SERC DOCTORAL STUDENT FORUM 2023 | NOVEMBER 14, 2023

A Graph Centered Approach to the Verification and Reuse of Systems

Daniel Dunbar

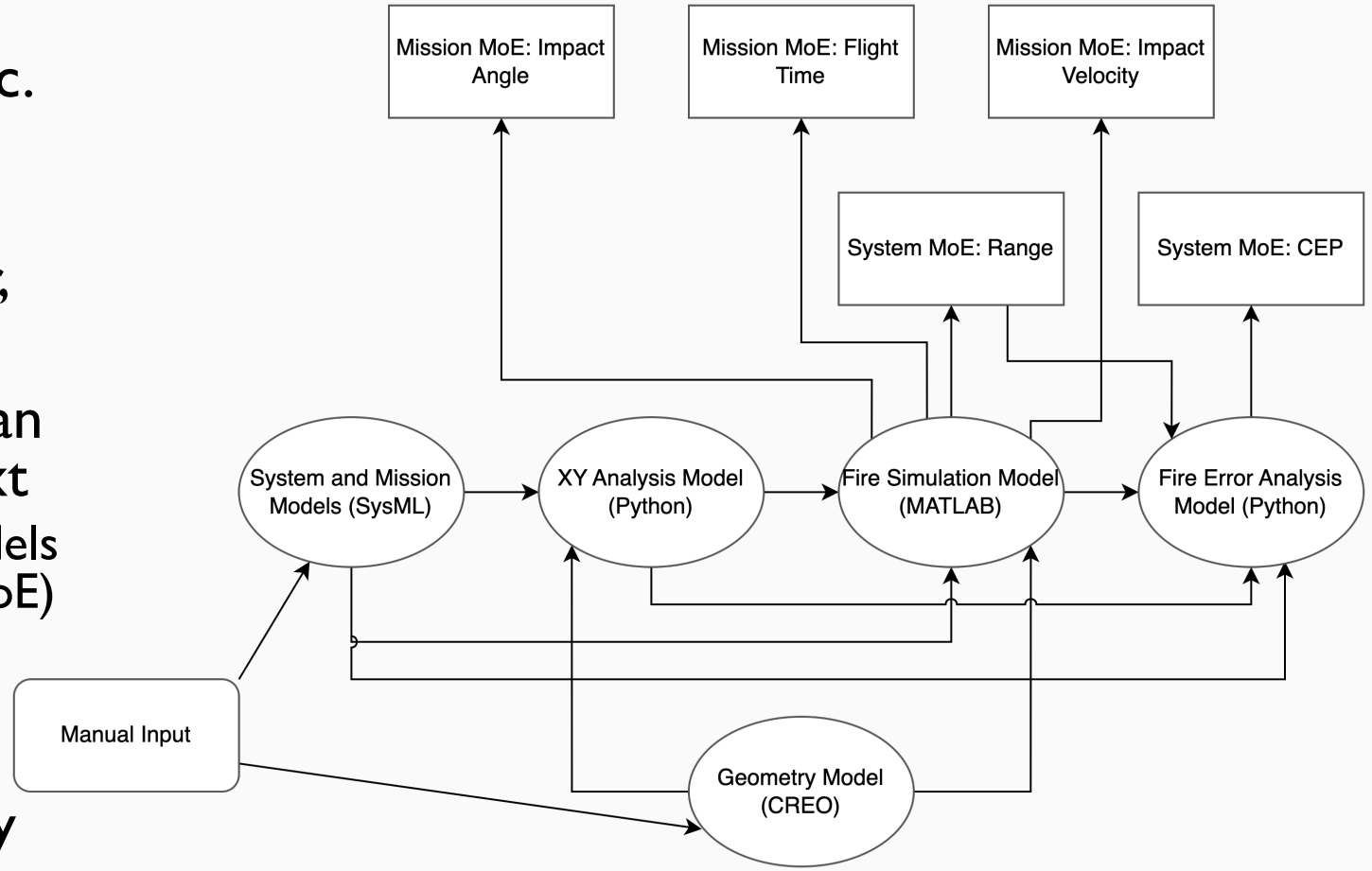
STEVENS
INSTITUTE OF TECHNOLOGY
1870

Digital Engineering

Connecting Model Based Engineering (MBE) to Model Based Systems Engineering (MBSE) to create integrated understanding

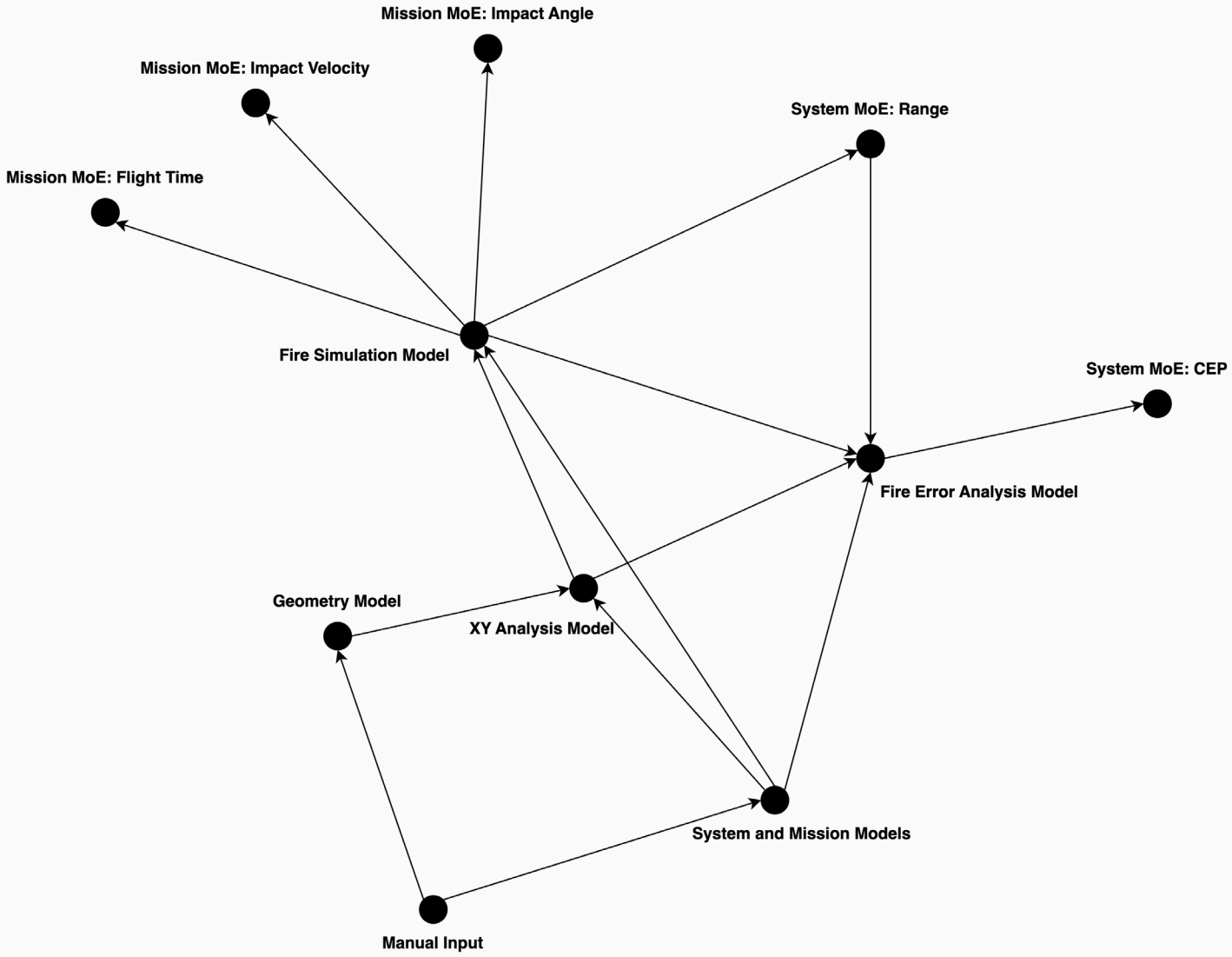
The Problem – Digital Thread and Higher Order Analysis

