# Concept Learning in Engineering based on Refinement Operator

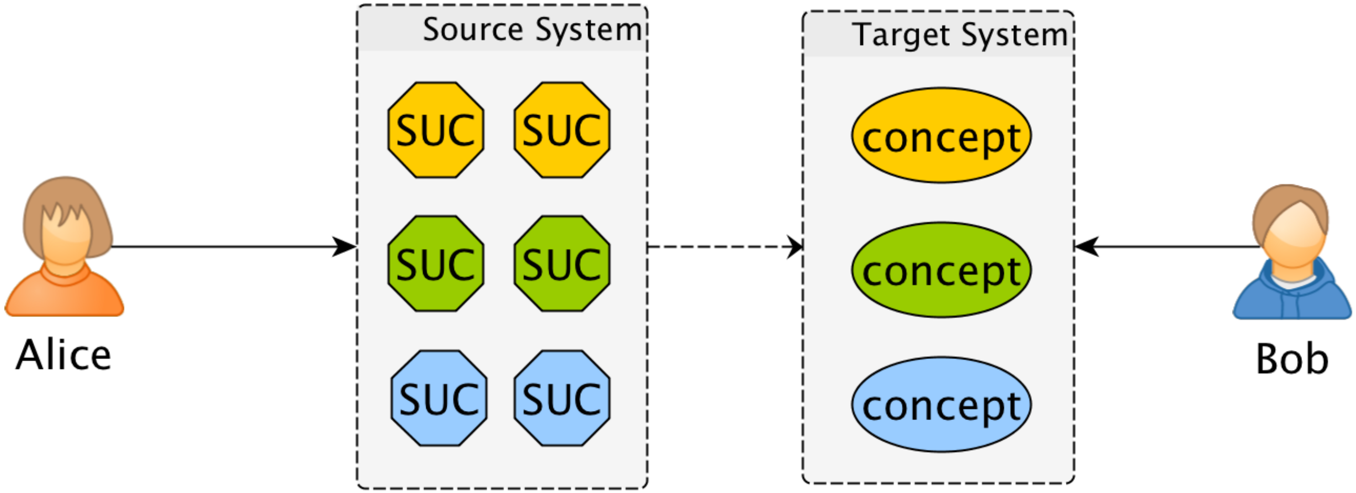
## 28th International Conference on Inductive Logic Programming


<u>Yingbing Hua</u>, Björn Hein

Institute for Anthropomatics and Robotics – Intelligent Process Control and Robotics (IAR-IPR)

