University of Stuttgart
Institute of Industrial Automation
and Software Engineering

Refining Automation System for Enhanced Modular Control

Beryl Wicaksono
Electrical Engineering
Supervisor: Yuchen Xia

MT-3610

Conceptional Project (co-pm)

Motivation and Background

Motivation and Background

Refining automation system

Why?

- Flexible production system
- Easier reconfiguration

Modular Data and Control Interface

Why?

IT

Information Technology

Goal: Modular Data and Control Interface to connect IT and OT

OT

Operational Technology

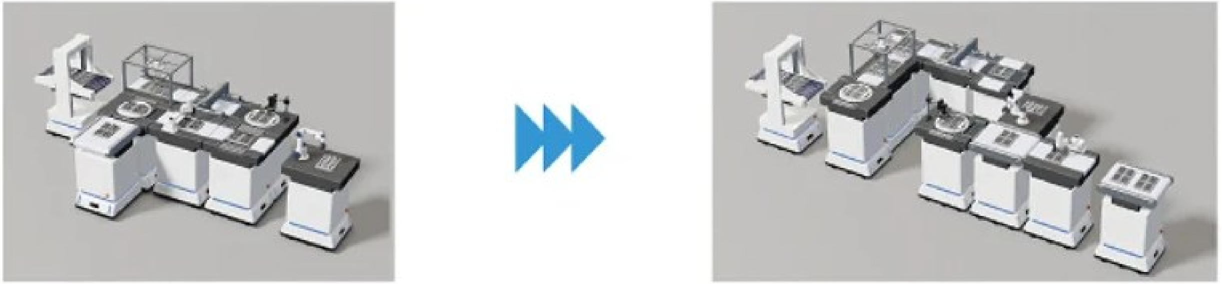
Motivation and Background

- Refine automation system into modular data and control interface.

- “**Modular**” : smaller parts of a whole that can be arrange differently.
- “**Automation**” : activity that is done with minimal human command.
- “**Modular Automation**” : production line that can be separated into mobile modules and can be quickly reconfigured for different purpose

IAS CP-Factory

From Layout A to Layout B



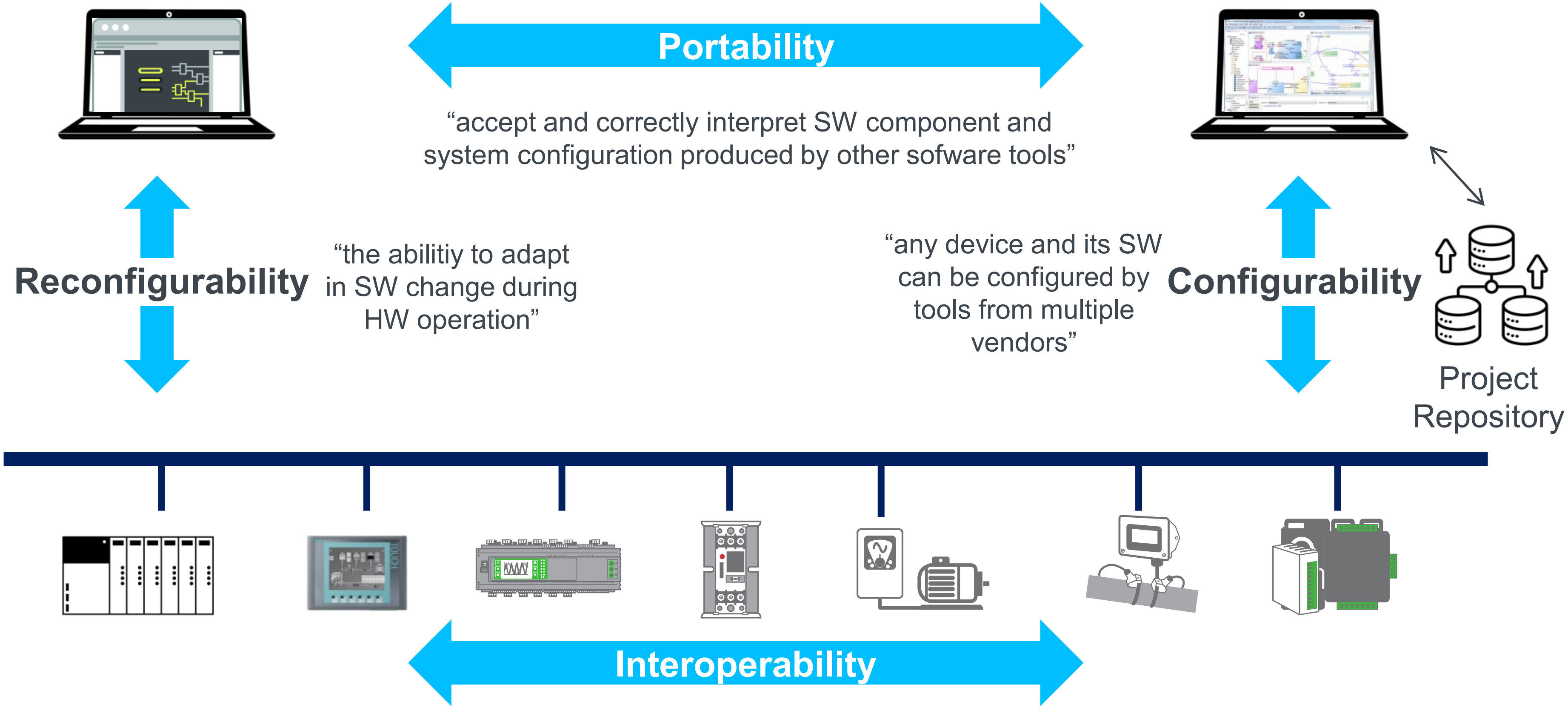
Layout A

Layout B

Source : Oriental motor [1]

