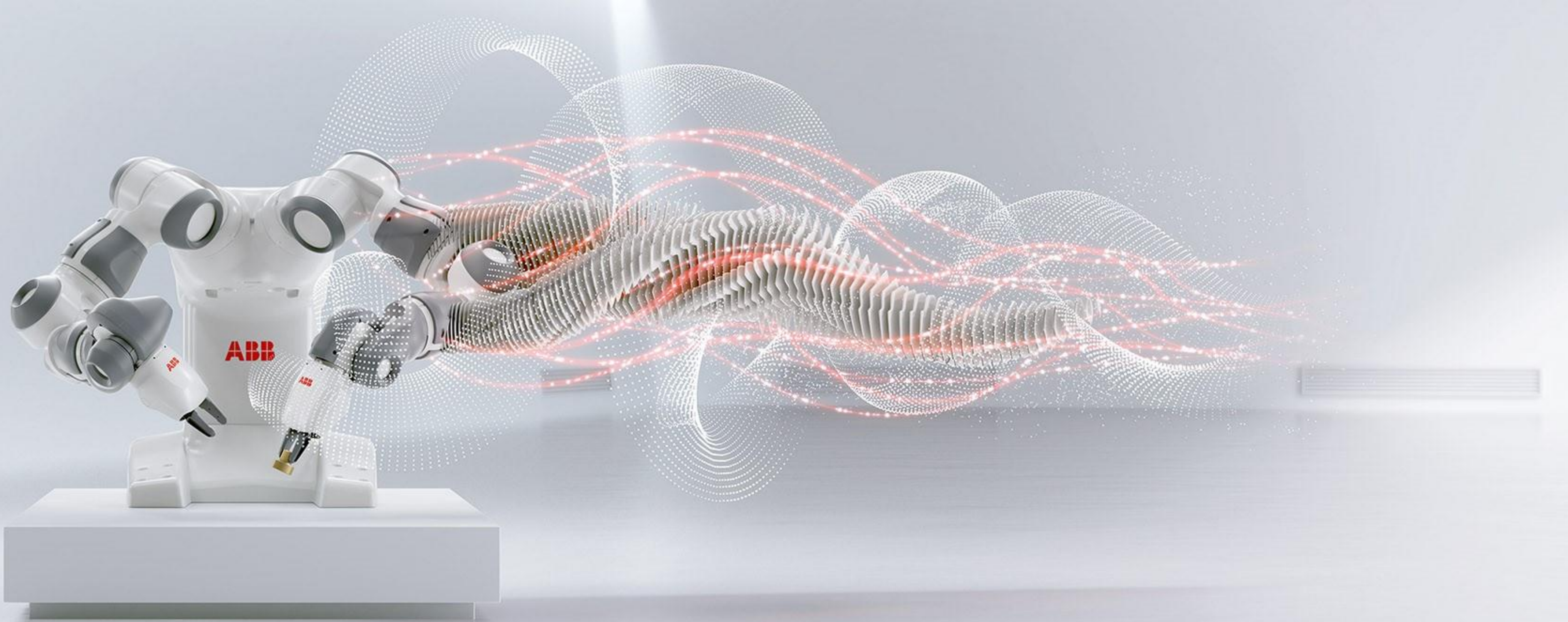


5TH AUTOMATIONML PLUGFEST

Semantic and Pragmatic Interoperability via AutomationML

Dr. Prerna Bihani*, ABB Corporate Research Center Germany
Prof. Dr.-Ing. Rainer Drath, Pforzheim University of Applied Sciences

Document ID.:

Rev.:

*The work has been partially supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) as part of the INTEGRATE Project.

Outline

- The Interoperability Requirement
- Semantic and Pragmatic Interoperability
- Solution approach using Reference Class Library
- Middleware for Reference Class Library

