# Modeling Tool Integration Plugins (MTIP)

January 31, 2022

Trent Severson The Aerospace Corporation Karina Martinez The Aerospace Corporation

Approved for public release. OTR 2022-00282.

© The Aerospace Corporation, 2021

### **Project MTIP Team Introduction**

#### Karina Martinez

- Software Tools and
   Assurance Department
- SparxEA tool developer



#### **Trent Severson**

- MBSE Office
- Cameo tool developer



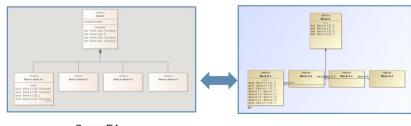
#### **Joel Thomas**

- MSD
- Cameo tool developer



## Agenda

- Definitions
- Problem Overview
- Approach to MBSE Tool Integration
- Examples
- Code Base and Usages
- Demo
- What's Next?



Sparx EA

Cameo





### Definitions: Ontology, Metamodel, SysML, UML

### Ontology

- Defines entities and relationships
- Helps introduce a sharable and reusable knowledge but can also add new knowledge about the domain.
- Metamodel
  - A set of rules that define valid construction of a model
  - Allows a model to be checked for syntactic correctness using rules and constraints defined by the metamodel
- Systems Modeling Language (SysML) Metamodel
  - A general-purpose graphical modeling language for specifying, analyzing, designing and verifying complex systems
  - Provides graphical representation with a semantic foundation for modeling system requirements, behavior, and structure.
- Unified Modeling Language (UML)
  - A standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing and documenting the artifacts of systems.

### Definitions: Cameo, Sparx EA, XML, XMI

- Cameo Systems Modeler
  - A collaborative Model-Based Systems Engineering (MBSE) environment, which provides robust, and intuitive tools to define, track, and visualize all aspects of systems in the most standard-compliant SysML models and diagrams.
- Sparx Enterprise Architect (EA)
  - A collaborative modeling design and management platform based on UML and related standards
- eXtensible Markup Language (XML)
  - A markup language designed to store and transport data.
- XML Metadata Interchange (XMI)
  - An XML-based integration framework for the exchange of models, and, more generally, any kind of XML data.
  - Used in the integration of tools, repositories, applications, and data warehouses.
- Huddle Unified Data Schema (HUDS)
  - Allows data to be easily shared between different plugins while still adhering to a custom set of semantics.
  - More human readable than XMI

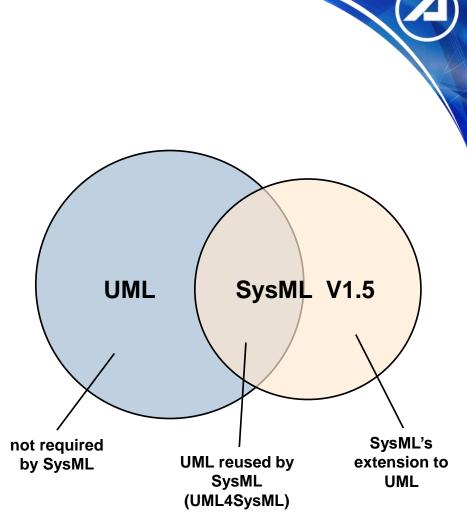


# Integration of MBSE Tools

## **Problem Overview**

Cameo and Sparx EA Interoperability

- Commercial tools are not sufficient to work with models that do not fall within UML standards
- XMI import/export is the current standard interchange format between MBSE tools.
- XMI format adheres to UML standard and does not account for additional architectures.
- SysML specific data is not transferred between tools
  - Only UML4SysML elements are transferred correctly



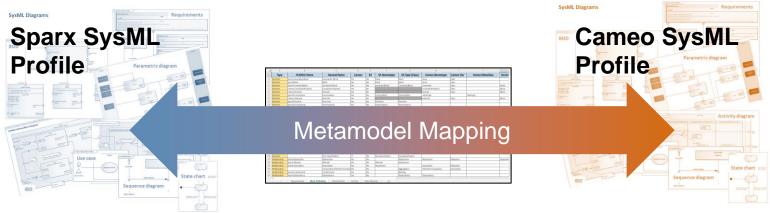
Relationship Between SysML and UML[1]

