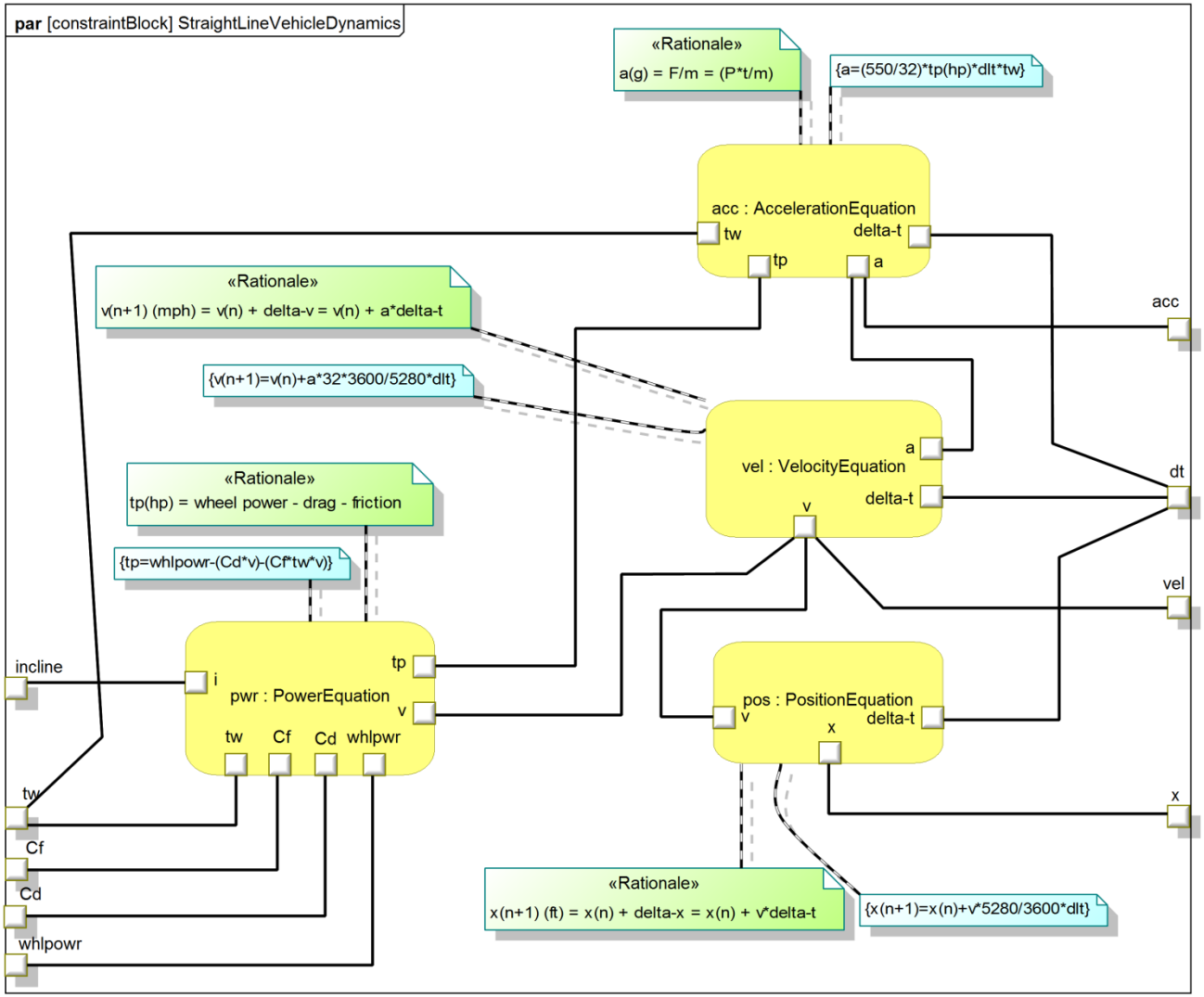
✓ Flexibility (to define the mapping)

Code Generation Customization (Transformation Model)