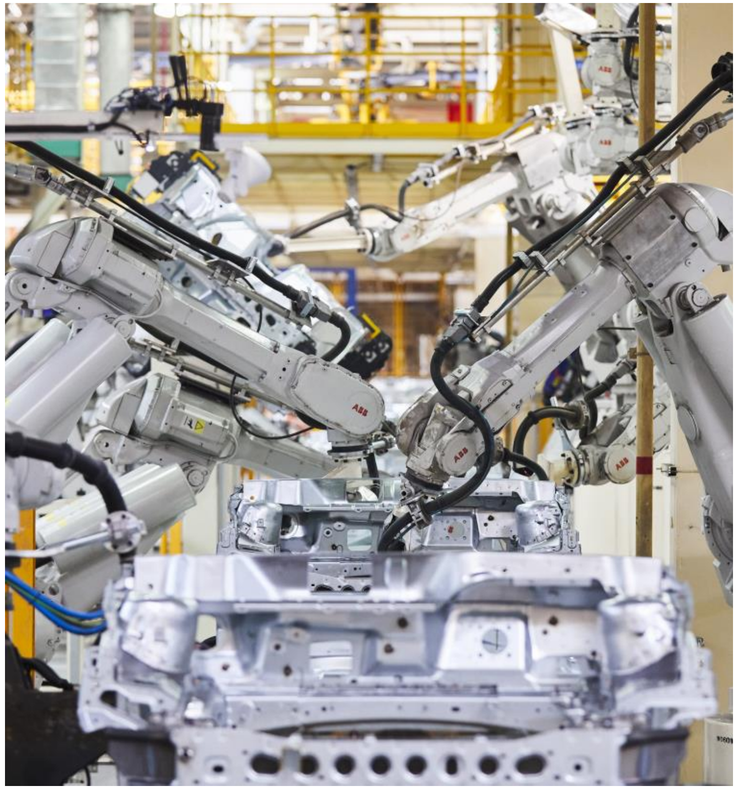
Engineering Tools

Interoperability Requirement

Background

Typical industrial product involves multiple activities during its engineering such as bidding, design, configuration, simulation, testing and commissioning.

- Collaboration requires not only verbal communication but also interoperability of heterogeneous tools
- Beyond syntax: alignment on data meaning and utility are required for meaningful interoperability



Engineering Tools

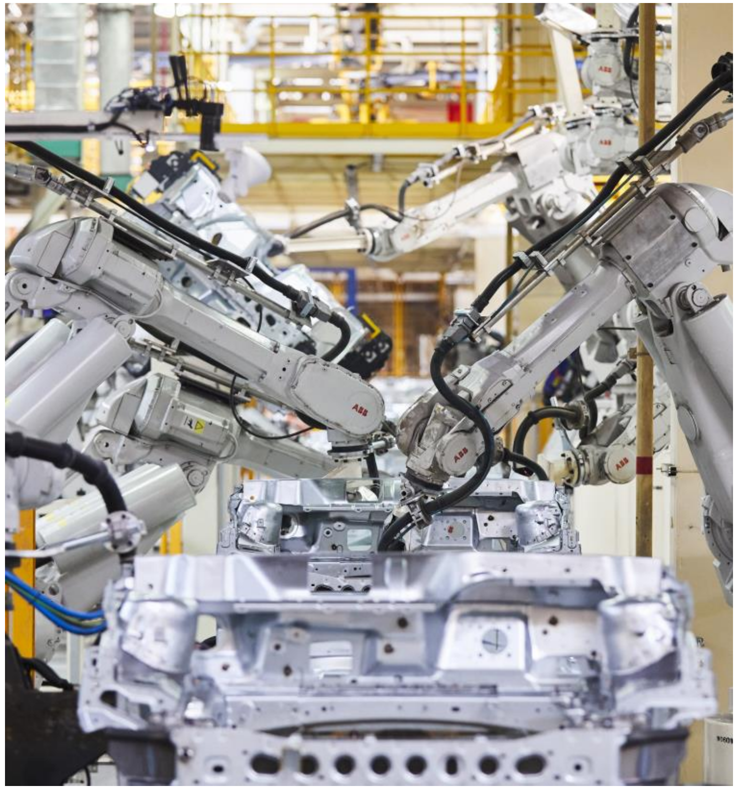
Interoperability Requirement

Motivation

Heterogeneity is a fact of life and ever increasing…

GOAL: Achieve meaningful interoperability

- Without assuming predefined semantic standard or transformation language
- Without introducing changes in the original tool data models