#### Current XMI standard insufficient for transferring SysML models

## Sparx EA ←→ Cameo

Approach to Tool Integration for SysML Metamodel

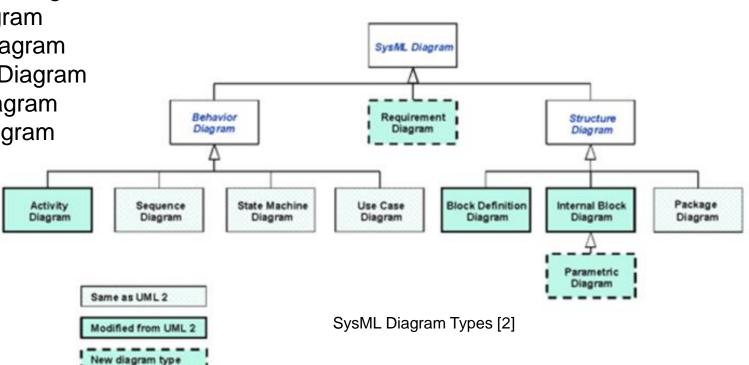
- Project Goal
  - Create software plugins and exchange standards to enable 1-to-1 translation from Sparx EA to Cameo and back
  - Create a way for SysML models to be exchanged freely between tools
- Implementation Approach
  - Metamodel Mapping
    - Map SysML metamodel in Sparx EA and Cameo to a common schema
  - Data imported/exported via plugins using Huddle Unified Data Schema (HUDS)
     V2 XML format
- Current Capabilities
  - Transfers SysML models from Sparx EA into Cameo and back
    - Includes all elements, relationships and diagrams



## **Diagram Types**

Organization of the SysML Metamodel

- Block Definition Diagram
- Internal Block Diagram
- Activity Diagram
- State Machine Diagram
- Package Diagram
- Parametric Diagram
- Requirement Diagram
- Sequence Diagram
- Use Case Diagram



# Graphic nodes and paths for diagram types listed above are supported in MTIP plugin translation

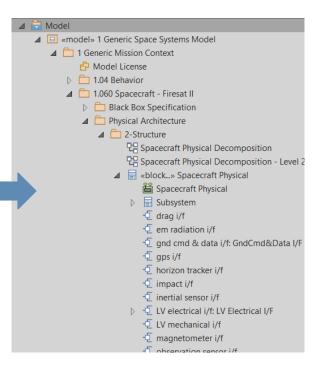
## HUDS V2 XML Format Cameo ←→ HUDS V2 XML ←→ Sparx EA

#### Cameo



### HUDS XML

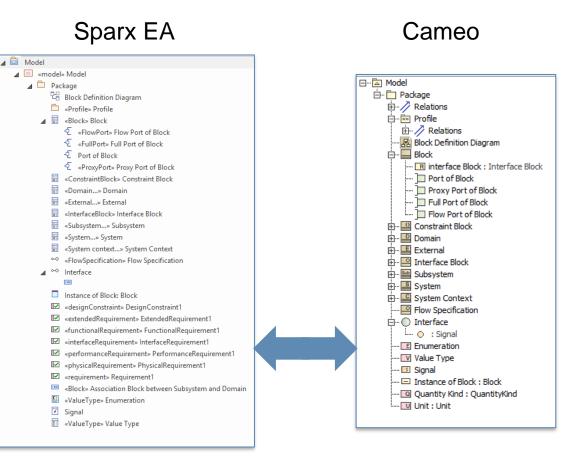
### Sparx EA



Each element has its own data block. Each data block consists of type, id, attributes, and relationships required during MTIP plugin translation

### **Block Definition Diagram Package Structure**

- Diagrams
  - Block Definition Diagram
- Elements
  - Port
  - Full Port
  - Proxy Port
  - Flow Port
  - Interface Block
  - Flow Specification
  - Constraint Block
  - Domain
  - Subsystem
  - External
  - System
  - System Context
  - Value Type
  - Enumeration
  - Signal
  - Instance Specification
  - Interface
  - Property
- Relationships
  - Interface Realization
  - Generalization
  - Association Block
  - Directed Association
  - Directed Aggregation
  - Directed Composition
  - Association
  - Item Flow
  - Usage



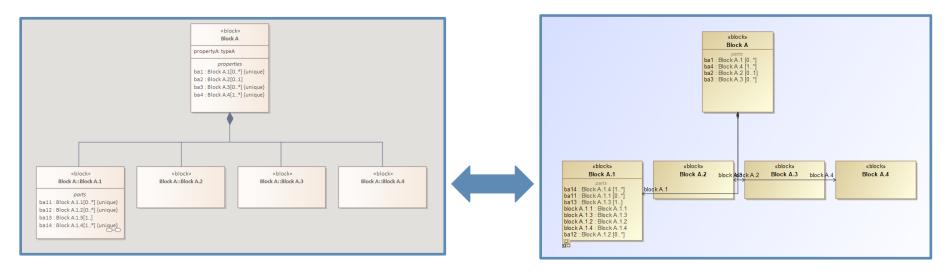
Side by side comparison of project browser/containment tree of a translated model in EA (left) and Cameo (right)