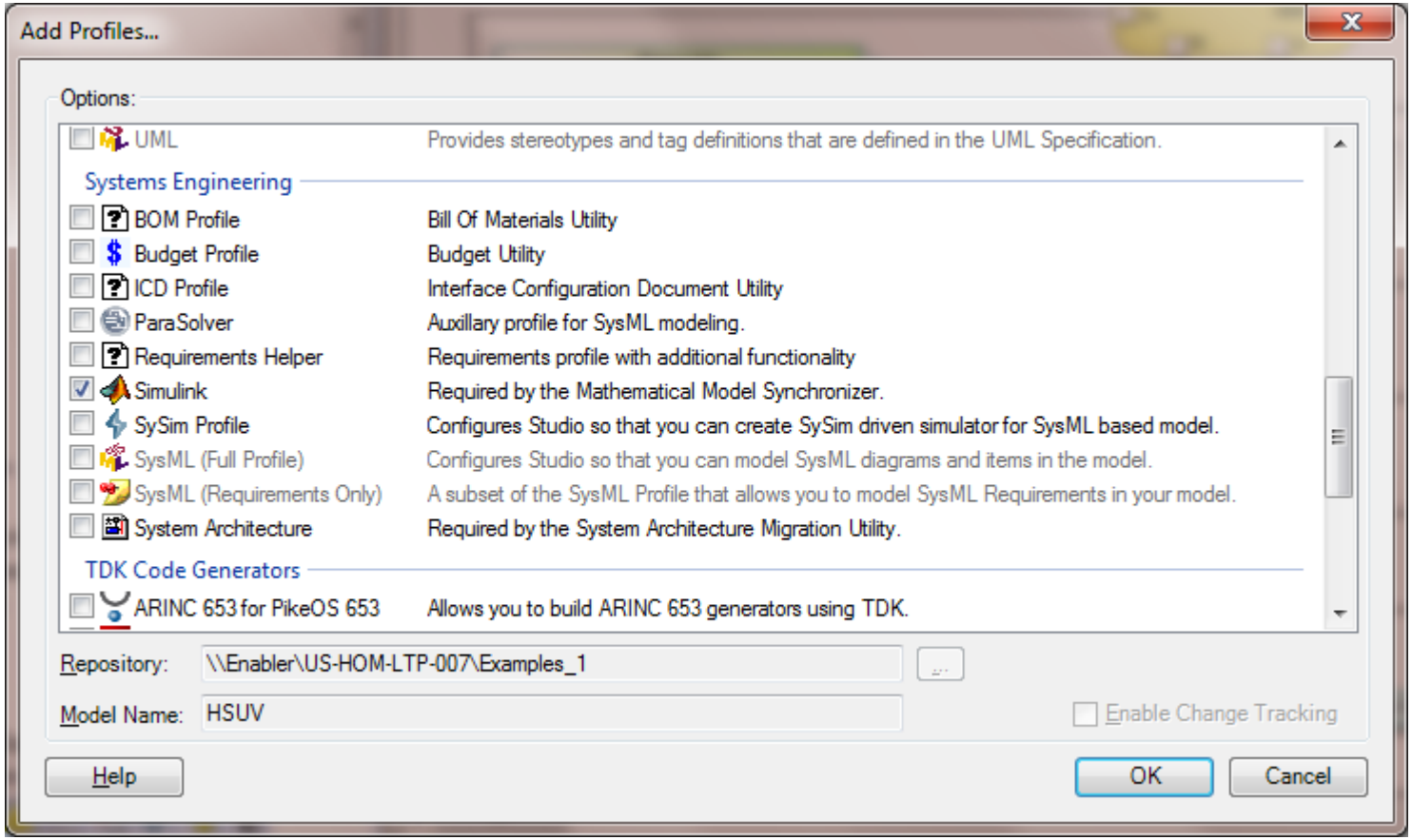


Matlab Sync

Parametric Diagram

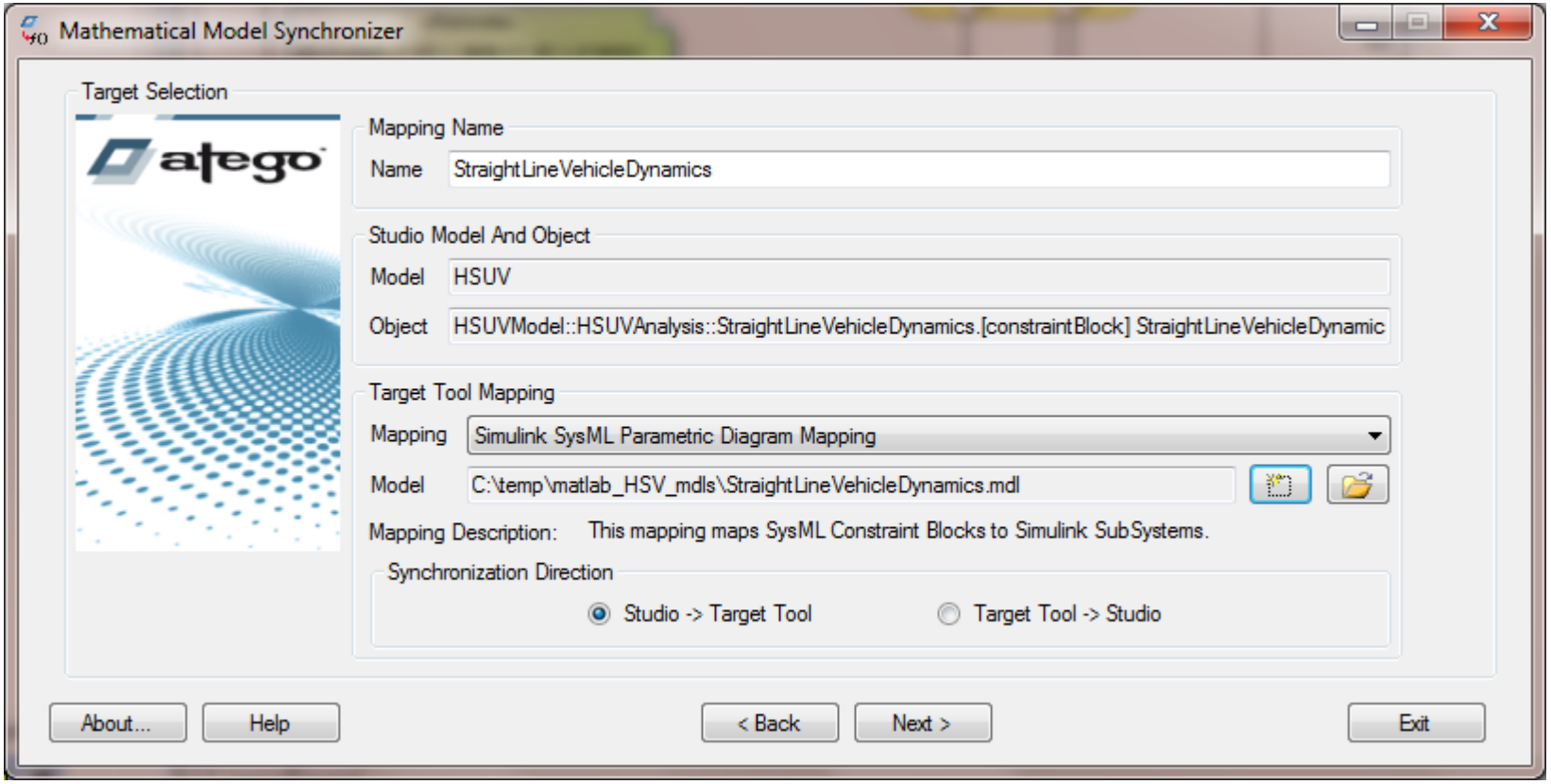


Add Simulink profile to your model

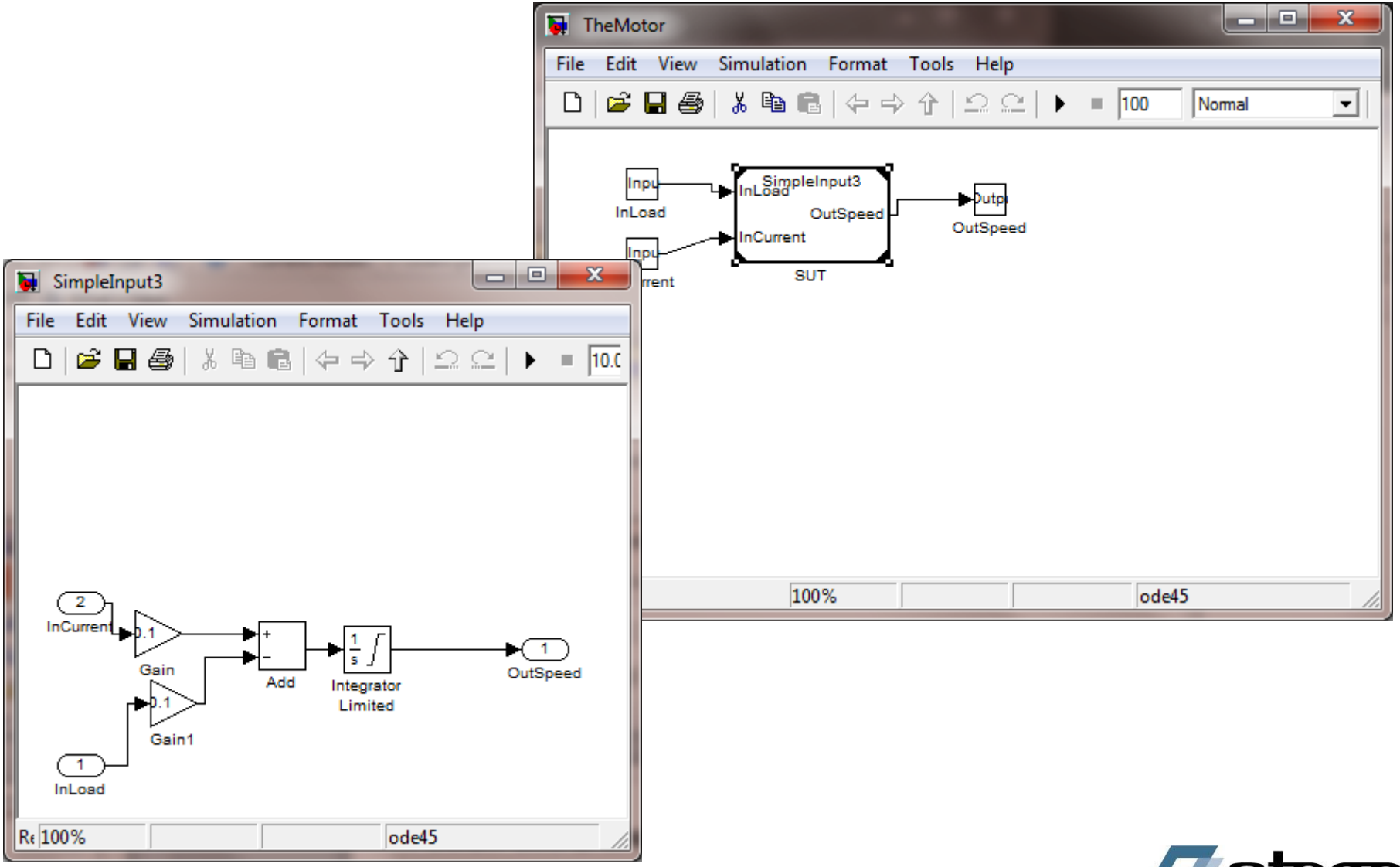


Matlab Syncronizer

- Allows mapping of types
- Creates parametric block to Matlab block mapping



Matlab sample