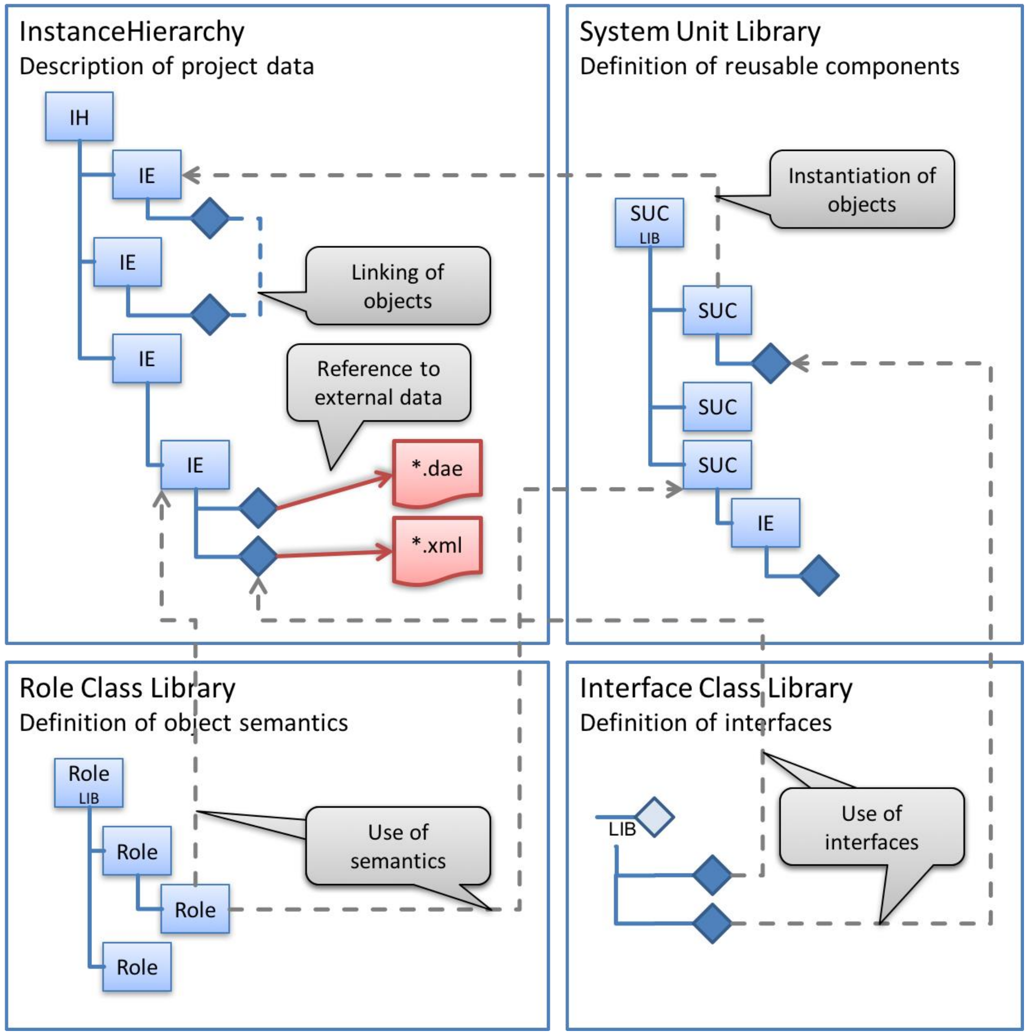

# Motivation

- Semantic Interoperability between engineering systems
- Machine interpretation of user defined concepts
- *“What does one <u>target concept</u> mean using the language of the source system?“*



# AutomationML (IEC 62714)

```
KR5_01
Kinect_02
```




AutomationML Topology Description Architecture (Schmidt, N. and Lüder, A., 2015)

```
KR5:Robot
Kinect:Sensor
```

```
Actuator
 └─Robot
Sensor
 └─RGBSensor
```

```
IO
 ├─DigitalIO
 └─AnalogIO
```

# Semantic Lifting

| AML models | OWL Models | Example |
|---|---|---|
| role class | class | `Robot` |
| Interface class | class | `DigitalIOInterface` |
| system unit | individual | `KR5` |
| external interface | individual | `digital_io_1` |
| relationship | object property | `hasIE, hasEI` |
| attribute | data property | `hasWeight` |

Add semantic annotation in the OWL models to indicate their roles in CAEX (Runde et. al, 2009)

# Semantic Lifting – Example

attributes

substructures

```xml
<SystemUnitClass Name="KR5">
    <Attribute Name="model" AttributeDataType="xs:string" ChangeMo
        <Description>product model of this hardware</Description>
        <Value>KR5</Value>
    </Attribute>
    <Attribute Name="manufacturer" AttributeDataType="xs:string" C
        <Description>manufacturer of this hardware</Description>
        <Value>KUKA</Value>
    </Attribute>
    <Attribute Name="weight" AttributeDataType="xs:double" Unit="k
        <Description>weight of this hardware</Description>
        <Value>127</Value>
    </Attribute>
    <InternalElement Name="RobotArm" ID="5f5f2c62-5ac5-4ca8-8bc8-d
        <RoleRequirements RefBaseRoleClassPath="ReAppRoleClassLib/
            <Attribute Name="payload" Unit="kg" AttributeDataType=
                <Description>Maximum payload of this robot arm</De
                <Value>37</Value>
            </Attribute>
            <Attribute Name="numAxes" AttributeDataType="xs:unsign
                <Description>Number of axes this robot arm has</De
                <Value>6</Value>
            </Attribute>
            <Attribute Name="repeatability" AttributeDataType="xs:
                <Description>Repeat accuracy of this robot arm: [m
                <Attribute Name="minRepeat" Unit="mm" AttributeDat
                    <Value>-0.1</Value>
                </Attribute>
                <Attribute Name="maxRepeat" Unit="mm" AttributeDat
                    <Value>0.1</Value>
                </Attribute>
            </Attribute>
            <Attribute Name="temperatureRange" AttributeDataType="
                <Description>Appropriate ambient Temperature for t
                <Attribute Name="minTemperature" Unit="Celsius" At
                    <Value>0</Value>
```

...

# Semantic Lifting – Example

```xml
<SystemUnitClass Name="KR5">
    <Attribute Name="model" AttributeDataType="xs:string" ChangeMo
        <Description>product model of this hardware</Description>
        <Value>KR5</Value>
    </Attribute>
    <Attribute Name="manufacturer" AttributeDataType="xs:string" C
        <Description>manufacturer of this hardware</Description>
        <Value>KUKA</Value>
    </Attribute>
    <Attribute Name="weight" AttributeDataType="xs:double" Unit="k
        <Description>weight of this hardware</Description>
        <Value>127</Value>
    </Attribute>
    <InternalElement Name="RobotArm" ID="5f5f2c62-5ac5-4ca8-8bc8-d
        <RoleRequirements RefBaseRoleClassPath="ReAppRoleClassLib/
            <Attribute Name="payload" Unit="kg" AttributeDataType=
                <Description>Maximum payload of this robot arm</De
                <Value>37</Value>
            </Attribute>
            <Attribute Name="numAxes" AttributeDataType="xs:unsign
                <Description>Number of axes this robot arm has</De
                <Value>6</Value>
            </Attribute>
            <Attribute Name="repeatability" AttributeDataType="xs:
                <Description>Repeat accuracy of this robot arm: [m
                <Attribute Name="minRepeat" Unit="mm" AttributeDat
                    <Value>-0.1</Value>
                </Attribute>
                <Attribute Name="maxRepeat" Unit="mm" AttributeDat
                    <Value>0.1</Value>
                </Attribute>
            </Attribute>
            <Attribute Name="temperatureRange" AttributeDataType="
                <Description>Appropriate ambient Temperature for t
                <Attribute Name="minTemperature" Unit="Celsius" At
                    <Value>0</Value>
```

hasManufacturer(kr5, "KUKA")

hasExternalInterface(kr5, kr5_digitalIn1)