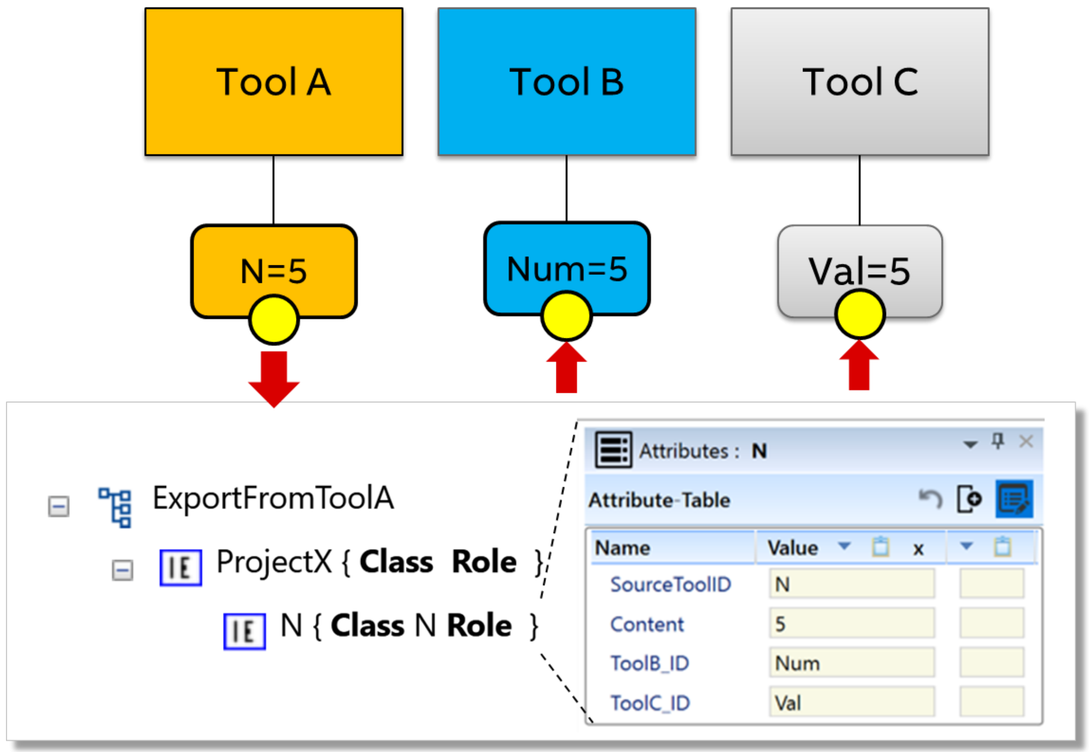
Simple Semantic Alignments

What we already do and why

Comparison of different mapping approaches

	Standardizing data model of the tools	Mapping in exporters or importers	Mapping in the intermediate data model
Tools remain unchanged		✓	✓
Tools innovate independently		✓	✓
Generic exporters and importers	✓		✓
Flexible and extensible mappings			✓

1:1 Semantic Mappings in AutomationML-based Model

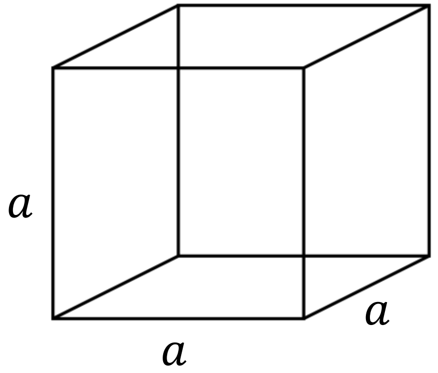


Moving towards Complex Semantic Alignments

Alignment on the meaning of data

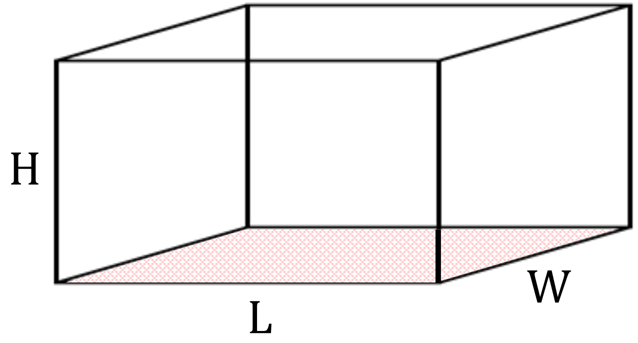
Tool A:

- a is the length parameter of a cube
- $SA = 6a^2$
- $V = a^3$



Tool B:

- L, W, H are the length, width, and height parameters respectively of a rectangular solid