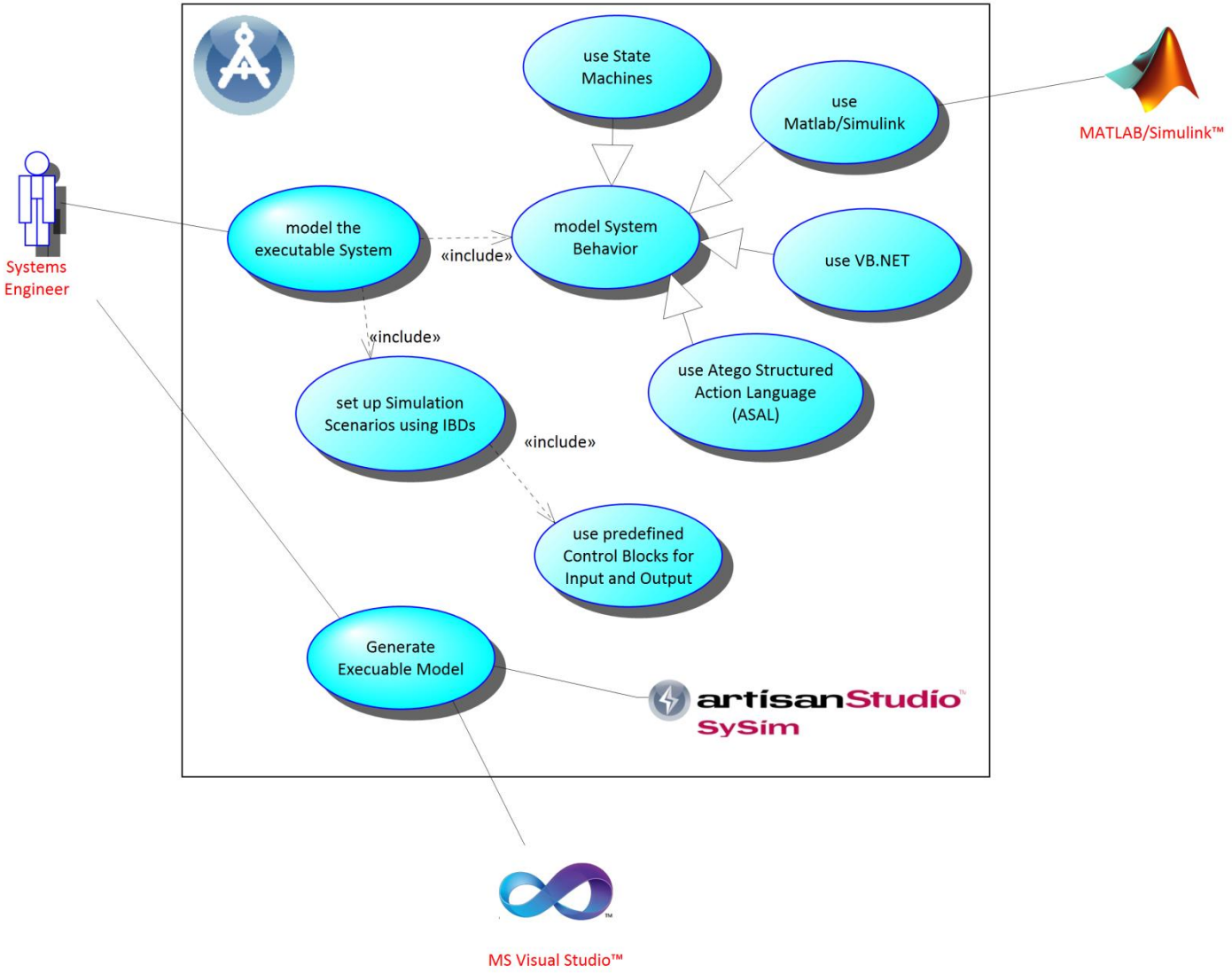


SySim

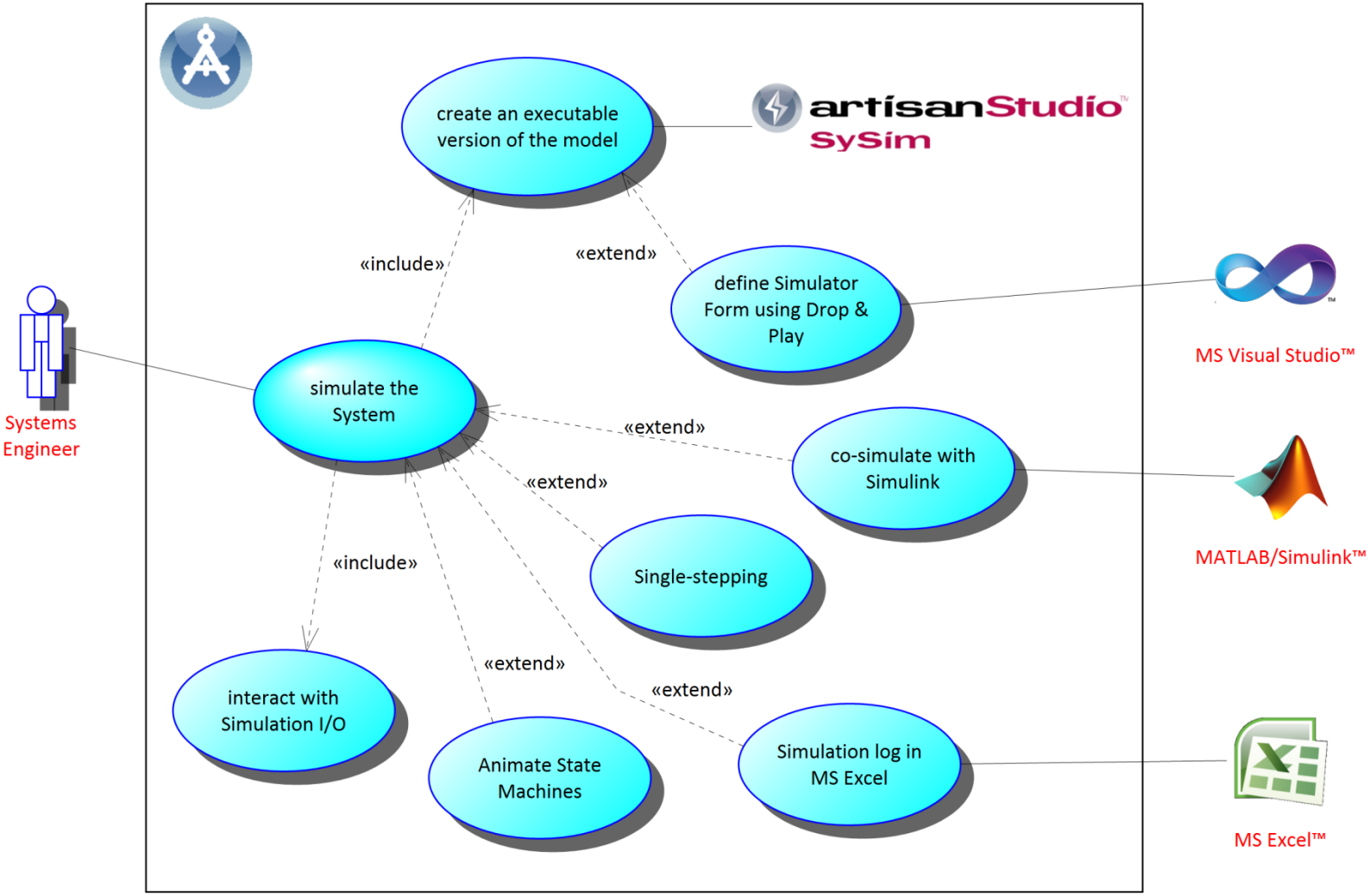
artisanStudio SySim : Model Execution for Systems Engineers

- Early Validation of Complex Behavior
  - Use your existing SysML designs for live model execution
  - Enable incremental design, starting with basic behavioral features, adding more capabilities over time
  - Enforce consistency of system models with ongoing, dynamic checks
- Block Reuse
  - Easily reuse your existing subsystems and combine them into different executable *scenarios*
  - Use *instance parameters* for differentiating the behavior of reused blocks

Executable Models: Modeling Use Cases



Executable Models: Simulation Use Cases



Executable Model Example: Speed Controller

Parasolver

What‘s Needed for System Validation Using Constraints?

artisanStudio™ ParaSolver

Typical SE Tasks

- Simulation
- Analysis
- Trade Studies
- Optimization
- Requirements Checking
- Risk Assessment
- Etc.