- Analyses often involve multiple models, simulations, domains, etc.
- Multiple models can interact in different ways (share common parameters, inform one another, etc.)
- Different levels of abstraction can occur in a broad analysis context
  - Domain, System, and Mission models and Measures of Effectiveness (MoE) can be used in the same analysis context
- As an analysis increases in complication, the ability to verify analysis structure increases in difficulty



Digital Threads are Graphs

- Digital thread as simple, directed graph
  - Edges represent sequencing information
- Graph-Based Analysis can provide insight:
  - Cycles in the directed graph
  - Order for automation
  - Complexity
- Simple graph analysis only looks at simple edges



Ontology-Aligned Data

"An ontology is a directed labeled graph that is identified by a unique IRI (Internationalized Resource Identifier) and that describes a set of things by a set of propositional statements that are regarded to be true in some context." (Wagner et. al)

- Multiple edge types enables representation of different relationships
- Has language and syntax to be ingested and interpreted by a computer
- Axioms for giving robust representation of a knowledge base
- Can conform to Description Logic, which enables entailments using formal mathematical logic

Wagner, D. A., Chodas, M., Elaasar, M., Jenkins, J. S., & Rouquette, N. (2022). Ontological Metamodeling and Analysis Using openCAESAR. In A. M. Madni, N. Augustine, & M. Sievers (Eds.), *Handbook of Model-Based Systems Engineering* (pp. 1–30). Springer International Publishing. https://doi.org/10.1007/978-3-030-27486-3_78-1

Digital Engineering Framework for Integration and Interoperability

Digital Engineering Framework for Integration and Interoperability (DEFII)[1]

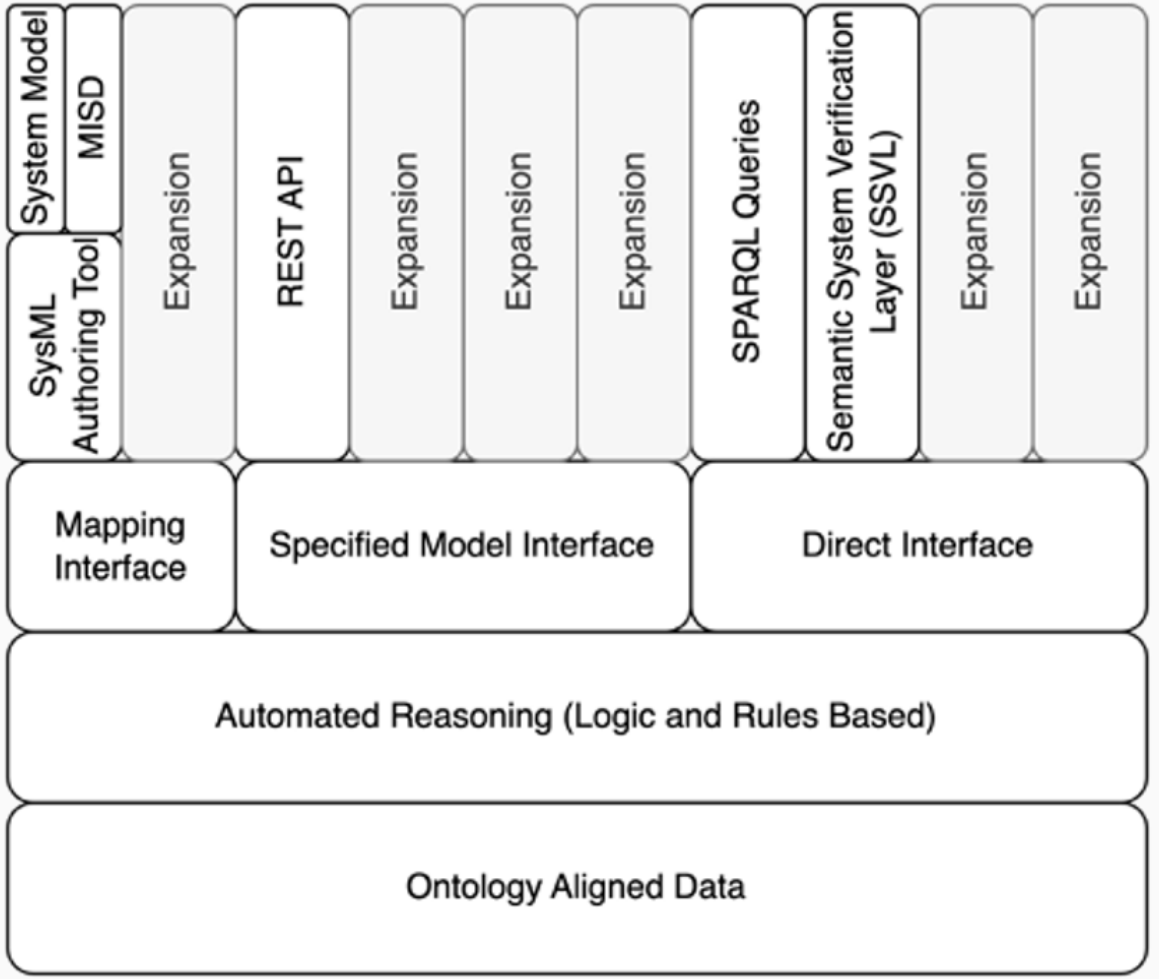
- Three different types of interfaces
  - Mapping Interface
  - Specified Model Interface
  - Direct Interface