Examples of Semantic Correspondences between Tools A and B:

Length: $a \sim L$

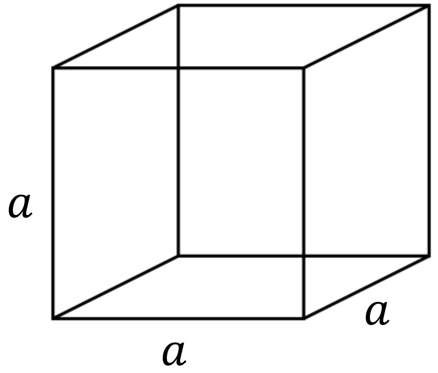
Base Area: $a^2 \sim L * W$

Alignment on the meaning of data

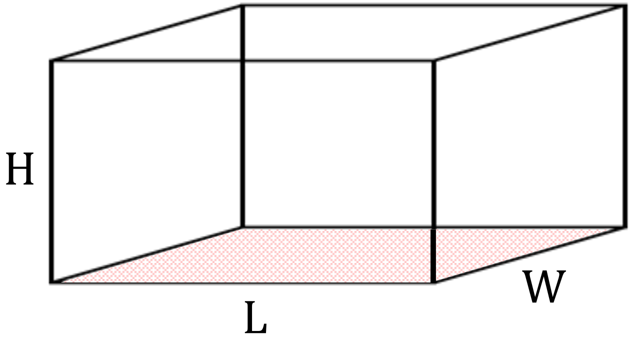
Pragmatic Interoperability

Alignment on the utility of data

Tool A:
- a is the length parameter of a cube
- $SA = 6a^2$
- $V = a^3$



Tool B:
- L, W, H are the length, width, and height parameters respectively of a rectangular solid



Examples of Pragmatic alignments for Tool B → Tool A data transfer:

Volume preservation: $$a = \left(\sqrt[3]{BA * H}\right)$$

OR

Surface Area preservation: $$a = \sqrt{\frac{BA+H*W+L*H}{3}}$$

Alignment on the utility of data

*Note that $\sqrt[3]{L * W * H} \neq \frac{L*W+H*W+L*H}{3}$ for L=1, W=3, H=9

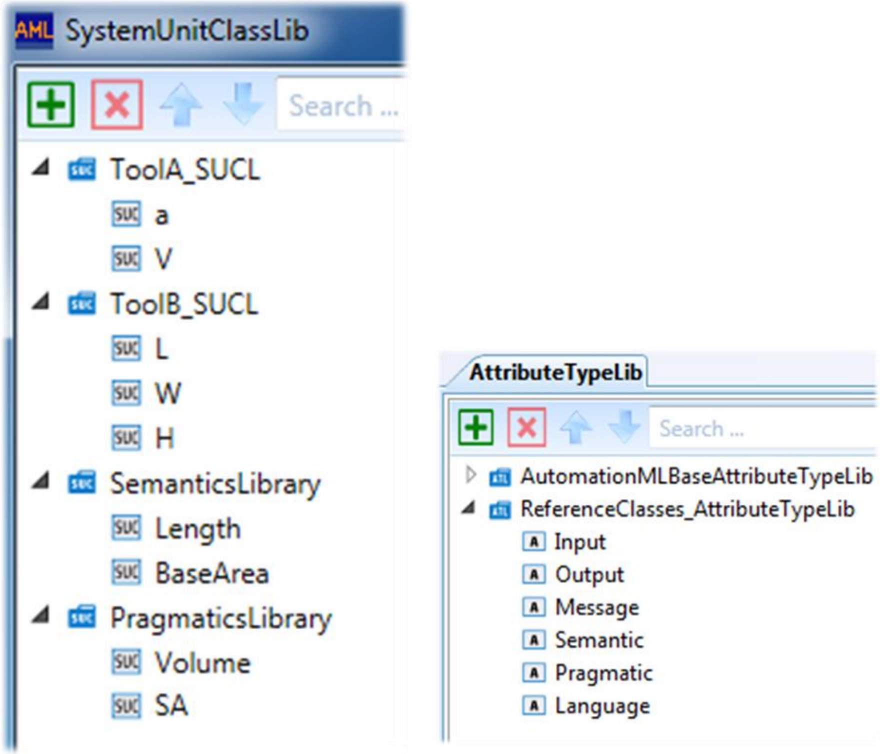
Introducing the Reference Class Library

Encoding Semantic and Pragmatic Interoperability Alignments outside participating tools