The benefits of modular automation

“Characterized by Portability, Configurability, Interoperability”



Portability

“accept and correctly interpret SW component and system configuration produced by other sofware tools”

Reconfigurability

“the abilitiy to adapt in SW change during HW operation”

“any device and its SW can be configured by tools from multiple vendors”

Configurability

Project Repository

Interoperability

“embedded devices can operate together to perform the functions needed for distributed applications”

Goal and Requirements

Goal: Modular Data and Control Interface to connect IT and OT

1. Modular **data query** interface
2. Modular **control** interface

- **In the context IAS CP-Factory, it is so difficult to have modular control interface.**
- **R1** : PLC code IEC 61131 is not modular.
  - It cannot be easily ported on another PLC vendor IDE
  - It cannot be reconfigurable by another PLC vendor IDE
  - Interoperability is not possible through different device.
- **R2** : Production control has constraint to production process.
  - inherent dependency

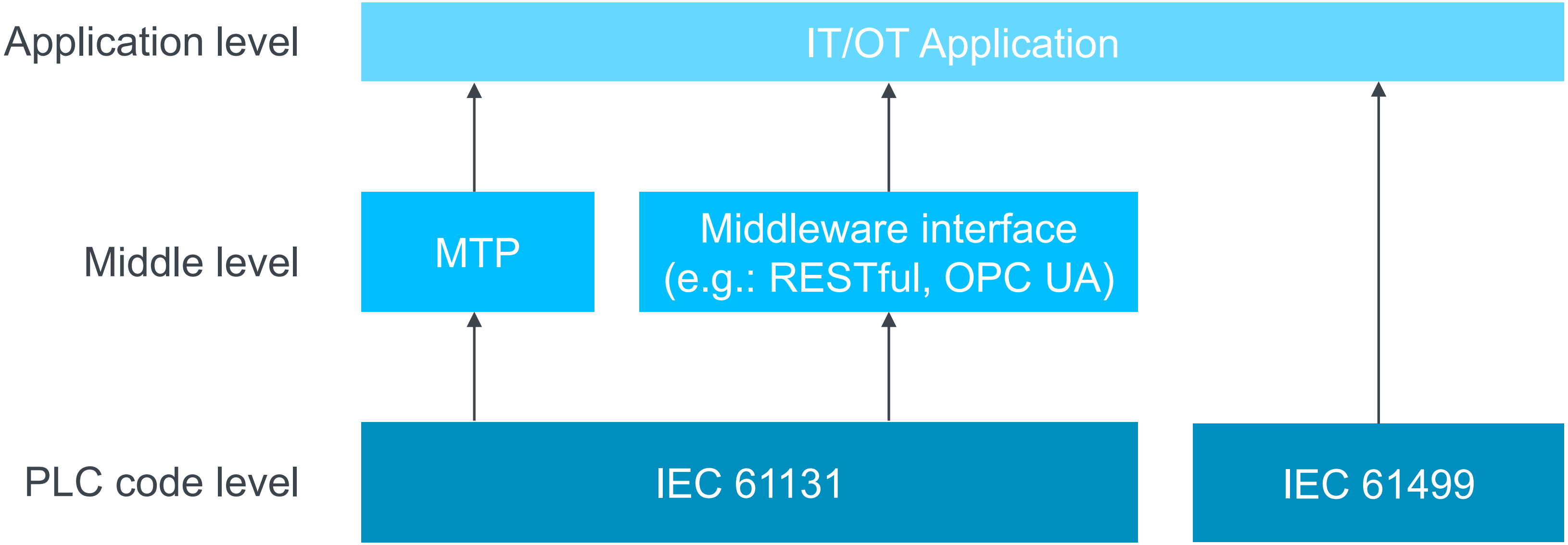
"We want to explore whether we can ***migrate*** *and* ***translate*** *the IEC 61131 standard for modular data and control interface with high code utilization"*