#### 11

# **Block Definition Diagram**

Sparx EA  $\leftarrow \rightarrow$  Cameo

Block Definition Diagram



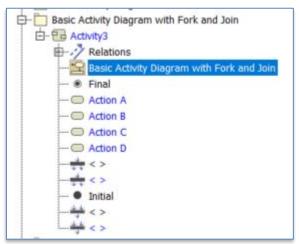
Sparx EA

Cameo

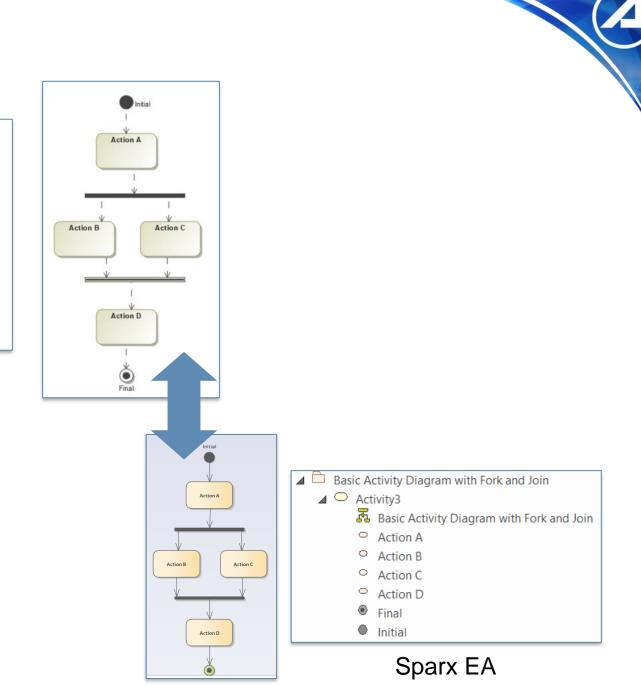
- Diagram information captured by the plugins:
  - Which elements and relationships appear on the diagram
  - Position of the elements on the diagram

### Path styling, colors, and other formatting not translated

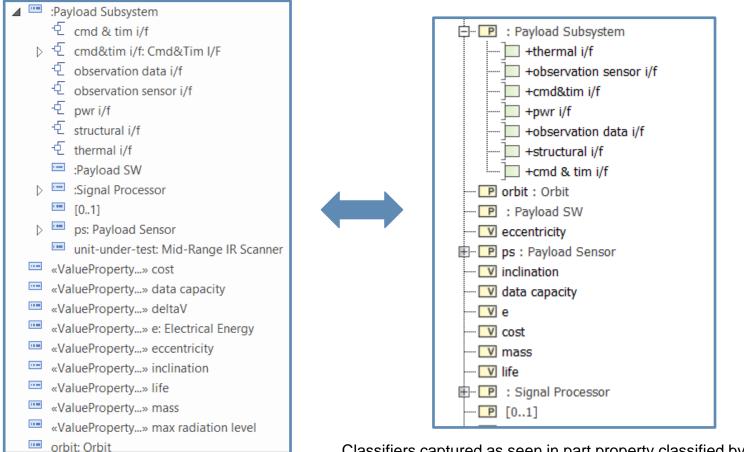
### Activity Diagram



Cameo



### **Classifiers and Types**



Classifiers captured as seen in part property classified by Payload Subsystem block. Shows inherited ports

### **Projects Benefiting from MTIP Plugins**

- Metamodel support
  - Establishing a metamodel for corporate-wide use
  - Developing tooling to check for consistency and to ease use of a standardized metamodel
- Exporting behavioral data for simulation
  - Translating structural and behavioral diagrams based on a mission into an external tool's simulation scenario templates
- Mapping to reference models
  - Exporting a Government Reference System Model in SysML from a collection of ASOTs including enterprise model in SysML, requirements, conceptual design, etc.
- Interfacing with classification tool
  - Aerospace project that is developing a capability to apply classification markings to a model's elements and relationships based on rules derived from Security Classification Guides
- Supporting SysML model translations for in-tool simulations
  - Brought existing simulation ran with Sparx EA's simulation engine into Cameo for team to run using Cameo's Simulation toolkit

## Code Base and Usage

Overview

Cameo Plugin	Sparx EA Plugin
Apache License 2.0	Apache License 2.0
Written in Java	Written in C#
~13,000 SLOC	~10,000 SLOC
~400 classes	~50 classes