The **Reference Class Library** is an evolving interoperability knowledge base implemented in AutomationML

- As a collection of System Unit Class Libraries and custom Attribute Types
- Containing type information for individual tool data models in original tool semantics (auto-generated)
- With a Semantics Library for storing semantic alignments between the tools
- With Pragmatics Library for storing pragmatic alignments between the tools

References to types use their paths as identification, e.g. element type *a* in Tool A is identified by *ToolA_SUCL/a*

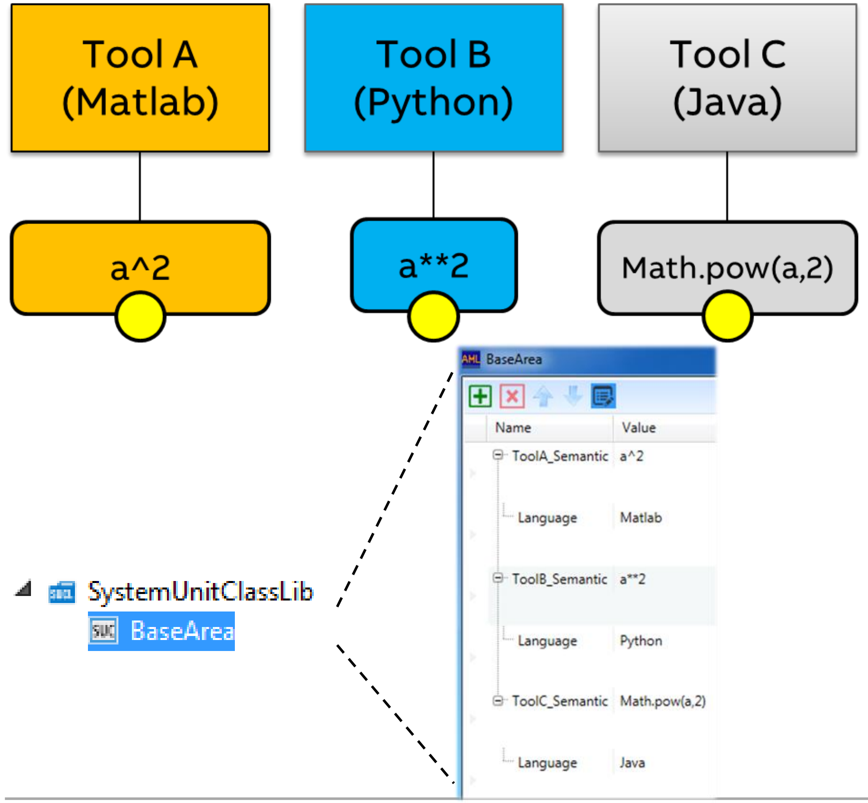


Example of a Reference Class Library

Dealing with heterogeneous tool languages

Transformations in Semantic and Pragmatic Interoperability

Mapping Heterogeneous Tool Languages



Encoding processing language within the intermediate model allows flexibility and transparency, while offering the same benefits as before:

→ Tools remain unchanged

→ Tools innovate independently

→ Generic exporters and importers

→ Flexible and extensible mappings

Example implementation of Reference Class Library

For Tools A and B

Simple Semantic Class

Length correspondence

Length

Name	Value	Reference
ToolA_Semantic	ToolA_SUCL/a	ReferenceClasses_AttributeTypeLib/Semantic
ToolB_Semantic	ToolB_SUCL/L	ReferenceClasses_AttributeTypeLib/Semantic

Complex Semantic Classes

Base Area correspondence

BaseArea

Name	Value	Reference
ToolA_Semantic	<math xmlns='http://www.w3.org/1998/Math/MathML'><msup><mi>ToolA_SUCL/a</mi><mn>2</mn></msup></math>	ReferenceClasses_AttributeTypeLib/Semantic
Language	MathML	ReferenceClasses_AttributeTypeLib/Language
ToolB_Semantic	ToolB_SUCL/L*ToolB_SUCL/W	ReferenceClasses_AttributeTypeLib/Semantic
Language	Matlab	ReferenceClasses_AttributeTypeLib/Language