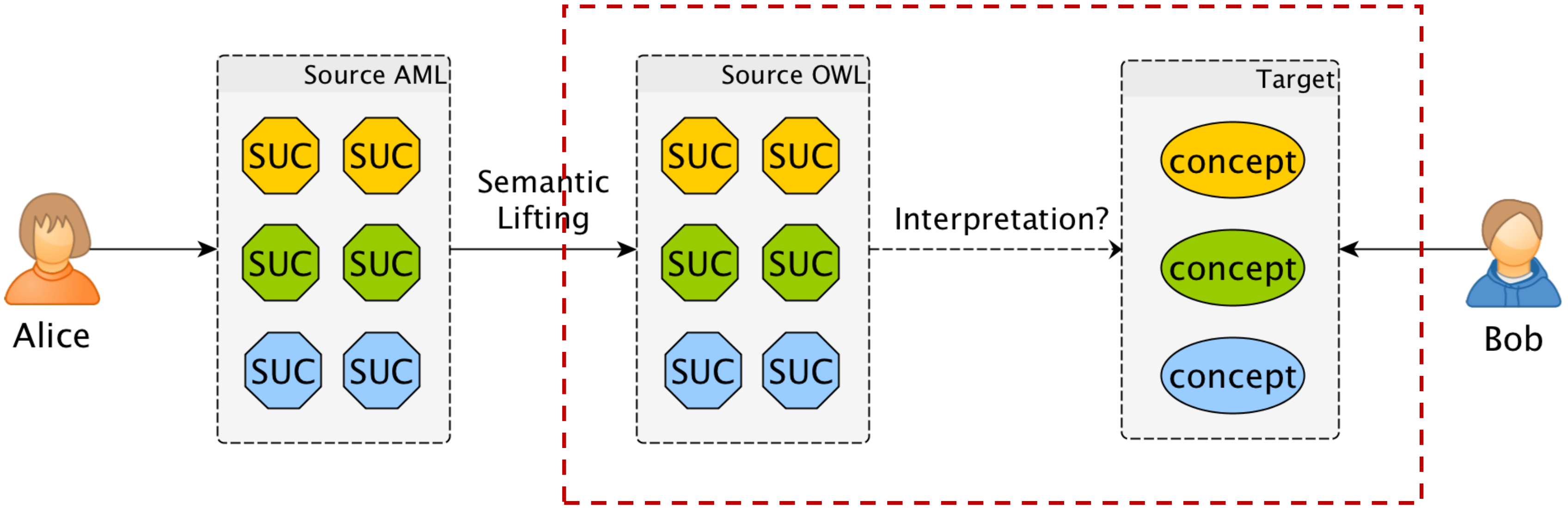
hasInternalElement(kr5, kr5_arm)

DigitalIOInterface(kr5_digitalIn1)

Robot(kr5_arm)

...

Given the **semantic representation** of AML data,
how can we **learn** a **concept description** of the data?



# Concept Learning in AML – Setting

- Input:
  - Background knowledge $\mathcal{K}$: lifted AML
  - Target Concept name $C$
  - Examples (user chosen) $\mathcal{E}$: pos. and neg.
  - **Closed-world assumption**
- Output:
  - Class description of $C$ in OWL 2 DL: $\mathcal{K} \cup C \models \mathcal{E}^+, \mathcal{K} \cup C \not\models \mathcal{E}^-$

$$\boxed{\mathsf{RobotTypeX}} \equiv \exists \boxed{\mathsf{hasWeight}}.[\leq 78.5] \sqcap \exists \boxed{\mathsf{hasInternalElement}}.\boxed{\mathsf{Robot}}$$