### Installation and Usage

- Cameo Plugin is packaged in compressed (zipped) folder
  - Imported via the Resource Plugin Manager in Cameo Systems Modeler
  - Accessed from the top menu bar of the application
- Sparx EA plugin can be installed directly into the application using installer executable
  - Accessed in the "Specialized" tab in Sparx Enterprise Architect
- Plugins import HUDS V2 XML file created on export of the model



### Sample Model Translation Demo Video

				K
	MTIP_Sample_Model - Enterprise Architect			- & ×
Start Design Layout Develop Pub	ish Simulate Specialize Construct Execute Configure 🖓 Find Command			💿 Perspective 🍷 💄 User
Search Portals Explore Design Share Run Workspaces Perspectives Desktop	Image: Style Preferences       Image: Style Pr			
1 🔄 🧿 ▶ / ▶ Model ▶			Find Package	e 🔎 🏴 🗏
Browser 💌 🖣 🗧	· »	• x	Properties	▼ ₽ 3
2 🖻 혐 🕇 ∔ 🌊 ≡ - 🔹 📦			⊟ = • 🖻 🕨	
Project Context Diagram Resources			Element General Name	
<ul> <li>Generic Mission Context</li> <li>Model License</li> <li>1.04 Behavior</li> </ul>			Type Stereotype Alias	
<ul> <li>Perform Mission</li> <li>Control Trajectory</li> <li>Deploy Mechanisms</li> <li>Launch S/C</li> </ul>			Keywords Status Version	
Maintain Spacecraft Operations			<ul> <li>Model (from Sysl</li> <li>Package</li> </ul>	AL 1.3 )
Perform Mission			URI	
Provide Observation Data			Visibility	Public
<ul> <li>Separate from L/V</li> <li>1.060 Spacecraft - Firesat II</li> <li>1.10 Supporting Elements</li> <li>Custom Stereotypes</li> <li>≪stereotype» analysis</li> <li>≪stereotype» caused by</li> <li>≪stereotype» failure mode</li> <li>«stereotype» general category</li> <li>«stereotype» mechanical</li> <li>«stereotype» orbit</li> <li>«stereotype» tore</li> <li>«stereotype» test component</li> <li>«stereotype» tpm</li> </ul>	₽ Features	▼ # ×	Project	
	Attributes Operations Receptions Parts / Properties Interaction Points			
Browser Inspector			Properties Toolbox M	lotes