Model Interface Specification Diagram (MISD)[1]

- Packages system elements in a useful way for analysis
- Can be aggregated to form a System of Analysis (SoA)

Semantic System Verification Layer (SSVL)[2]

- Uses the Direct Interface to perform verification and/or validation operations on the ontology-aligned data



[1] Dunbar, D., Hagedorn, T., Blackburn, M., Dzielski, J., Hespelt, S., Kruse, B., Verma, D., & Yu, Z. (2023). Driving digital engineering integration and interoperability through semantic integration of models with ontologies. Systems Engineering, sys.21662. https://doi.org/10.1002/sys.21662

[2] Dunbar, D., Hagedorn, T., Blackburn, M., & Verma, D. (2022). Use of Semantic Web Technologies to Enable System Level Verification in Multi-Disciplinary Models. In B. R. Moser, P. Koomsap, & J. Stjepandić (Eds.), Advances in Transdisciplinary Engineering. IOS Press. https://doi.org/10.3233/ATDE220632

Shift Portions of Verification and Validation (V&V) to the Left

- Model Verification asks, "Is the model trustworthy?"[1]
- Five Approaches Identified for Model-Based V&V[1]
  - Model Appraisal
  - Guided Modeling
  - Simulation
  - Formal Proof
  - Digital Twin and Digital Thread

[1]Madni, A. M., & Sievers, M. (2018). Model-based systems engineering: Motivation, current status, and research opportunities. *Systems Engineering, 21*(3), 172–190. https://doi.org/10.1002/sys.21438

Model Verification

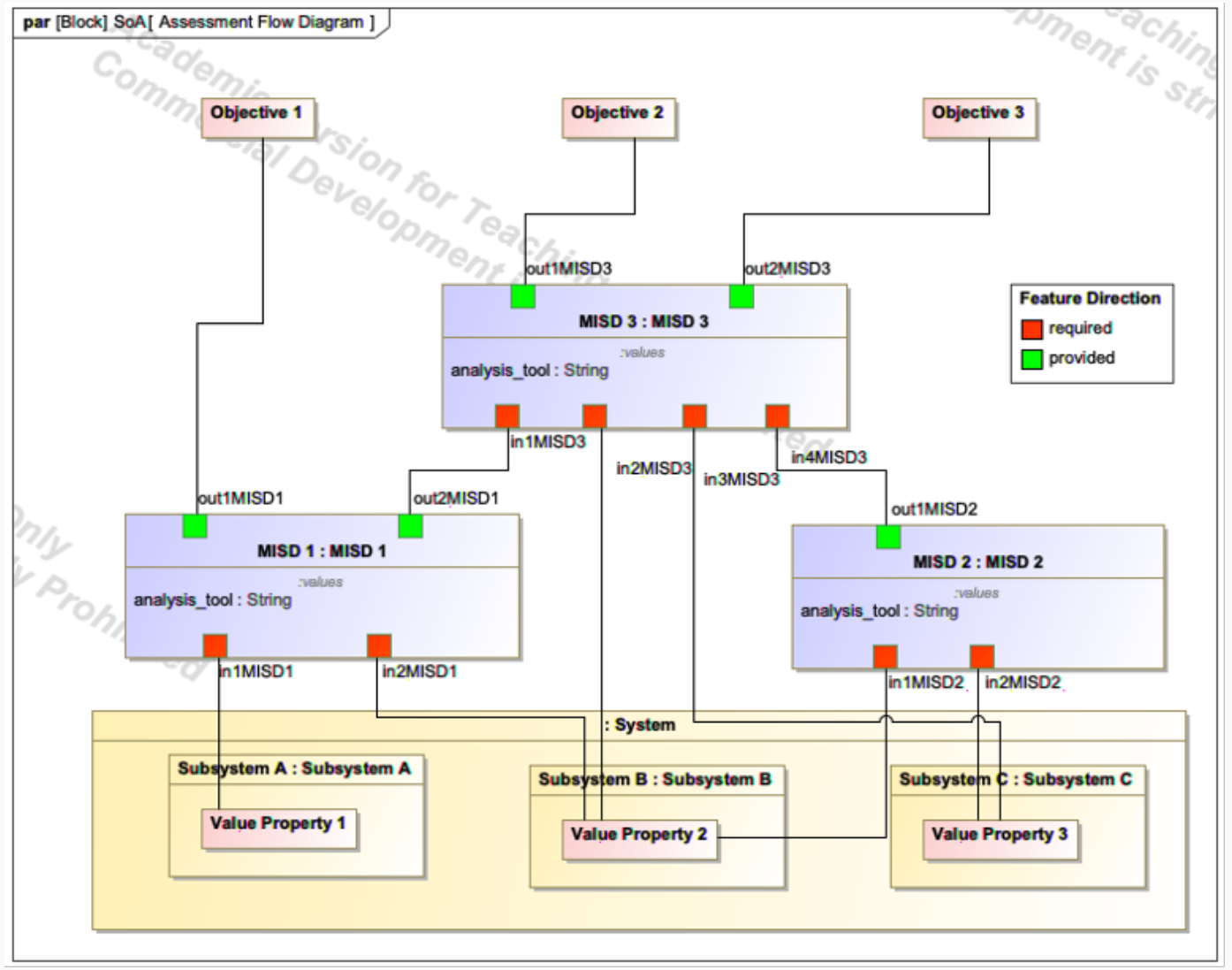
Three-pronged approach to evaluate trustworthiness of model

Abstract System of Analysis (SoA)