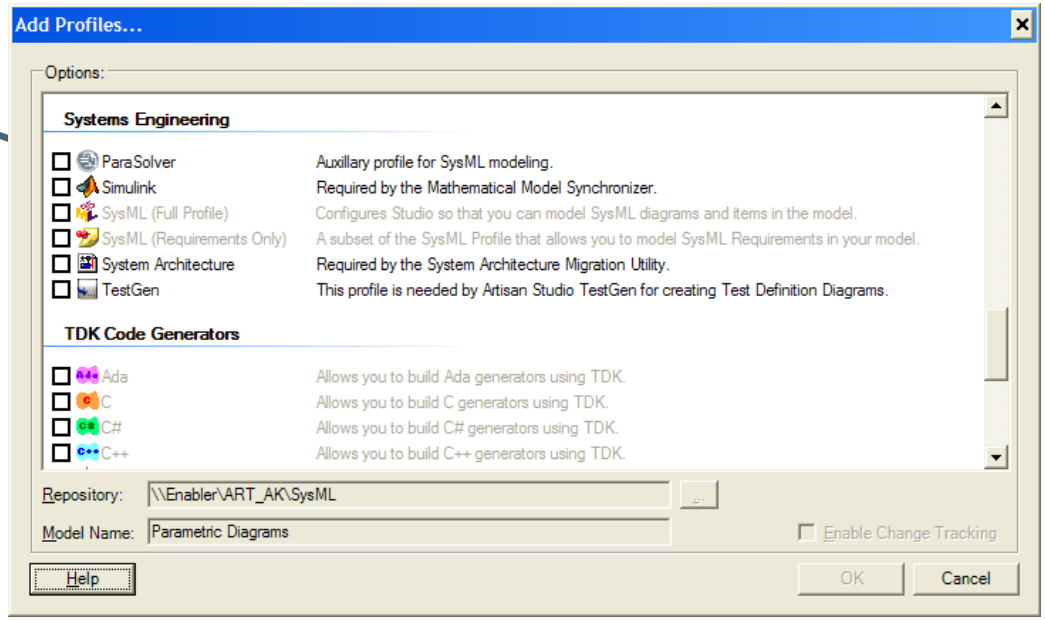
Features needed

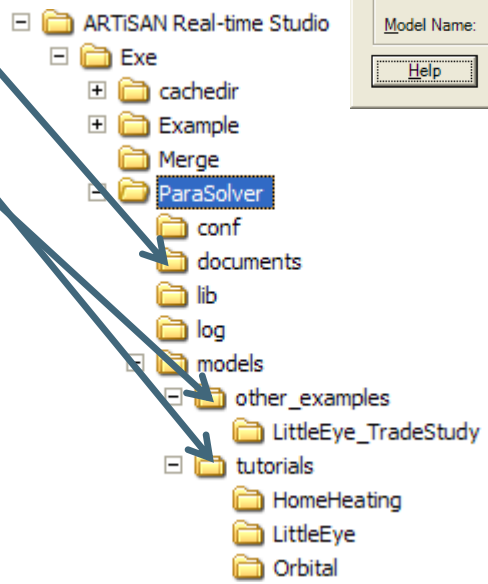
- Solvable or executable language for constraints
- SysML instance model to express design alternatives in slot values
- Solver technology, which can handle the acausal parametrics
- Input/Output to common systems engineering tools

atego™

Artisan Studio ParaSolver Parts

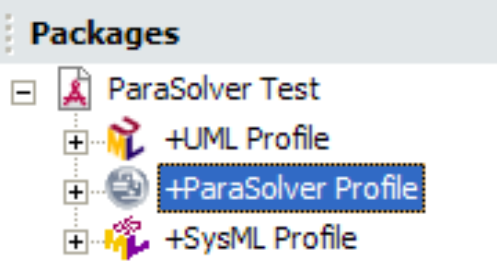
- ParaSolver Profile
- User's Guide
- Tutorial
- Example Models