Target

Source

AML

# DL-Learner (Bühmann et. al, 2016)

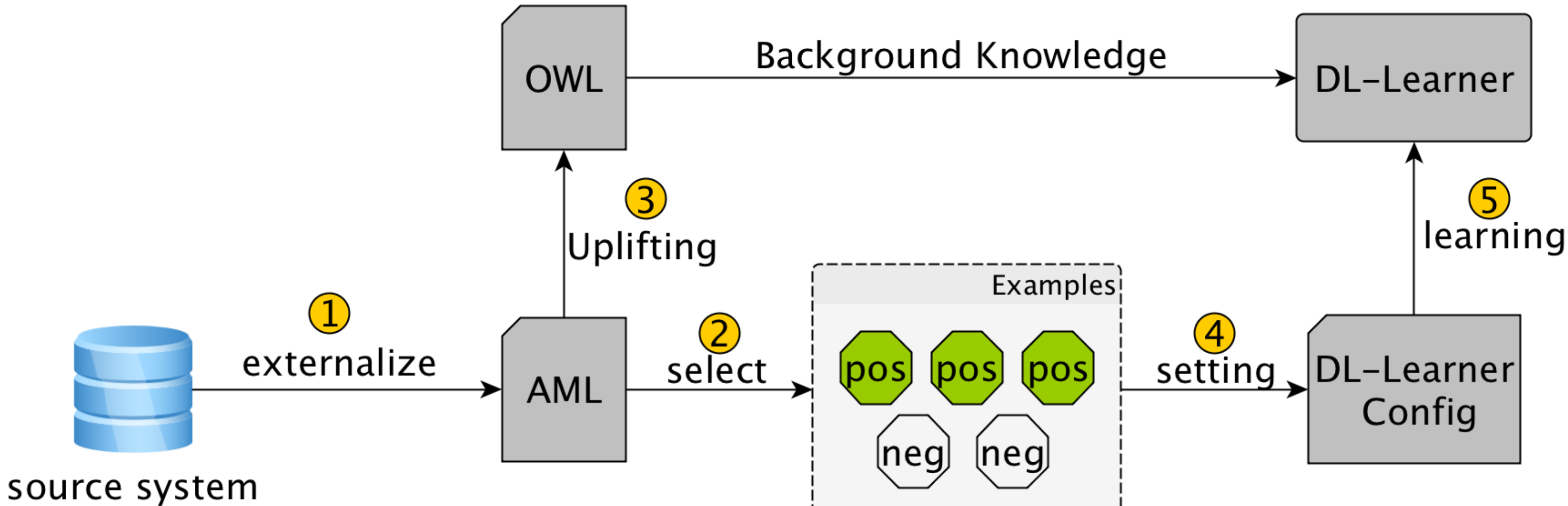
- Framework for concept learning in description logics

- Top-down refinement operators
  - $\mathcal{ALC}$ (complete)
  - $\mathcal{EL}$ (ideal)
  - $\mathcal{ALC}$ with features from OWL 2 DL: concrete roles, cardinality restrictions ...

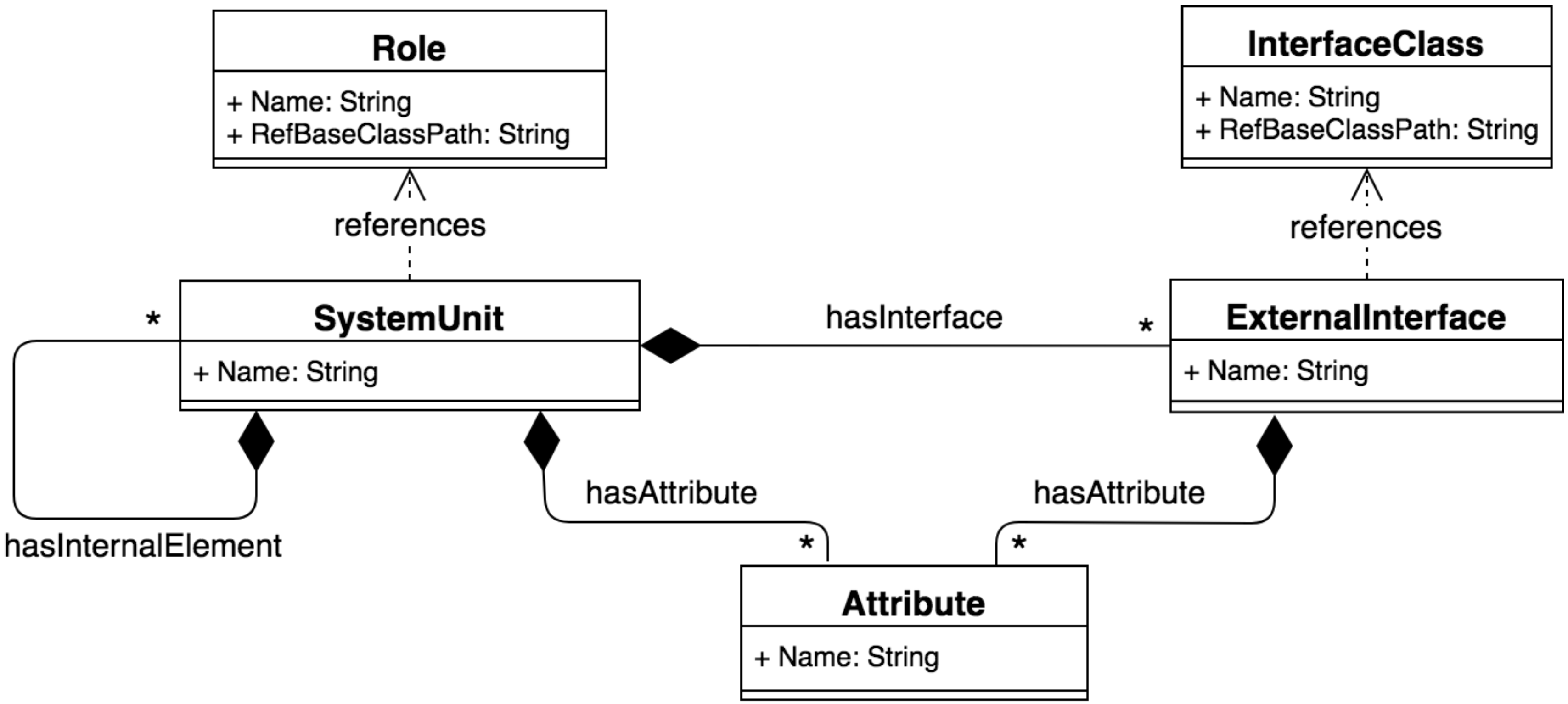
- Learning algorithms for OWL 2 DL:
  - DL-Learner OWL Class Expression Learner (OCEL)
  - Class Expression Learner for Ontology Engineering (CELOE)