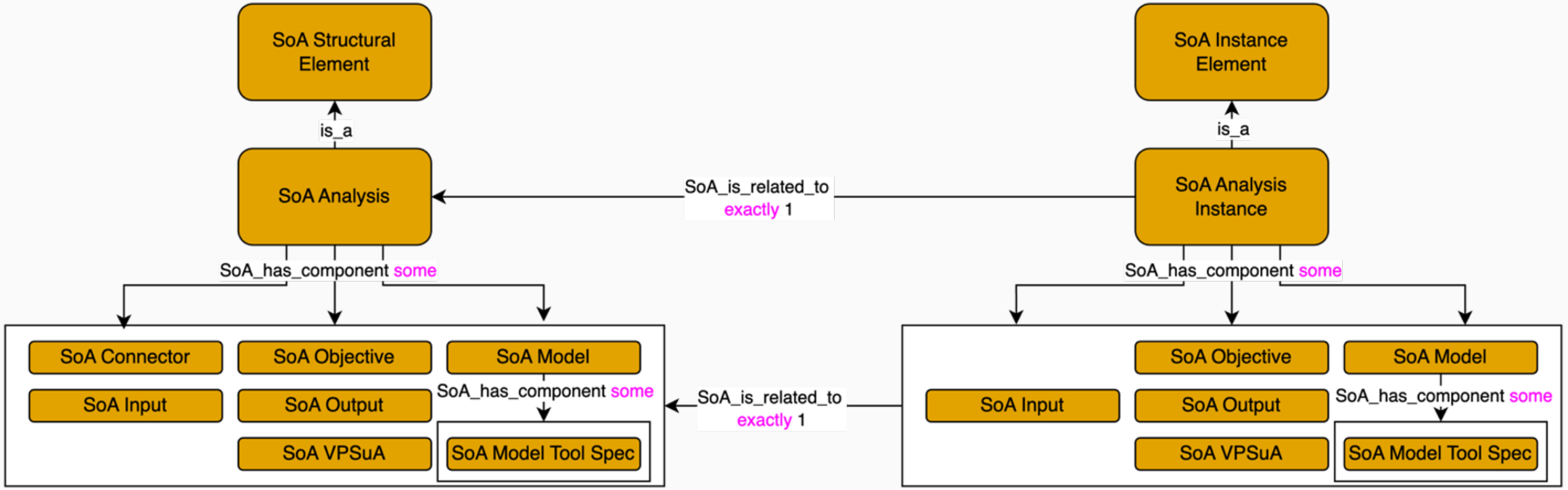
Value Properties identified as Objectives of the SoA

Intermediate Analyses defined through model interface specifications within SysML

Value Properties of System Under Analysis



System of Analysis Ontology Development



Three-Pronged Approach to Model Verification

DL Reasoning

- Open World

Examples

- Irreflexive Property Violation
- Maximum Cardinality

SHACL Constraints

- Closed World

Examples

- Existence
- Minimum Cardinality

Graph Analysis

- Separate from SWT

Examples

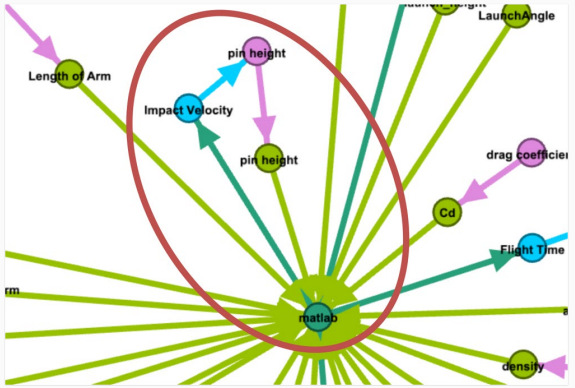
- Confirmation Graph is a DAG
- Confirmation Graph is Weakly Connected

SSVL Verification Applied to SoA Well-Formedness

Scenario Name	Scenario Description	DL-Reasoning	SHACL	GRAPH	Pellet Message
C10	Gravity to Air Temp Inputs on Fire Simulation Model	1	0	0	1) Functional SoA_terminated_to_target 09de70c2-0dfd-4599-8ebe-1dfaab9b7d61_SoA SoA_terminated_to_target 69ec40fc-3edd-4635-8e56-f21431071a86_SoA 09de70c2-0dfd-4599-8ebe-1dfaab9b7d61_SoA SoA_terminated_to_target a4f578a8-ce36-4417-ad15-81f982ebf855_SoA

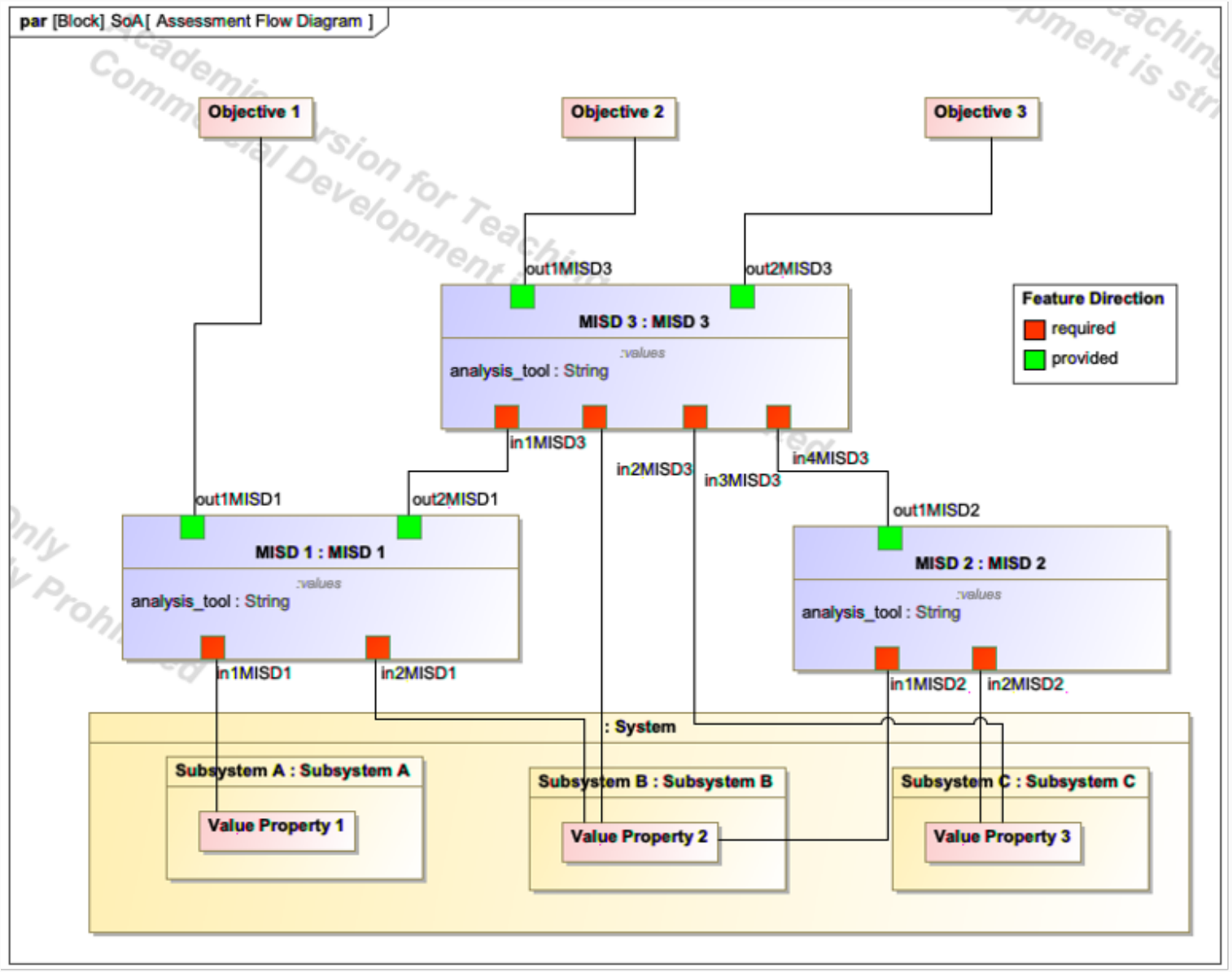
Scenario Name	Scenario Description	DL-Reasoning	SHACL	GRAPH	SHACL Messages
C16	Removed CircularErrorProbability (CEP) from objective	0	1	0	Violation: Axiom 7 - Value Property: 'CEP' is not tagged with a loaded ontology term