State of the Art

The 3 modular control interface approach

- Interface based on IEC 61131
- Module Type Package
- IEC 61499 Migration

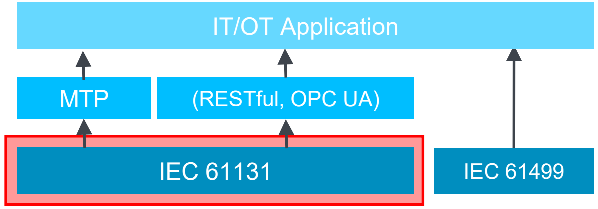
State of the Art

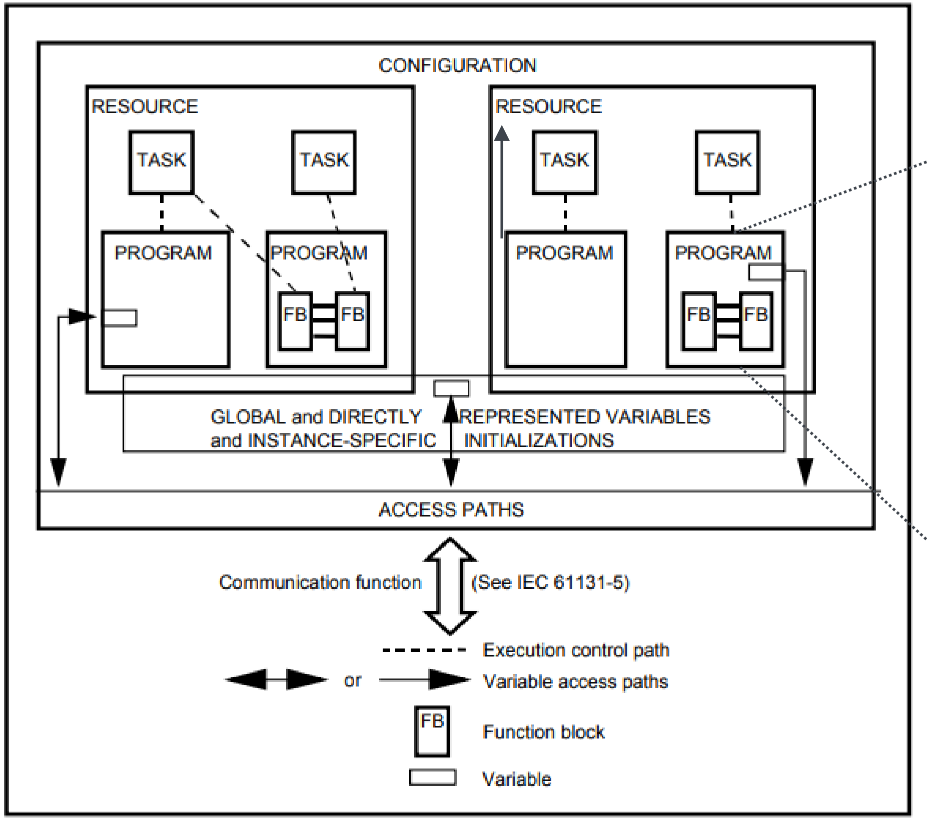


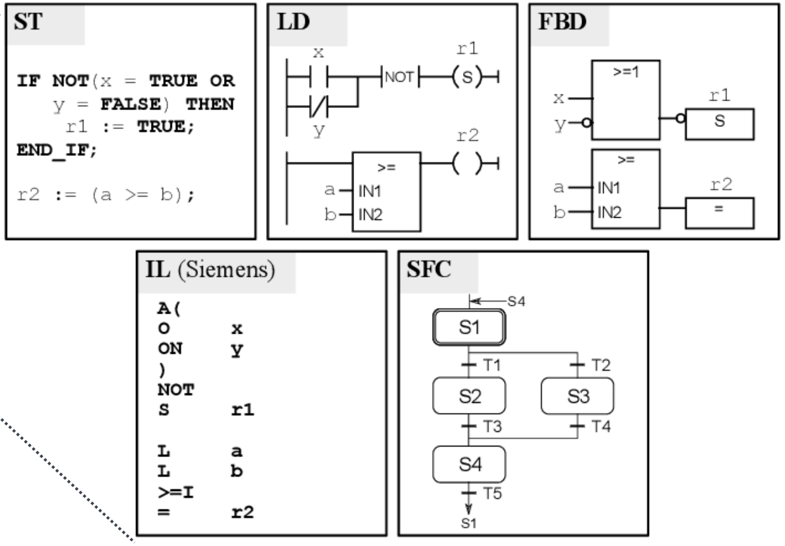
- 1. modular interface
- 2. code utilization