- **Partial Closed-World Reasoning**
  - Instance retrieval of named classes before learning: a single model
  - Closed-world reasoning using the single model
  - much faster and more suitable in machine learning setting

# Concept Learning in AML – Pipeline



# Extending the Refinement Operator



# Extending the Refinement Operator

- Use knowledge in the XML schema to restrict the search space
  1. Only external interfaces can reference interface classes
  2. Each external interface can only reference one interface class
  3. A system unit can reference multiple role classes
  4. A system unit can have (recursive) internal structures
  5. An external interface has no internal structure

- Integrate these constraints into the refinement operator

- Implemented on top of DL-Learner

- Can dramatically reduce the number of concept hypotheses

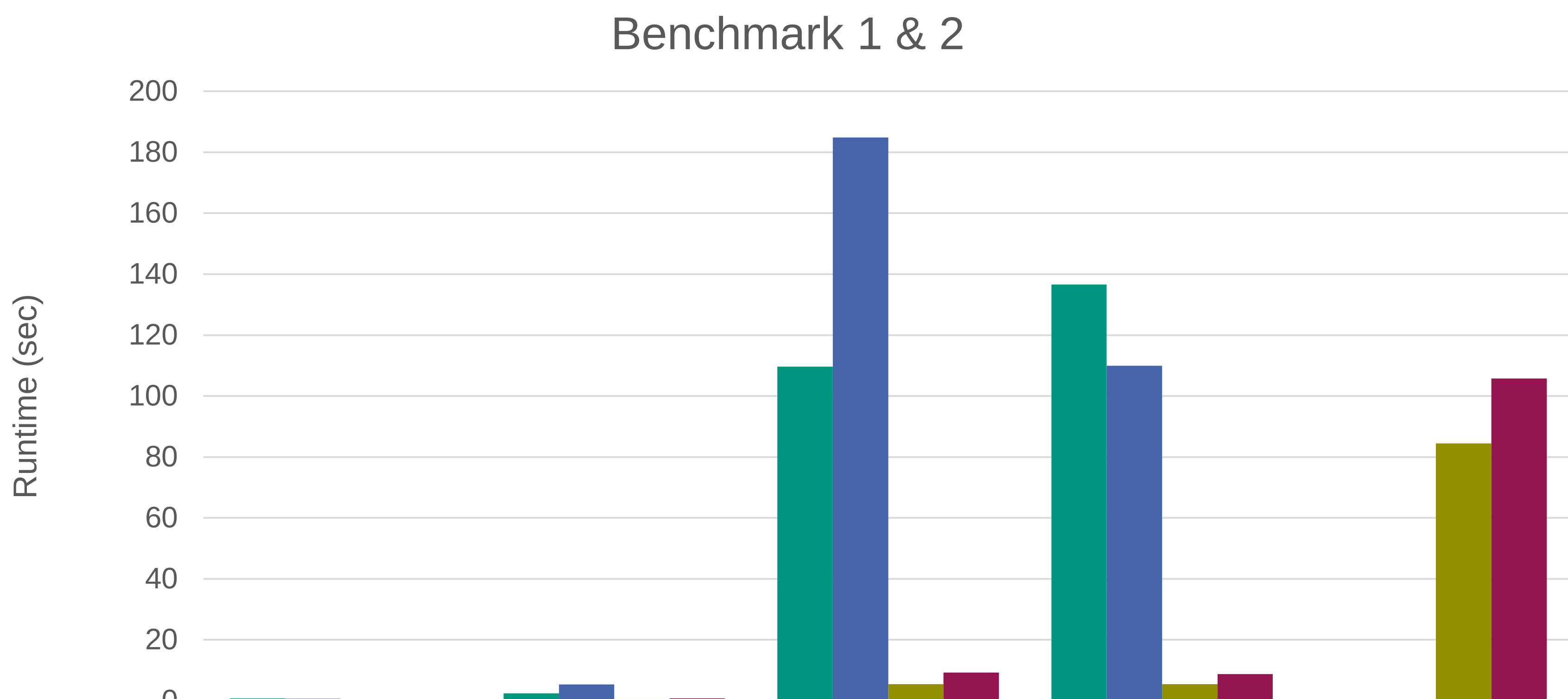
# Extending the Refinement Operator

$$\rho(C) = \begin{cases} \{\bot\} \cup \rho_\top(C) & \text{if } C = \top \\ \rho_\top(C) & \text{otherwise} \end{cases}$$