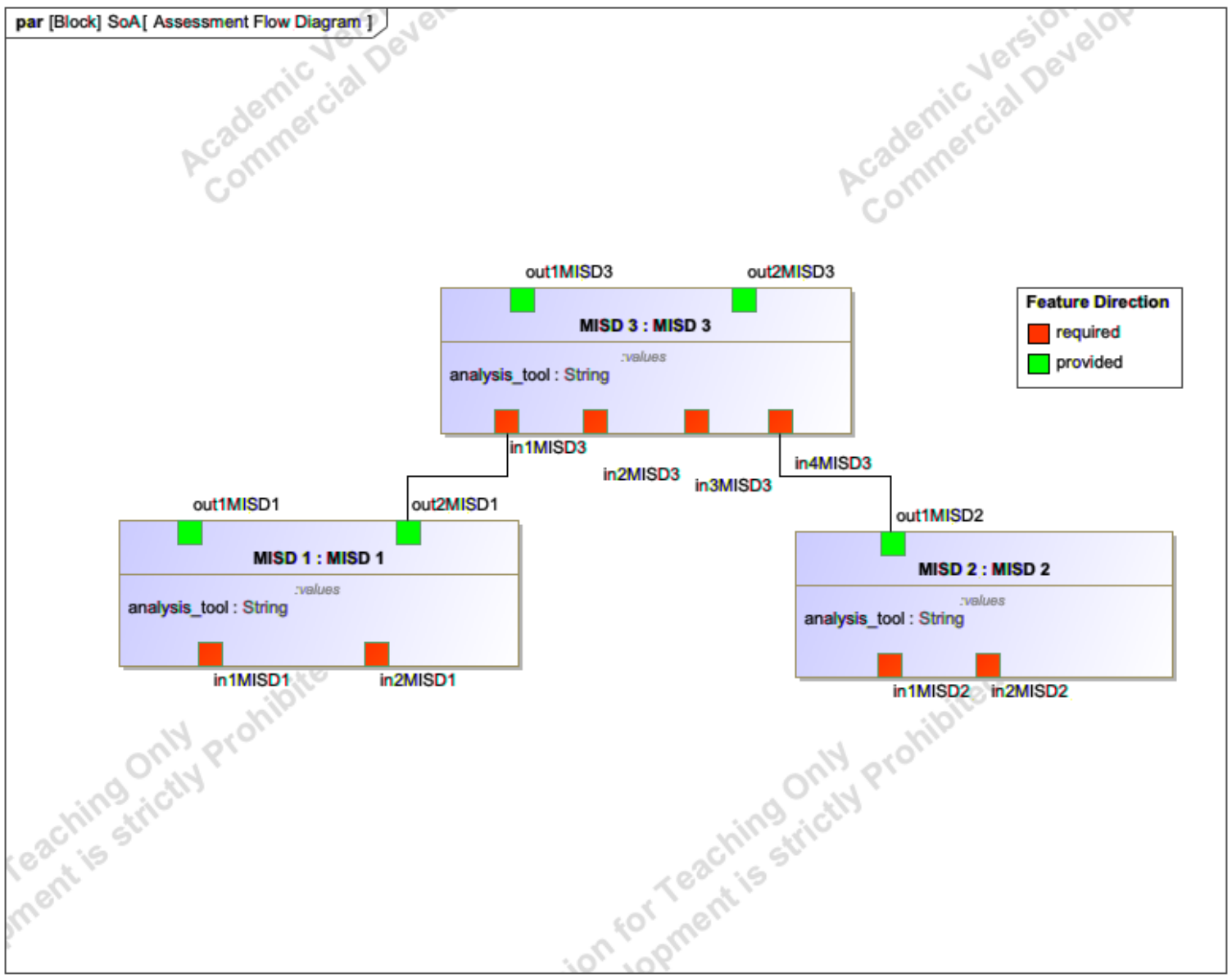
Scenario Name	Scenario Description	DL-Reasoning	SHACL	GRAPH
C26	Cycle from Impact Velocity to Pin Height	0	0	1