Interface based on IEC 61131

Overview of IEC 61131







ST

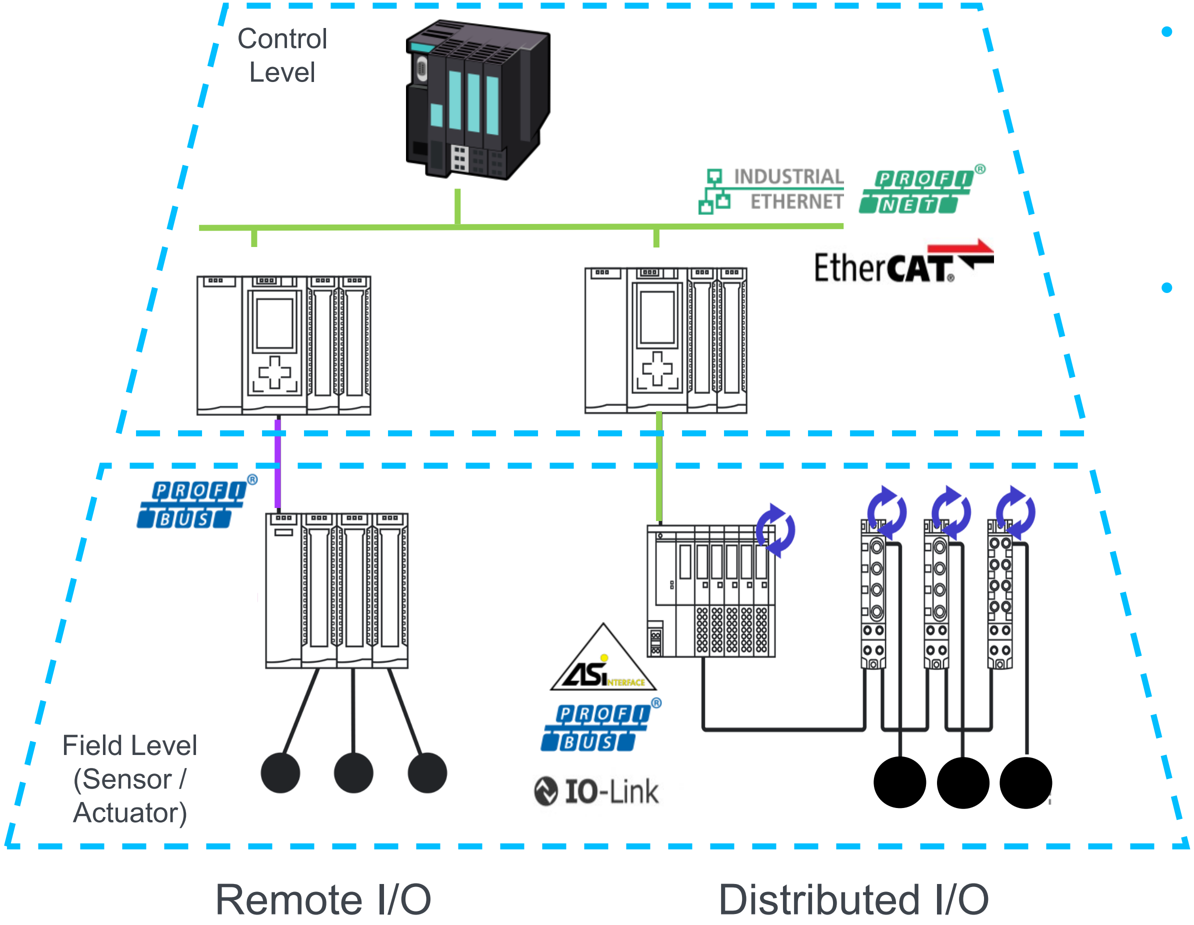
```
IF NOT(x = TRUE OR
   y = FALSE) THEN
    r1 := TRUE;
END_IF;

r2 := (a >= b);
```

IL (Siemens)

```
A(
O      x
ON     y
)
NOT
S      r1

L      a
L      b
>=I
=      r2
```

Interface based on IEC 61131 Communication Paradigm

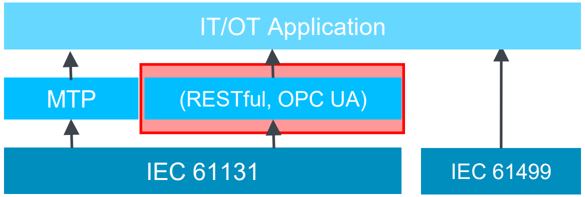
"Precise cyclic scan based architecture" [4]

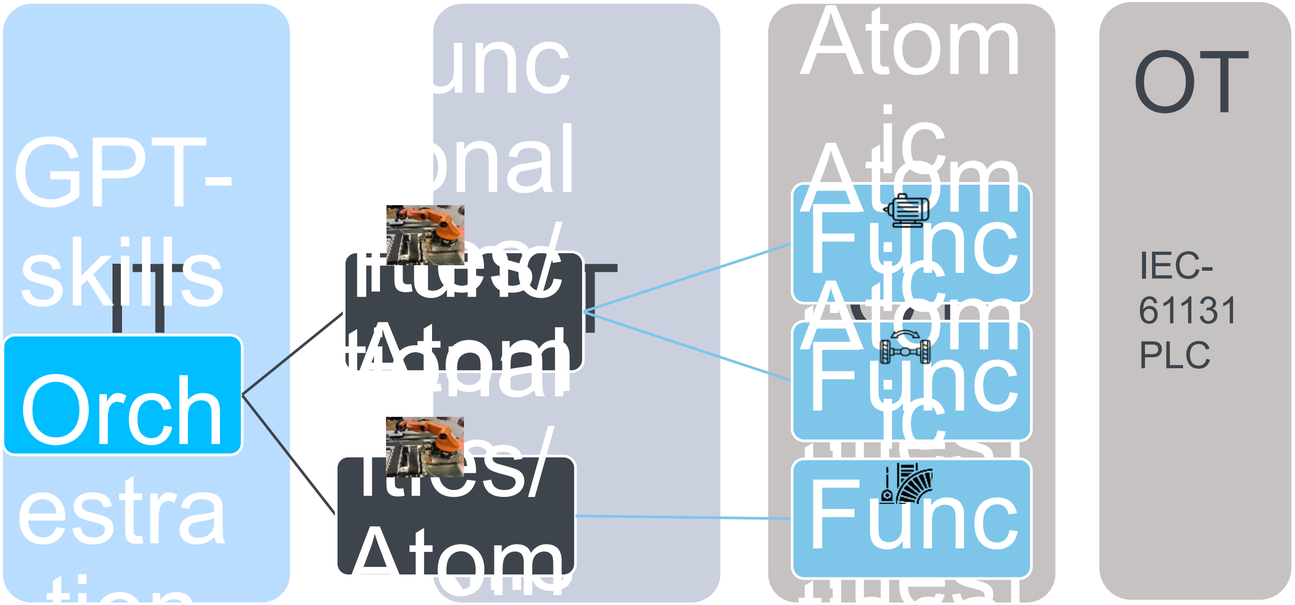


- **Advantages** :
  - Robust : ensure synchrony of data
  - Deterministic behaviour
- **Disadvantages** :
  - Ownership issues : a DO or AO must be owned by one PLC only.
  - Extra communication between PLC is needed to pass program and I/O data
  - Reduce performance and reliability because copying data between PLCs. (Reaction time increase)