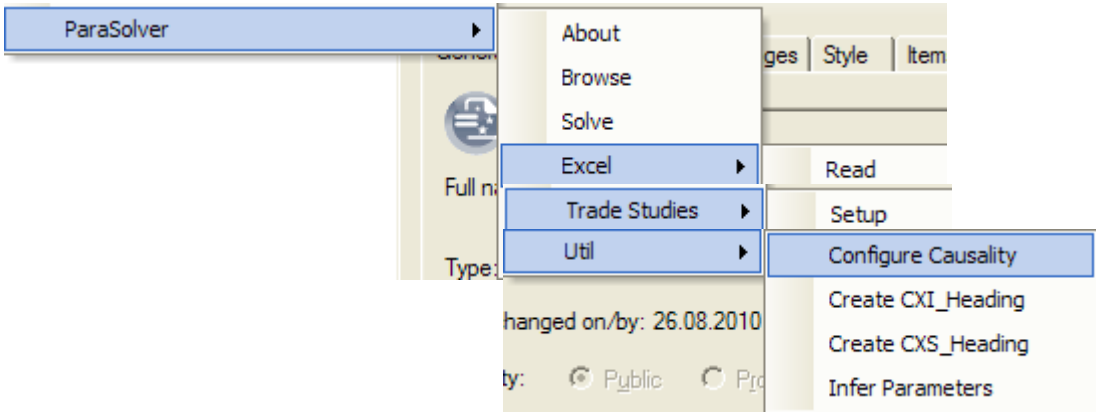


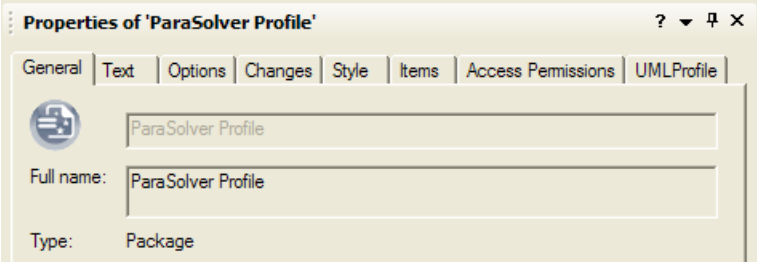
In Artisan Studio

- Importing the ParaSolver Profile causes the SysML Profile and the UML Profile to be included as well



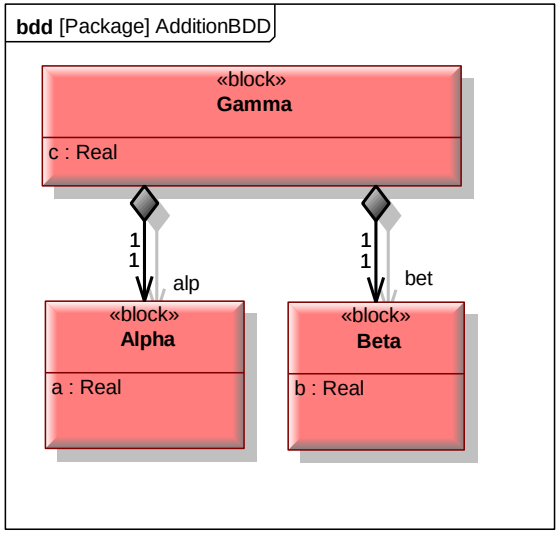
- The Tools Menu shows the ParaSolver functions