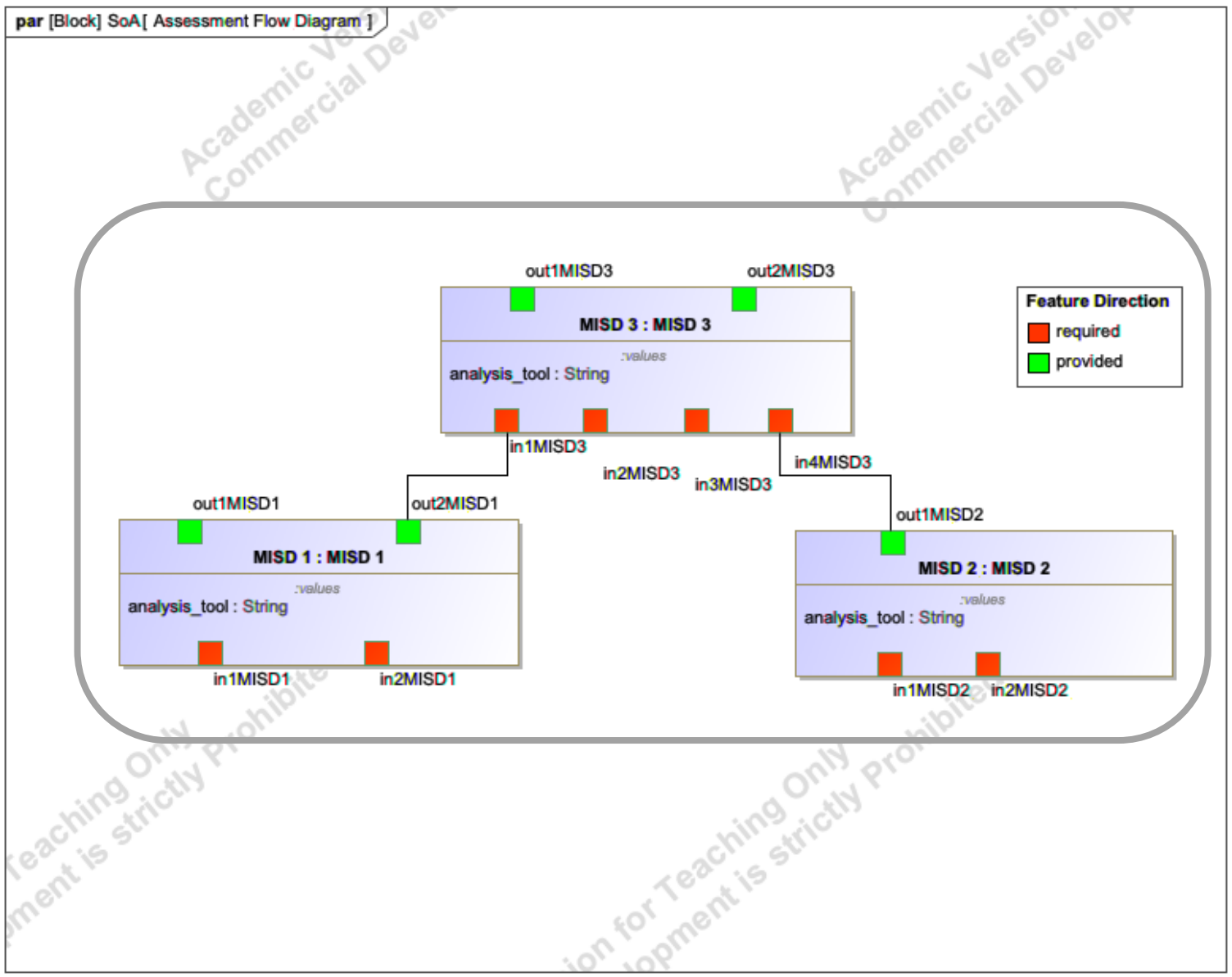


Model Reuse (Guided Modeling)

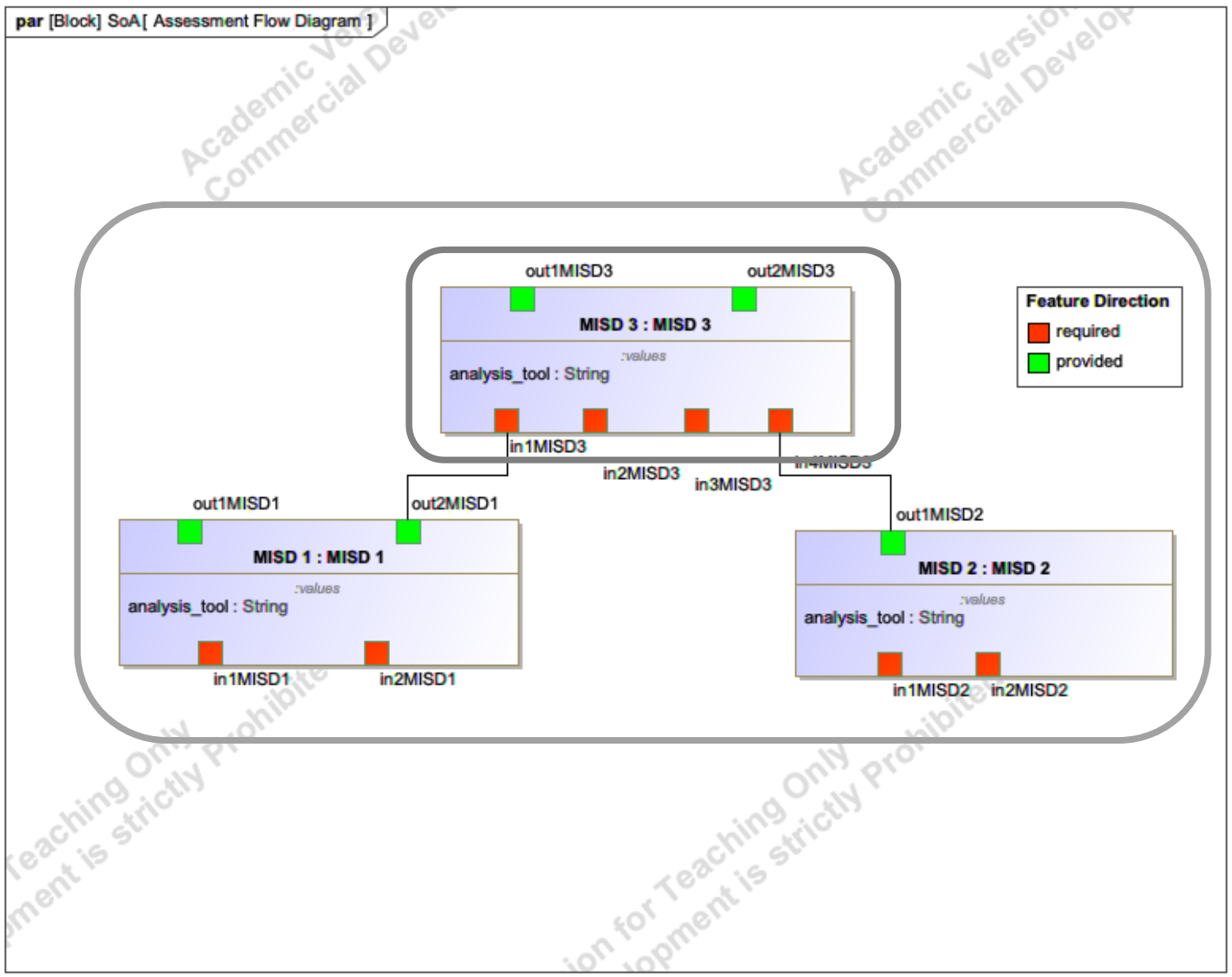
Use of component library to enable capture and reuse of previous models and analyses


par [Block] SoA[Assessment Flow Diagram]
Objective 1
Objective 2
Objective 3
out1MISD3
out2MISD3
MISD 3 : MISD 3
:values
analysis_tool : String
Feature Direction
required
provided
in1MISD3
in2MISD3
in3MISD3
in4MISD3
out1MISD1
out2MISD1
out1MISD2
MISD 1 : MISD 1
:values
analysis_tool : String
MISD 2 : MISD 2
:values
analysis_tool : String
in1MISD1
in2MISD1
in1MISD2
in2MISD2
: System
Subsystem A : Subsystem A
Value Property 1
Subsystem B : Subsystem B
Value Property 2
Subsystem C : Subsystem C
Value Property 3