$$\rho_B(C) = \begin{cases}
\emptyset & \text{if } C = \bot \\
\{C_1 \sqcup \dots \sqcup C_n \mid C_i \in M_B \ (1 \le i \le n)\} & \text{if } C = \top \\
\{A' \mid A' \in sh_\downarrow(A)\} & \text{if } C = A \ (A \in N_C) \\
\quad \cup \{A \sqcap D \mid D \in \rho_B(\top)\} & \\
\{\neg A' \mid A' \in sh_\uparrow(A)\} & \text{if } C = \neg A \ (A \in N_C) \\
\quad \cup \{\neg A \sqcap D \mid D \in \rho_B(\top)\} & \\
\{\exists r.E \mid A = ar(r), E \in \rho_A(D)\} & \text{if } C = \exists r.D \\
\quad \cup \{\exists r.D \sqcap E \mid E \in \rho_B(\top)\} & \\
\quad \cup \{\exists s.D \mid s \in sh_\downarrow(r)\} & \\
\{\forall r.E \mid A = ar(r), E \in \rho_A(D)\} & \text{if } C = \forall r.D \\
\quad \cup \{\forall r.D \sqcap E \mid E \in \rho_B(\top)\} & \\
\quad \cup \{\forall r.\bot \mid & \\
\qquad D = A \in N_C \text{ and } sh_\downarrow(A) = \emptyset\} & \\
\quad \cup \{\forall s.D \mid s \in sh_\downarrow(r)\} & \\
\{C_1 \sqcap \dots \sqcap C_{i-1} \sqcap D \sqcap C_{i+1} \sqcap \dots \sqcap C_n \mid & \text{if } C = C_1 \sqcap \dots \sqcap C_n \\
\qquad D \in \rho_B(C_i), 1 \le i \le n\} & (n \ge 2) \\
\{C_1 \sqcup \dots \sqcup C_{i-1} \sqcup D \sqcup C_{i+1} \sqcup \dots \sqcup C_n \mid & \text{if } C = C_1 \sqcup \dots \sqcup C_n \\
\qquad D \in \rho_B(C_i), 1 \le i \le n\} & (n \ge 2) \\
\quad \cup \{(C_1 \sqcup \dots \sqcup C_n) \sqcap D \mid & \\
\qquad D \in \rho_B(\top)\} &
\end{cases}$$

# Experiment Results – Performance

Karlsruhe Institute of Technology




| | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| default (b1) | 0,781 | 2,38 | 109,66 | 136,625 | |
| default (b2) | 0,704 | 5,333 | 184,86 | 109,9 | |
| aml (b1) | 0,508 | 0,595 | 5,483 | 5,465 | 84,419 |
| aml (b2) | 0,451 | 0,8 | 9,243 | 8,792 | 105,779 |

# Summary

✓ Pipeline of concept learning in AML using DL-Learner

✓ Extension of the default $\mathcal{ALC}$ refinement operator

➢ Application in data exchange

➢ Investigate other semantic languages and refinement operators

➢ Better searching algorithms

➢ Self-adaptive heuristics (parameter learning)

➢ Bottom-up approaches

# References


- N. Schmid and A. Lüder, *“AutomationML in a Nutshell”*, November 2015.
- M. Uschold, “Where are the semantics in the semantic web?” AI Mag., vol. 24, no. 3, Sept. 2003
- AutomationML e.V., *“Whitepaper AutomationML Part 1 – Architecture and general requirements”*, July 2013.
- S. Runde, K. Güttel and A. Fay, *“Transformation von CAEX-Anlagenplanungsdaten in OWL: Eine Anwendung von Technologien des Semantic Web,“* in Automation 2009, June 2009
- L. Bühmann, J. Lehmann, and P. Westphal, *„DL-Learner: A Framework for Inductive Learning on the Semantic Web,“* Web Semantics, vol. 39, Aug. 2016.