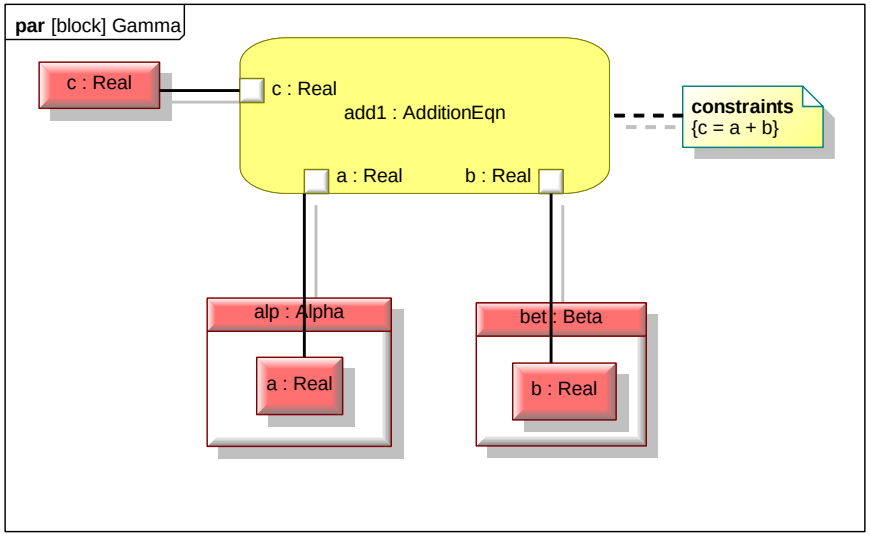


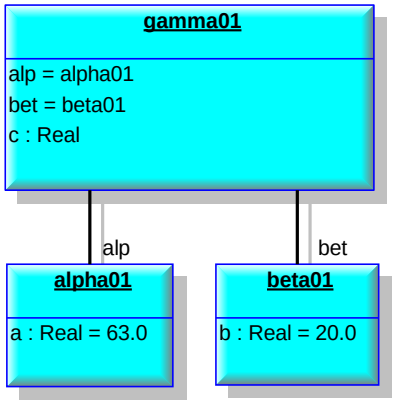


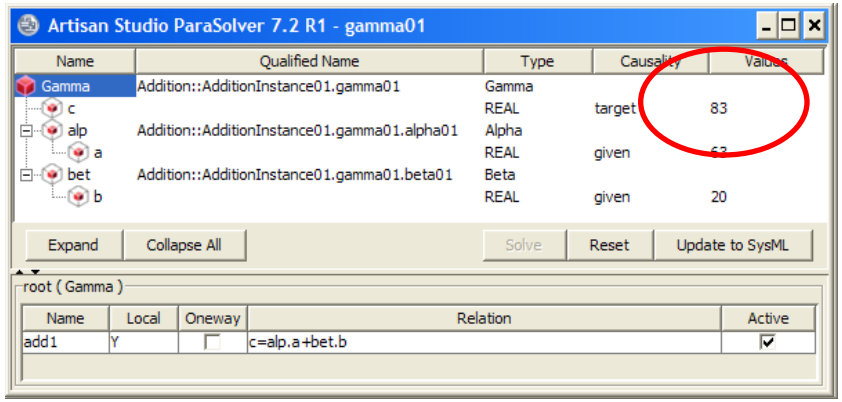
A Simple Example

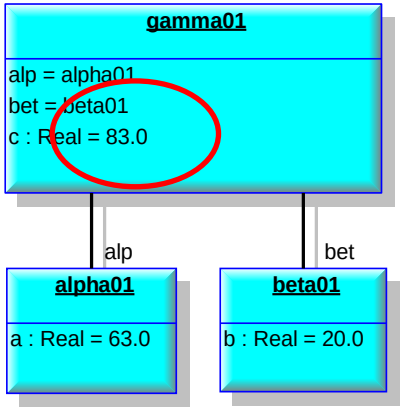
- bdd
- par
- od











Don‘t Forget...

- SysML Parametrics are acausal, so you can calculate

  c = a + b with given a and b or

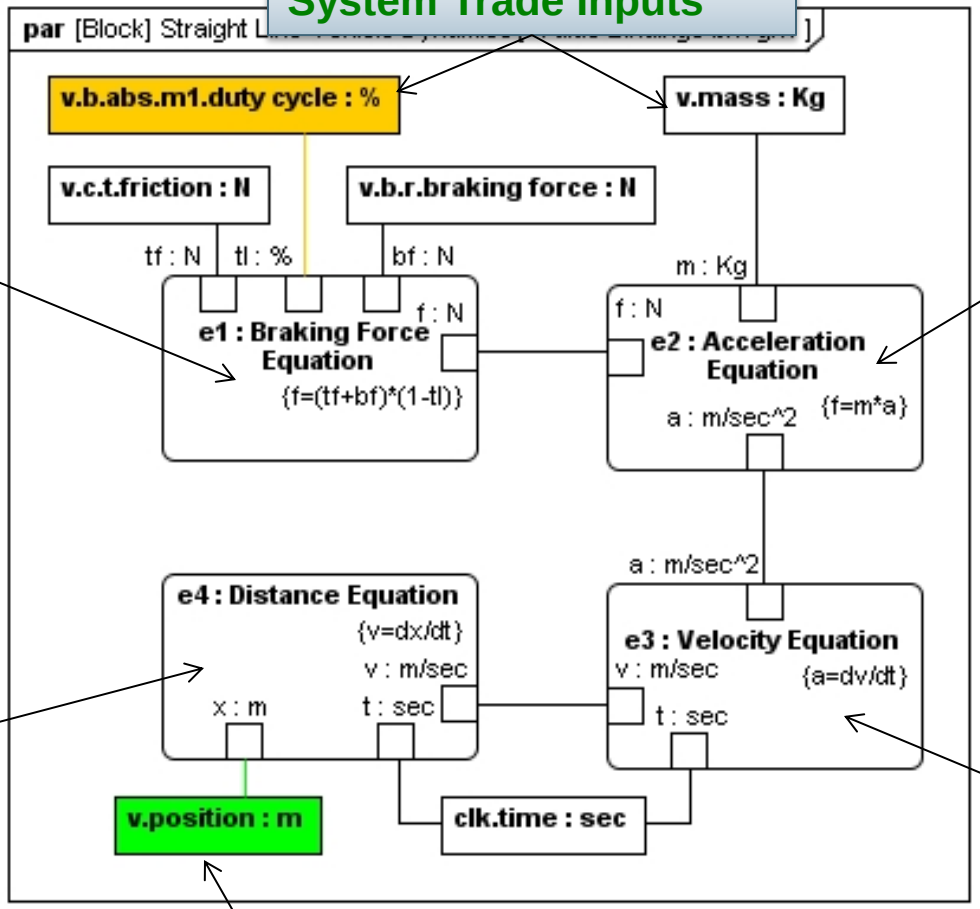
  a = c – b with given c and b or...

- WITHOUT changing anything in the model
- Of course this is a VERY simple example, but this works also for more complex ones, of course
- The Instance model holds all calculated values, enabling the model to contain more than one alternative

SysML Parametric Model Execution: Orchestrating External Solvers & Trade Studies

artisanStudio ParaSolver

System Trade Inputs



Simple math relation

MATLAB/Simulink model

Excel

Mathematica function

Key System MoE

+ acausal modeling

atego

Current and Planned Integrations

- Mathematica
- OpenModellica
- Matlab
- Matlab/Simulink
- Excel

- Planned:

  CAD/CAE Systems, e.g. Catia

Demos

- Satellite Power and Weight Analysis

- Little Eye UAV – Area Covered Analysis

Artisan Studio Benefits

- SysML benefits:
  - Requirements documentation and traceablity to view the system for completeness and adherence to the customer needs.
  - Express and document your design decisions.
- Parametrics benefits:
  - Build a mathematical model of the solution to analyze and to verify requirements against.
- ParaSolver benefits:
  - Enables the systems engineer to run instances of their design decisions against the parametrics models created.
  - Can run multiple scenarios to figure out the right solution.