Previous Work at IAS

Connect IT and OT via RESTful Interface (MT-3498Shenoy)





The task: Inspect and paint the workpiece

controlStopper (id=1,state= “down” , param = “fault”)

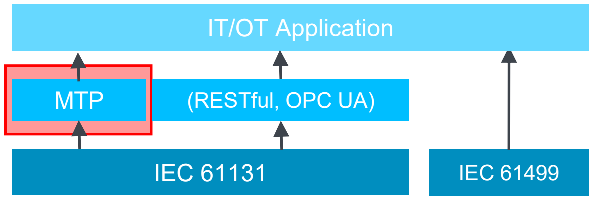
```
Opcua-client.get_node('ns=3;s="xW1_MB50“')
Opcua-client.get_node('ns=3;s="xW1_BG21“')
…..
```

{REST:API}

OPC UA

Module Type Package (MTP)

Modularize for process industry



IT/OT Application

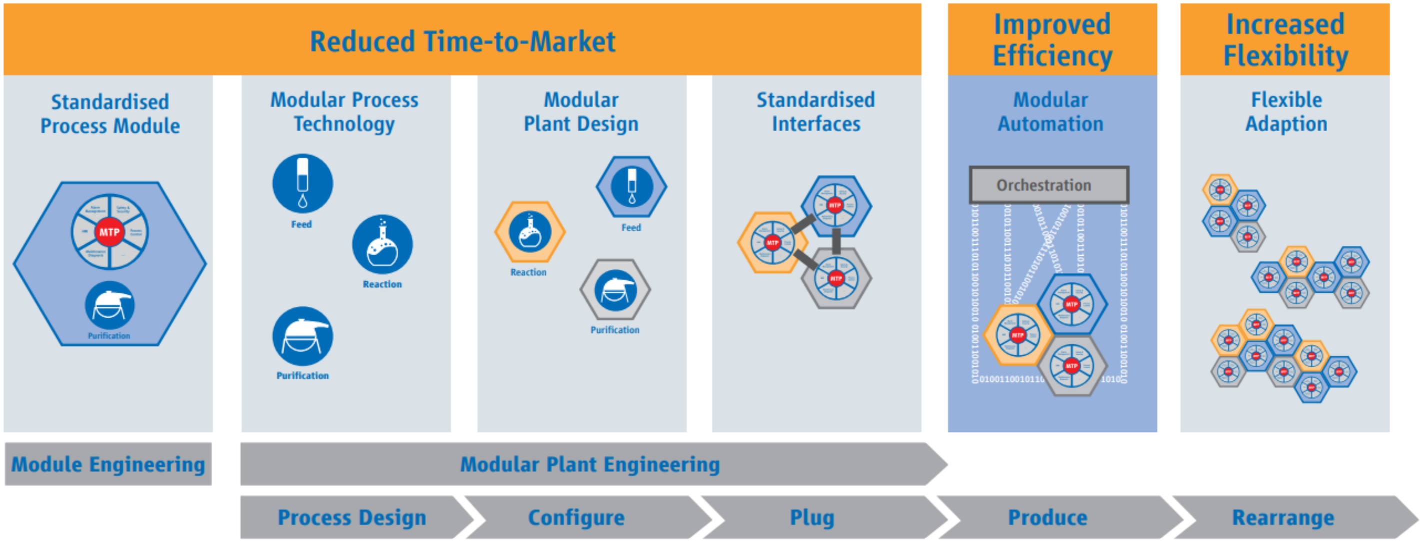
MTP

(RESTful, OPC UA)

IEC 61131

IEC 61499

- Main Idea : Modularize each process automation as **Process Equipment Assembly** (PEA) according to VDI/VDE/NAMUR Standard 2658.
  Several PEAs with controllers are integrated to **Process Orchestration Layer** (POL).
- Concept and Benefit:



Reduced Time-to-Market

Improved Efficiency

Increased Flexibility

Standardised Process Module

Modular Process Technology

Modular Plant Design

Standardised Interfaces

Modular Automation

Flexible Adaption

Module Engineering

Modular Plant Engineering

Process Design

Configure

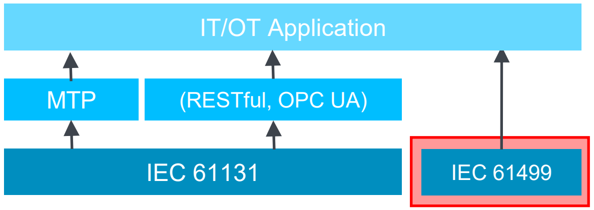
Plug

Produce

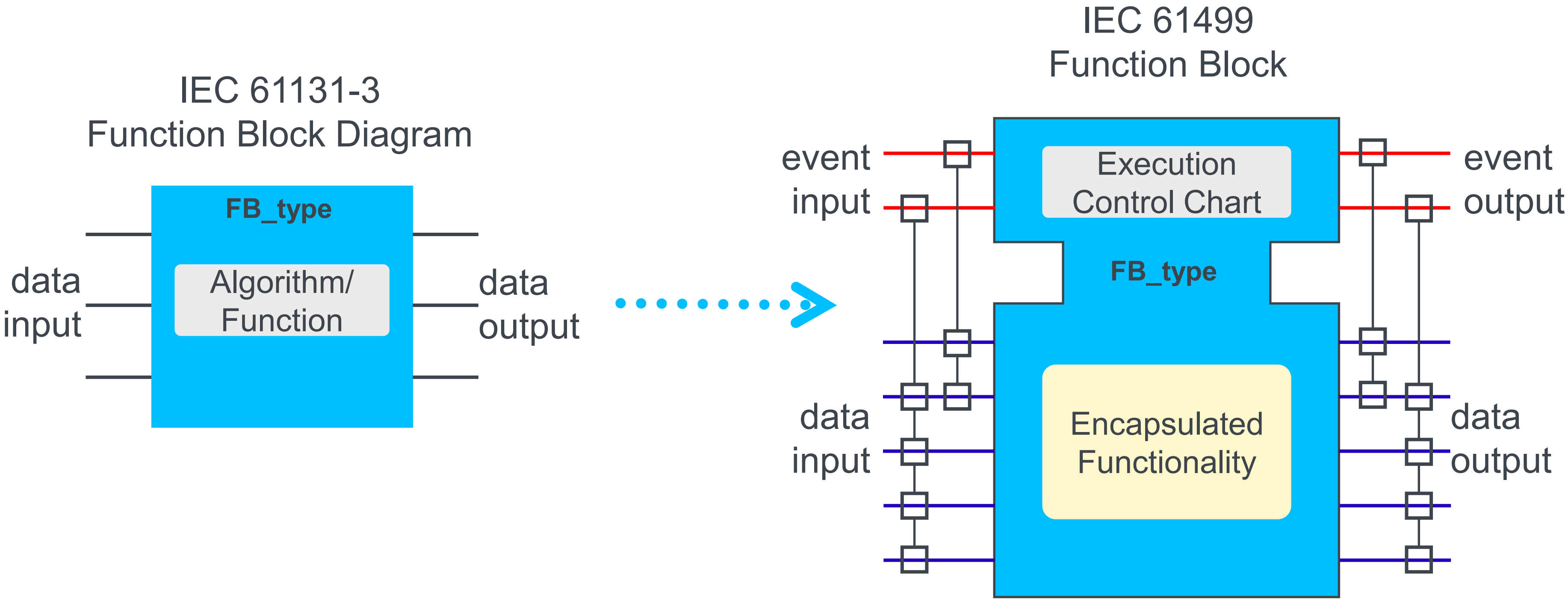
Rearrange

[2] Source: NAMUR, ProcessNet, ZVEI

Overview of IEC 61499



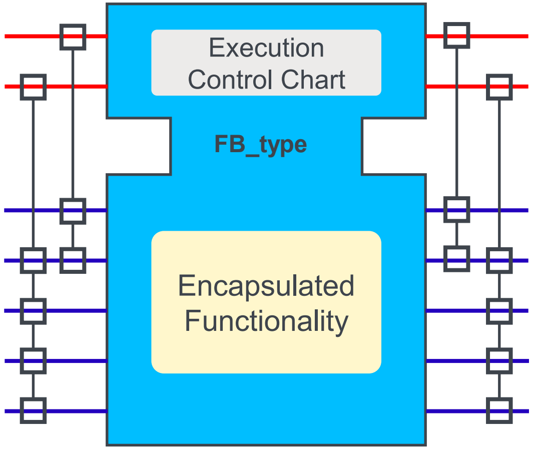
- IEC 61499 is **not programming language.**
- IEC 61499 is System-level design language for distributed control systems.



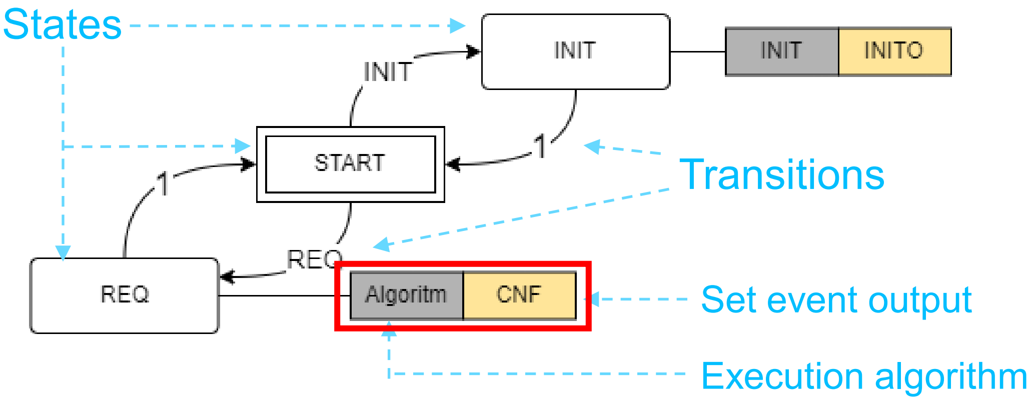
Overview of IEC 61499

System level design language to model the distributed control systems

1. Basic Function Block (BFB)



Execution Control Chart (ECC)