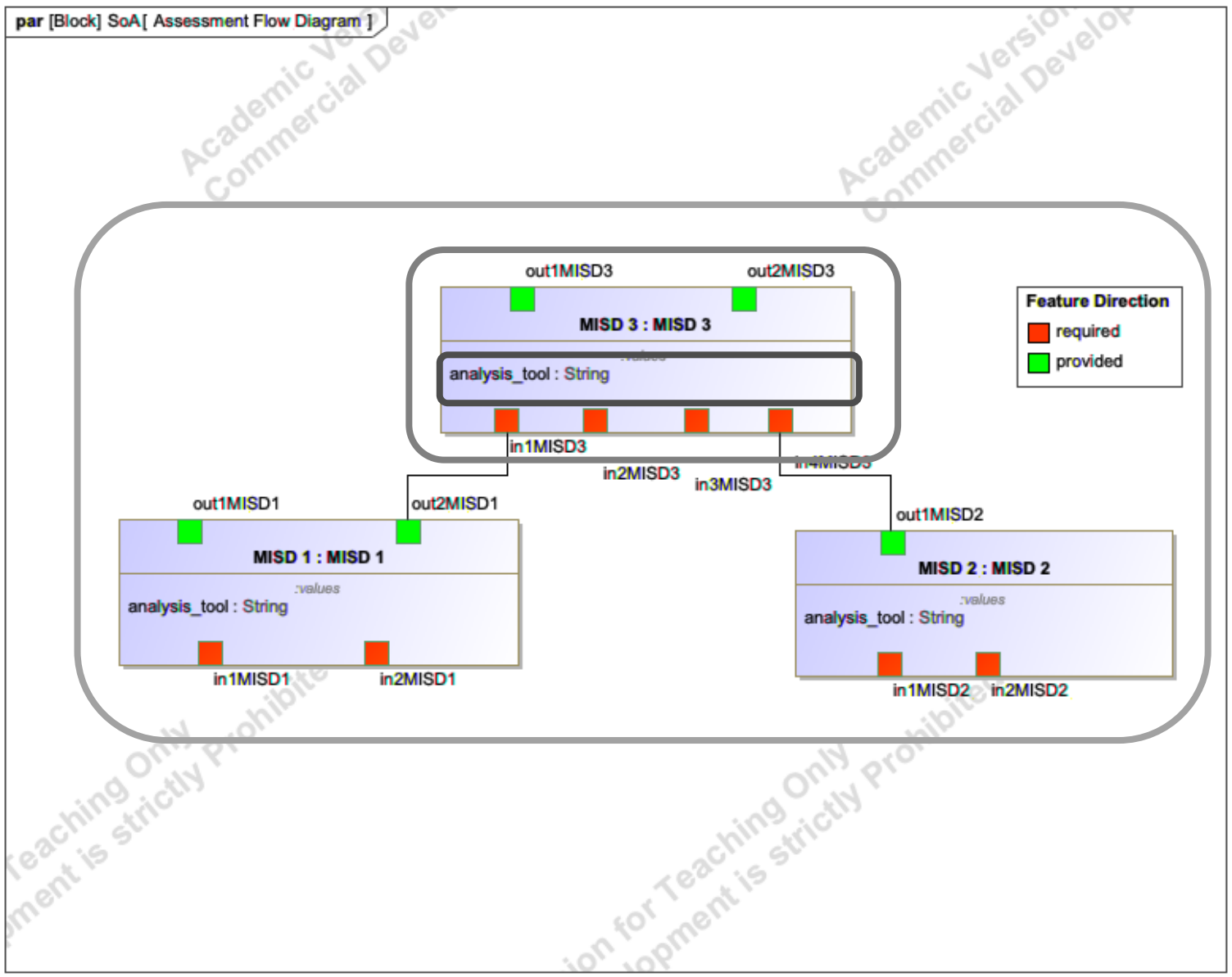

par [Block] SoA [Assessment Flow Diagram]
out1MISD3
out2MISD3
MISD 3 : MISD 3
:values
analysis_tool : String
in1MISD3
in2MISD3
in3MISD3
in4MISD3
Feature Direction
required
provided
out1MISD1
out2MISD1
MISD 1 : MISD 1
:values
analysis_tool : String
in1MISD1
in2MISD1
out1MISD2
MISD 2 : MISD 2
:values
analysis_tool : String
in1MISD2
in2MISD2


par [Block] SoA [Assessment Flow Diagram]
out1MISD3
out2MISD3
MISD 3 : MISD 3
:values
analysis_tool : String
in1MISD3
in2MISD3
in3MISD3
in4MISD3
Feature Direction
required
provided
out1MISD1
out2MISD1
MISD 1 : MISD 1
:values
analysis_tool : String
in1MISD1
in2MISD1
out1MISD2
MISD 2 : MISD 2
:values
analysis_tool : String
in1MISD2
in2MISD2