# Thank you for the attention!

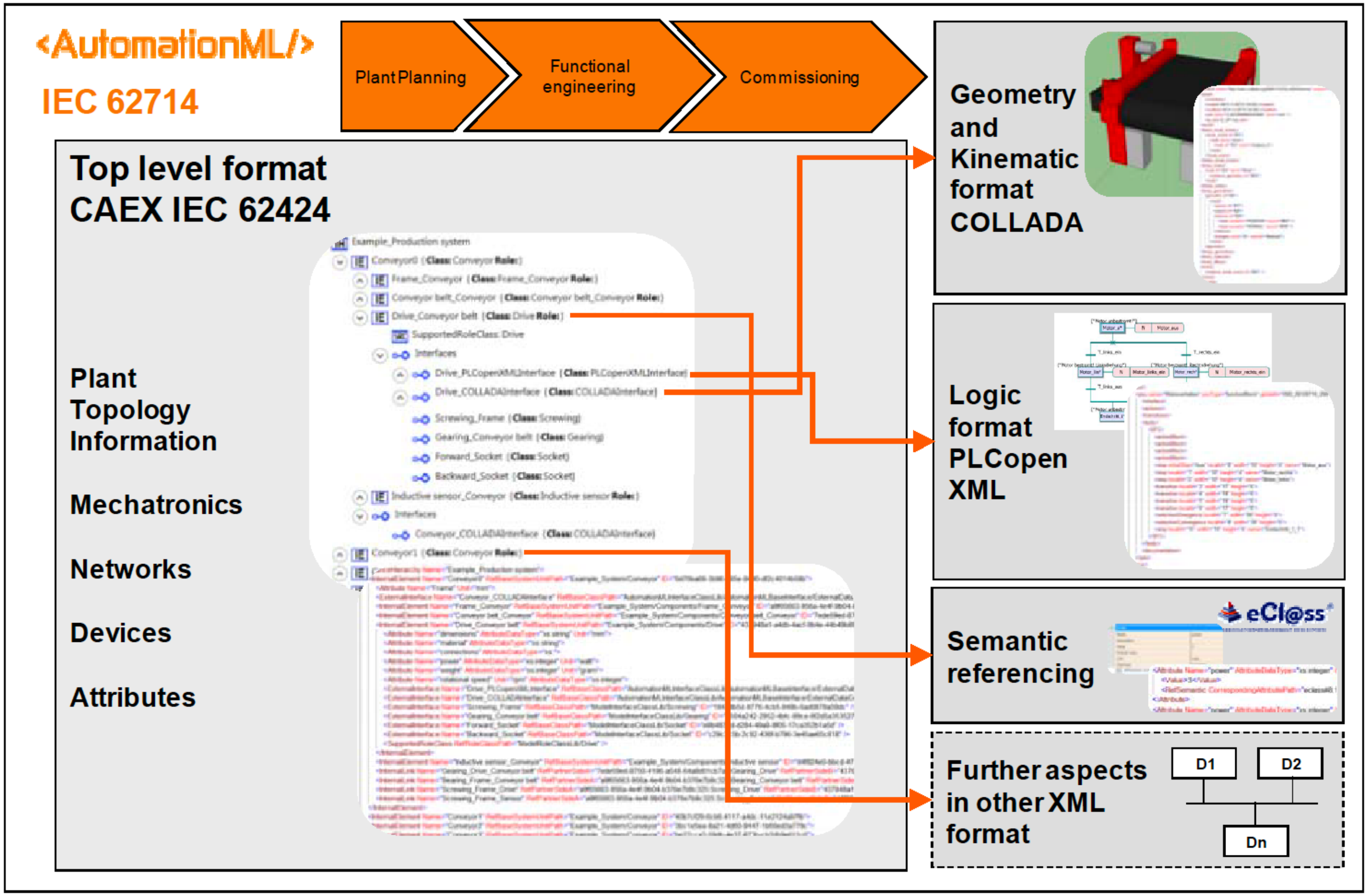
# Any Questions?

Karlsruhe Institute of Technology

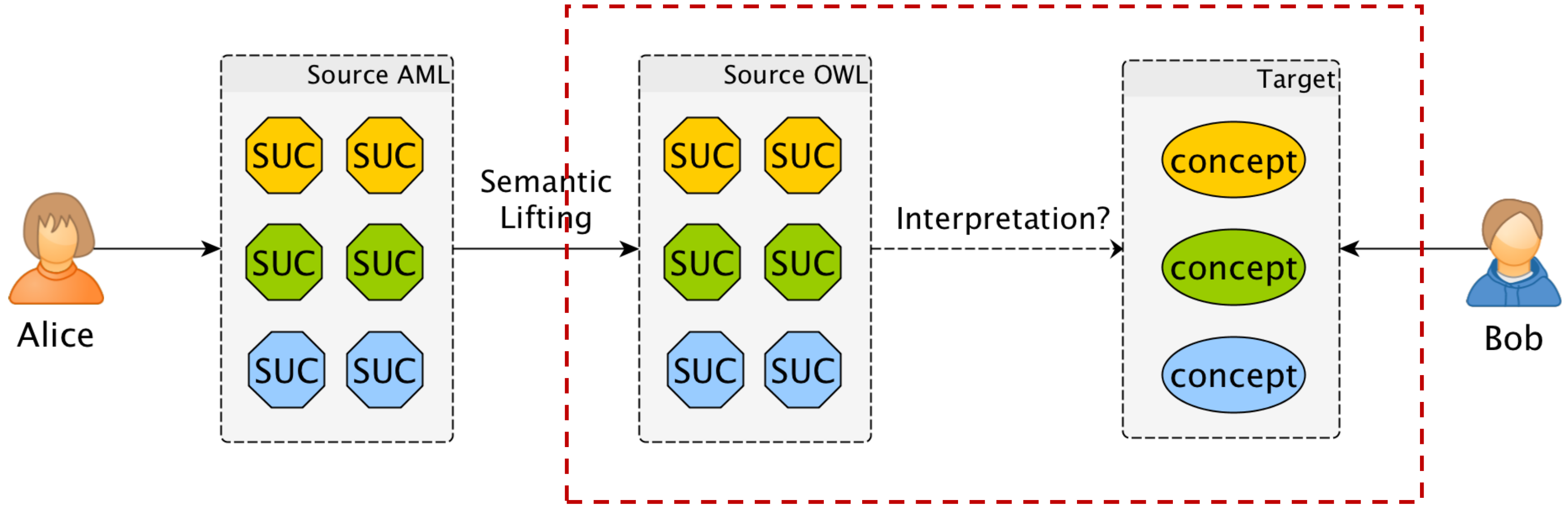
# AutomationML (AML)