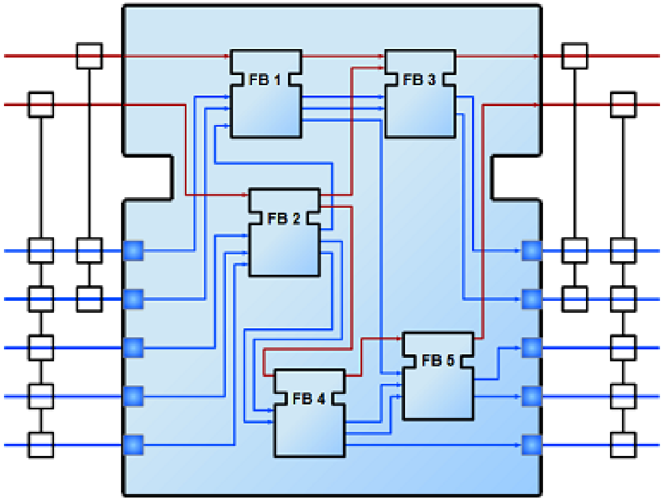


- The behavioral of BFB is defined as **event-driven state machine**, called **ECC**.
- When transition is fired, **active state** changes. (Only 1 active state at any one time)
- When a **new state** is entered, the corresponding **Actions** are executed
- The algorithm can be written in <u>any programming language</u>.
  If BFB defines strong encapsulation algorithm : **key characteristic of SW Component**

Overview of IEC 61499

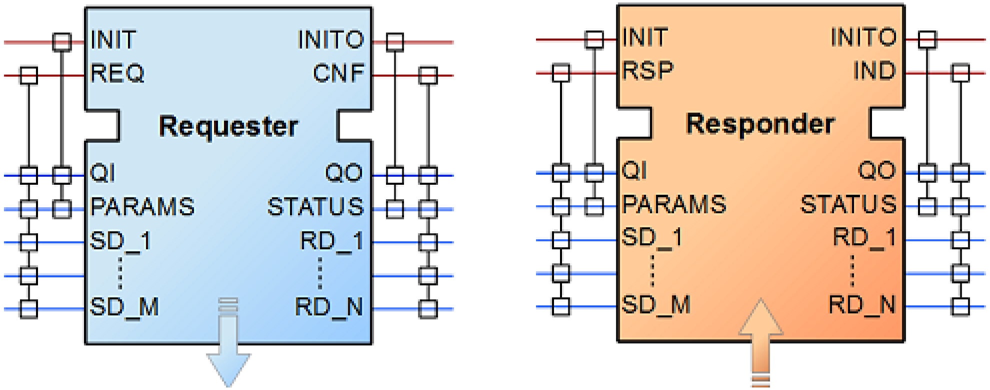
System level design language to model the distributed control systems

2. Composite Function Block (CFB)



Orchestration of modular functions with good interface

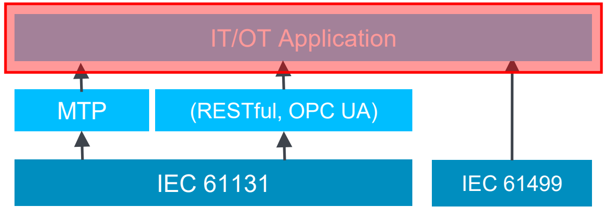
3. Service Interface Function Block (SIFB)