Pragmatic Class

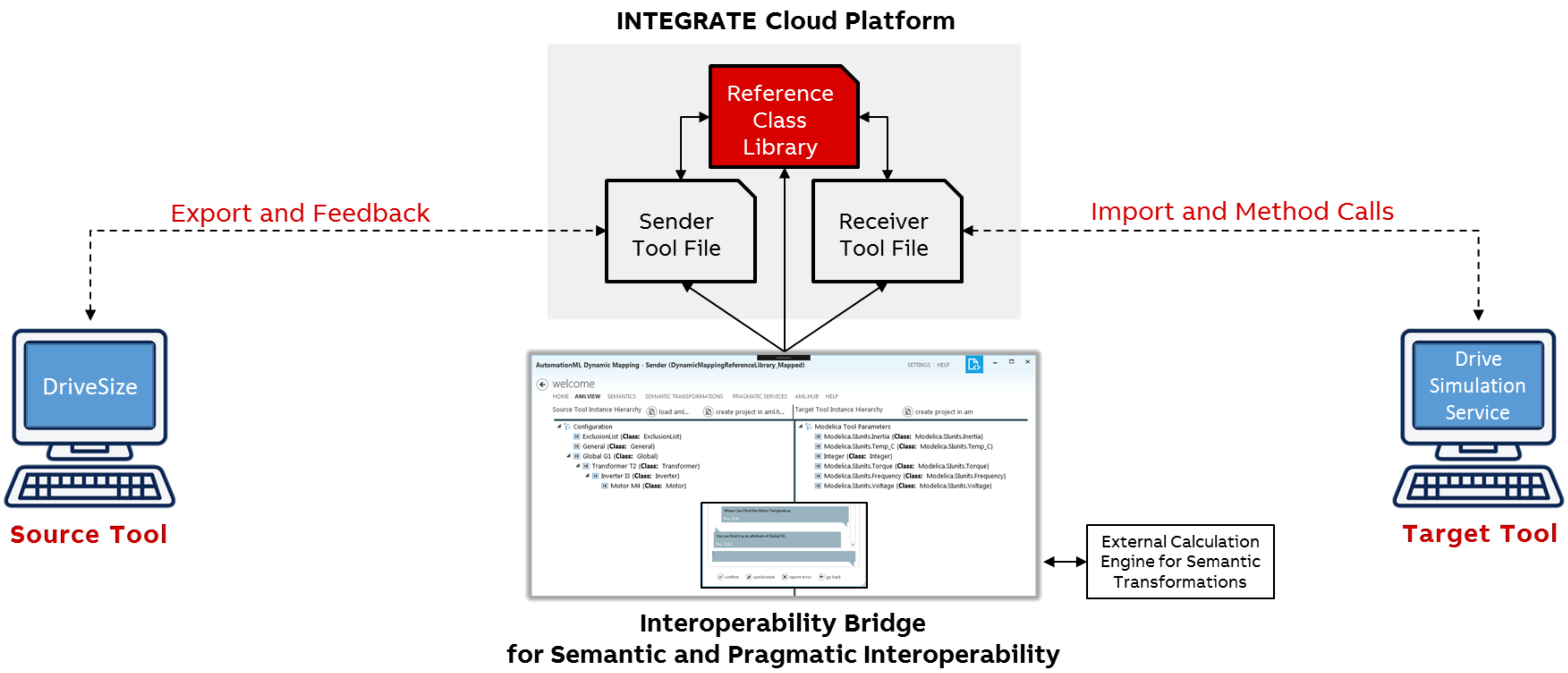
Volume preservation for Tool B → Tool A

Volume

Name	Value	Reference
ToolA_Pragmatic	ToolA_SUCL/V=(ToolA_SUCL/a)^3	ReferenceClasses_AttributeTypeLib/Pragmatic
ToolB_Input	ToolA_SUCL/a=(SemanticsLibrary/BaseArea*ToolB_SUCL/H)^(1/3)	ReferenceClasses_AttributeTypeLib/Input
Language	Matlab	ReferenceClasses_AttributeTypeLib/Language
Language	Matlab	ReferenceClasses_AttributeTypeLib/Language