- Full SoA



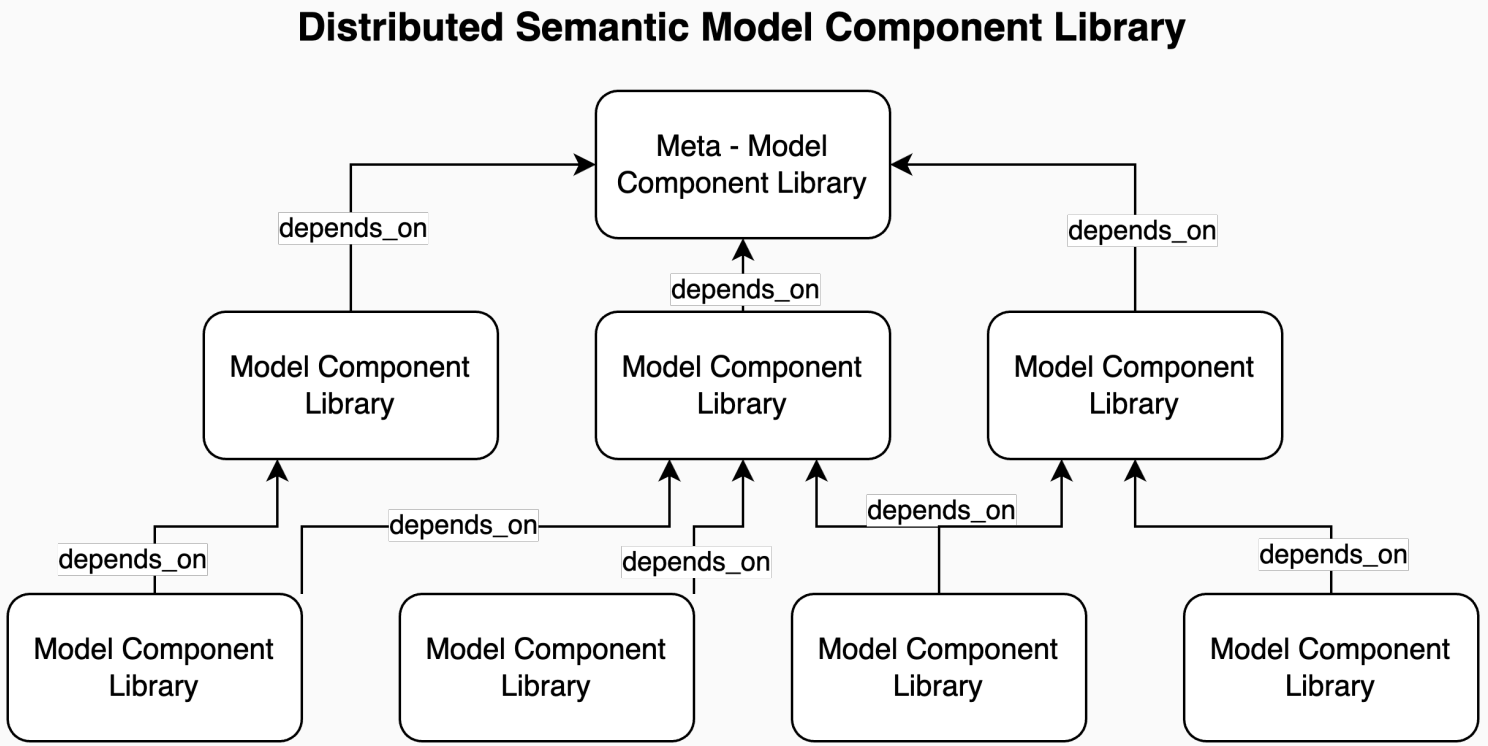
- Full SoA
- Individual Model



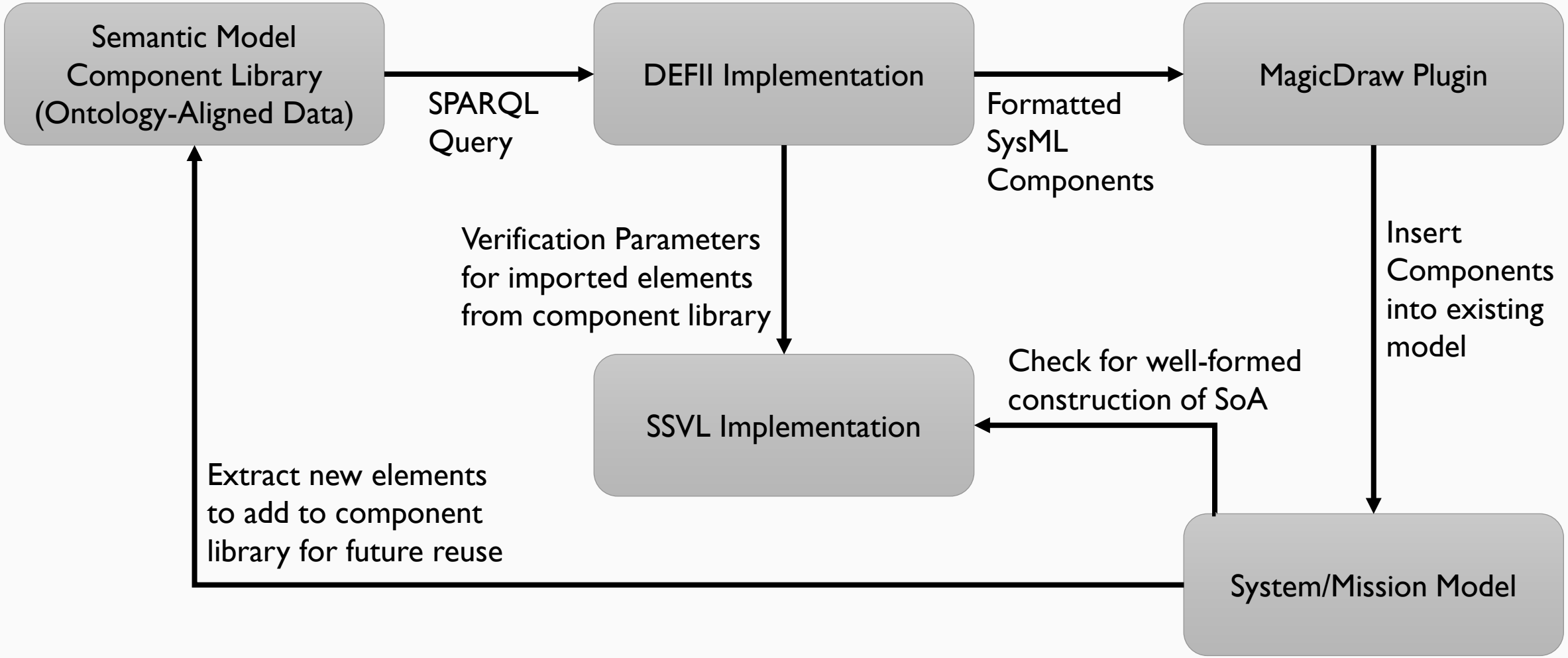
- Full SoA
- Individual Model
- Model Metadata

Semantic Model Component Library

- Can be distributed
- Extensible
- Tool Agnostic
- Can be versioned via git (text based)



Component Library Usage Flow in DEFII



Conclusion

Ontology-aligned data provides flexible access that promotes interoperability and extension opportunities as definitions become richer.

Ontology-Aligned Data as Foundation

- Flexible Access to Data
- Interoperability between tools
- Extensible across all interfaces

Model Verification via deductive reasoning

- DL-Reasoning
- Constraint Based Closed World Reasoning
- Graph Analysis

Model Reuse through Guided Modeling

- Component Library in searchable format
- Extensible and Distributed
- Connected to model verification approach

Thank you

Stay connected with SERC Online:

Email the presenter: Daniel Dunbar

ddunbar1@stevens.edu

Email the advisor: Dr. Mark Blackburn

mblackbu@stevens.edu