### Containment Tree Side-by-Side 1

Huddle_Sample_Model - Enterprise Architect	×	🛒 Cameo Systems Modeler 19.0 - HUDDLE EA Import.mdzip [C:\Users\tas30828\Documents\]	- 🗆 🗙
⊗ * Start Design Layout Specialize Publish Construct Simulate Code Execute Config	gure 💿 Perspective * 💄 User	File Edit View Layout Diagrams Options Tools Analyze Collaborate Window MBSE Office MBSE Translator Help Huddle	×
🔥 🛼 🛃 🔍 💀 🚣 Full Screen 🛛 🛼 拱 📰 💷	My Kanban 🔗 🔗	🗈 🖻 🚰 📲 🚨 🦓 🔸 🦘 🔸 🥙 Test Update Requirements and KPPs Reload	💽 🔹 🔀 Create Diagram
Portals Browse Properties Trace Search	My Gantt     Heln	B Containment to Diagrams B Structure	🔀 Spacecr 📢 🛙
🔹 👻 👻 🔹 👗 🔹 🔹 Library Journa	- 99	Containment @ 9	
Show Explore View Collaborate	Help	≪ ¢ ≝ ☆ Q <b>¢</b>	Selection
G 🕤 🕨 / 🔸 Model 🔸 1 Generic Space Systems Model 🔸 1 Generic Mission Context 🔸 1.060 Spacecraft - Firesat II 🕨	Value Types 🔎 📂 🚍	- Model	
Project Browser • * ×	»: Portals	E-1.10 Supporting Elements	Tools
	Perspective -	i - i 1.060 Spacecraft - Firesat II i - i Black Box Specification	요 속 죠 ㅋ
Global Context			Common
	✓ UML	E- Relations	Block Definition Dia
🔺 💼 Model	AILUML	- LV electrical i/f «Block»	Block
▲ 🔲 «model» 1 Generic Space Systems Model	Structural	È-	Interface Block
A 🛅 1 Generic Mission Context	Behavioral	E/ Relations	Flow Specifica
😚 Model License	Strategy	Ty probability of detection	Constraint Block
▲ 1.060 Spacecraft - Firesat II	Analysis		
▲ ☐ Black Box Specification	Requirements		Domain 👻
▲  Click Spacecraft Black Box	▷ UX Design		Value Type 👻
't⊖ Spacecraft Black Box ▲ 📓 «block» Spacecraft		Iffe	E Enumeration
∠ drag i/f	Business Modeling		3 Signal
-2_ drag //r -√2_ em radiation i/f	D Software Engineering		Instance
√_ gnd cmd & data i/f: GndCmd&Data I/F	Systems Engineering	I pointing accuracy	O Interface
√g gps i/f	SysML		] Port
	UAF		] Proxy Port
-€ impact i/f	UPDM		] Full Port
5 inertial sensor i/f	Executable State Machine	W maxradiation level     Junt racker i/f	] Flow Port
b √2 LV electrical i/f: LV Electrical I/F	Database Engineering	borizon tracker i/f	🖉 Interface R 👻
- 🖞 LV mechanical i/f		) gps i/f	/, Link
▷ - 1/2 magnetometer i/f	Enterprise Architecture	UV electrical i/f: LV electrical i/f	🚔 Association 👻
b 4 observation sensor i/f	Information Exchange	Observation sensor i/f     On d & data i/f	ᡚ Internal Block Diag
-⊊ solar radiation i/f	✓ Construction	- is star tracker i/f	Requirements Diag
√⊈ star tracker i/f	Project Roadmap	) inertial sensor i/f	R Requirement
4⊈ sun tracker i/f	Kanban		E Extended Reg
- thermal radiation i/f	Document Publishing		Satisfy
- 🖞 thrust i/f	Dashboards & Charts	— D em radiation i/f	
⇒ collect observation data()	Reviews & Discussion		Derive
control acceleration()	Construction View	impact i/f	Copy
= v control attitude()		🔚 🖢 🖿 thermal radiation i/f	7, Trace
=Q control separation()	Test Management	⊡-⊡ 3-Behavior	Verify
<ul> <li>control thermal environment()</li> <li>deploy antenna 0</li> </ul>	Change Management	🖆- 🛅 Spacecraft Failure Modes	✓ <sup>n</sup> <sub>8</sub> Refine
=	Resource Management		🔞 Test Case Acti 👻
⇒v deploy solar array() ≪v manage faults()	Legends	E- Control Acceleration(, )(context Spacecraft)	💫 Use Case Diagram
<ul> <li>⇒ manage faults()</li> <li>⇒ provide electrical power()</li> </ul>	▷ Management	住・役 Control Attitude(, )(context Spacecraft) 一 役 Control Separation(context Spacecraft)	C Profiling Mechanism
⇒ provide electrical power() ⇒ provide structural integrity()	My Perspectives		v 🔺
provide schedular integrity()     v     provide telemetry data()			
= v receive ground command()		S Notification Window	
= v receive ground command() = v return observation data()		Notification Window	t⊊ ¢ ×
w ValueProperty» cost		X 合 8 届 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
«ValueProperty» data capacity		6F01EC3C-FC00-43b1-9D6D-032BCFA25EB3and cameo id_19_0_2_c04024e_1569869226376_211871_42568	^
«ValueProperty» deltaV		Log 12019.09.30::11:47:11]with Supplier cameo ID: 19_0_2_c04024e_1569869226422_62927_42571 and EA ID: 269C77A3-0185-45c3-84 [2019.09.30::11:47:11] Creating relationship of type: Generalization and id: 26641DEB-D38A-49bd-8E64-34028F92655C with parent.	3D-7B3F63A7F51C
wValueProperty» e: Electrical Energy		[2019.09.30::11:47:11]with Client cameo ID: 19 0.2 c04024e 1569869227031_735730_42604 and EA ID: FECD2281-D903-43a6-A3	20-345862B15522
Debug Project Browser	4	a [2019.09.30::11:47:11]with Supplier cameo ID: _19_0_2_c04024e_1569869227096_95159_42608 and EA ID: 007FFD9C-4BC3-4e32-B	9FC-4F0B60BACFC7 v
SysML	- + CAP NUM SCRL CLOUD	Ready Q Q	100% ~ @ @ 0
Зузис	- + DAP NOM SCRE CLOOD .:	Reduy v.	100 % ~ ~

### What's Next?

- Support for additional Metamodels (i.e., UAF, DoDAF, UML)
- Continued adoption and testing through wider userbase
- Joining the OpenMBEE community



### **List of Sources**

[1] - Figure 1. Relationship between SysML and UML, Object Management Group (OMG). <u>https://www.omgsysml.org/what-is-sysml.htm</u>

[2] - Figure 2. SysML Diagram Types, Object Management Group (OMG). <u>https://www.omgsysml.org/what-is-sysml.htm</u>