Application to Feedback-based Dimensioning of Industrial Drives*

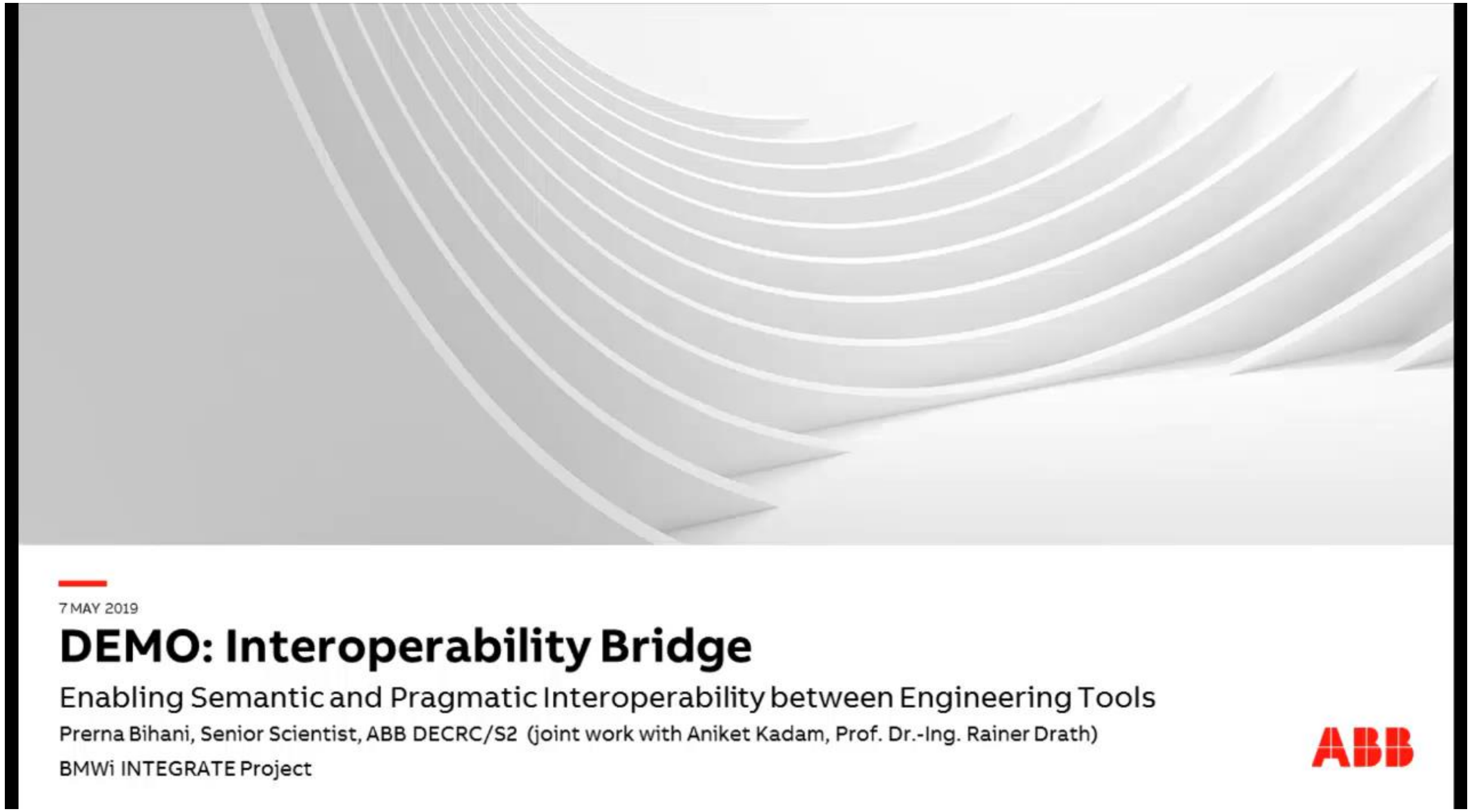
Overview of Architecture with Middleware



 *Joint work with Aniket Kadam as part of the INTEGRATE Project

Application to Feedback-based Dimensioning of Industrial Drives*

Video Demo



*Joint work with Aniket Kadam as part of the INTEGRATE Project

Benefits of Our Method

- No predefined semantic standard or transformation language required
- Interactive data discovery and creation of interoperability mappings
- Complex semantic mappings and flexible, external processing
- Enabling of Pragmatic Interoperability
- Simple architecture
- Collaborative Engineering “on the fly”

ABB