- The **Automation M**ark-up **L**anguage
- International standard as IEC 62714
- Data modeling and exchange in the field of production systems engineering and commissioning
- XML-based

# AutomationML (AML)



We need a **formal semantic** representation of AML data for concept learning

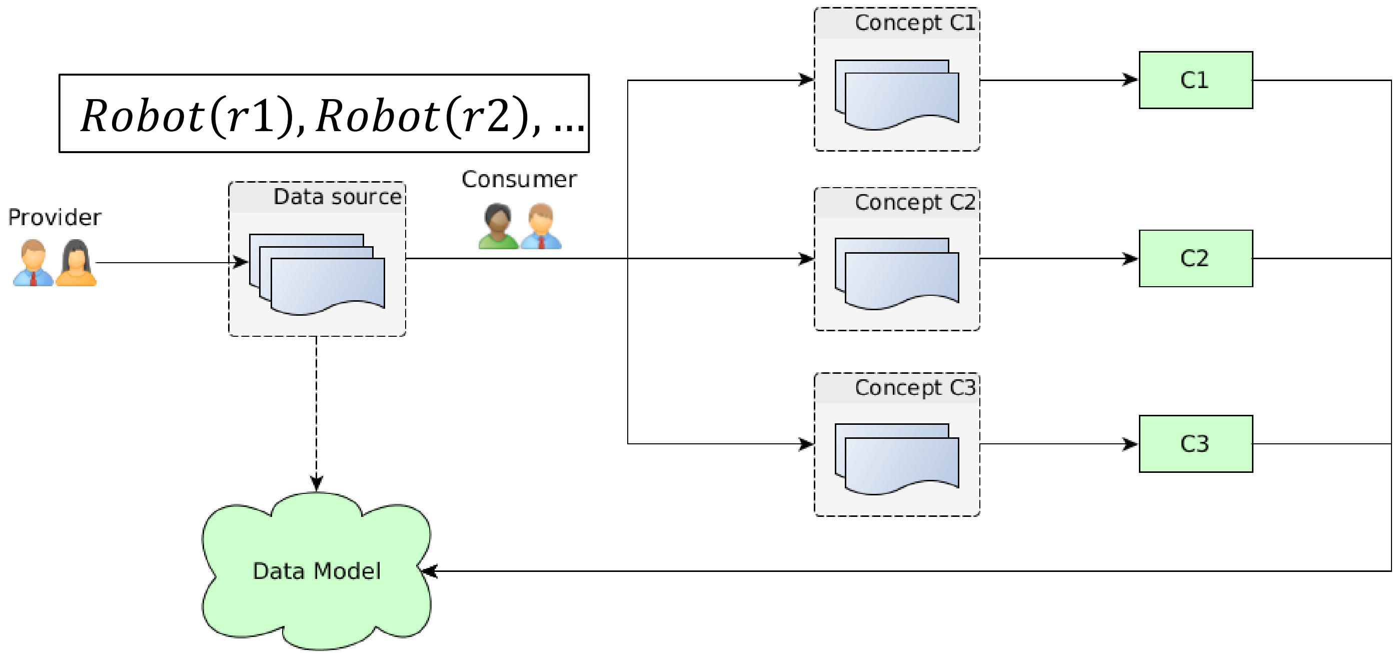


# Concept Learning in AML – Example

$$C1 = Robot \sqcap hasEndEffector.Gripper$$

$$Ind^{+}_{C1}(\mathcal{A}) = \{r1, r2, \dots\}$$

$$Robot(r1), Robot(r2), \dots$$




$$T = \{Robot, Sensor, Tool, \dots\}$$
$$R = \{hasPayload, hasEndEffector, \dots\}$$

# Experiments

- Source AML document:
  - 220 classes
  - 497 individuals
  - 73 data properties

- Two benchmarks
  - +50 role classes, +25 interface classes
  - +100 role classes, +50 interface classes

- Measure time until first 100% accurate solution:  synthetic ground truths
  - default refinement operator from DL-Learner
  - extended AML refinement operator