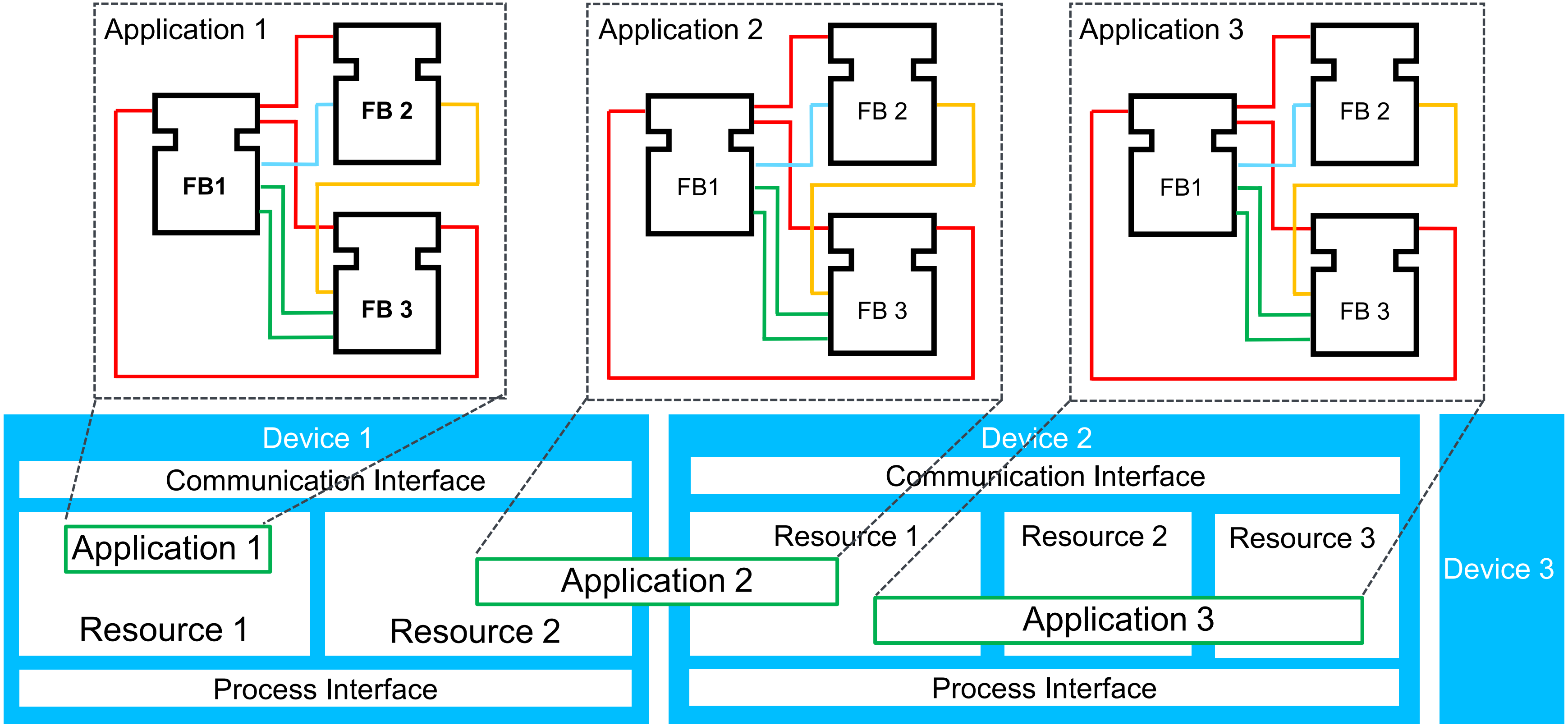


Interface for external interaction

Architecture of IEC 61499

Application model independent of underlying system model





Conclusion on IEC 61499

Seamless OT-IT Integration

OT- World

- **Automation Engineer**
  - Graphical function block
  - Precise control of execution order and data flow
- **Periodic scan based**

IT- World

- **Software Engineer**
  - Object
  - Methods
  - Sofware component
- **Transaction based / Event based**

- Event-driven software component.

  Software component can be used by other person to build complete application by plugging together software component.
- Independent of underlying hardware device.
- It can be distributed across different hardware devices.

Conception

Migration to IEC 61499 for CP factory

Conception

Migration from IEC 61131 to IEC 61499 Standard

Run IEC 61131

IAS CP-Factory

Translation

Run IEC 61499

IAS CP-Factory

Goal: Modular Data and Control Interface to connect IT and OT

- Easier interaction and integration with external system
- Modular interface

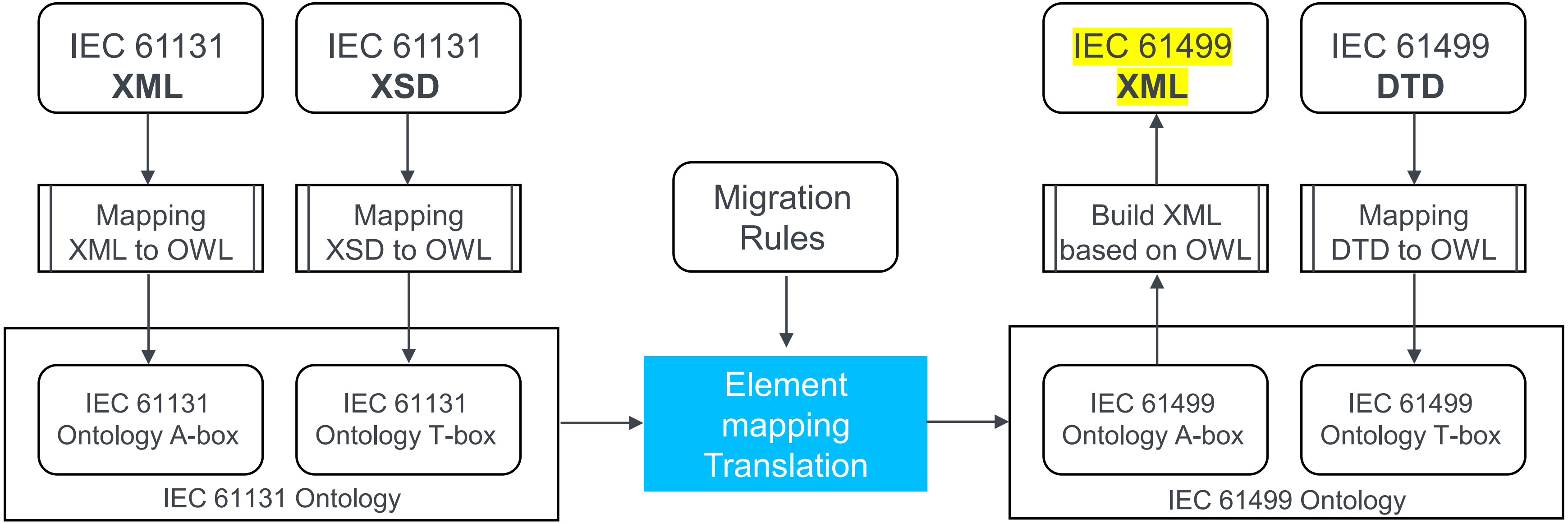
Migration Strategy from IEC 61131 to IEC 61499

- Three Approach for migration to IEC 61499 :
  - ○ Model-Driven Approach [4] [5] [6]
  - ○ Object/Class-Oriented Approach [7]
  - ○ Ontology-Based Approach [8] [9]

- All approach start with same procedure :
  - ○ Identification structure of XML file IEC 61131 and IEC 61499
  - ○ Investigate the possibility mapping between two XML

Migration to IEC 61499 Strategy

Concept of Ontology Mapping

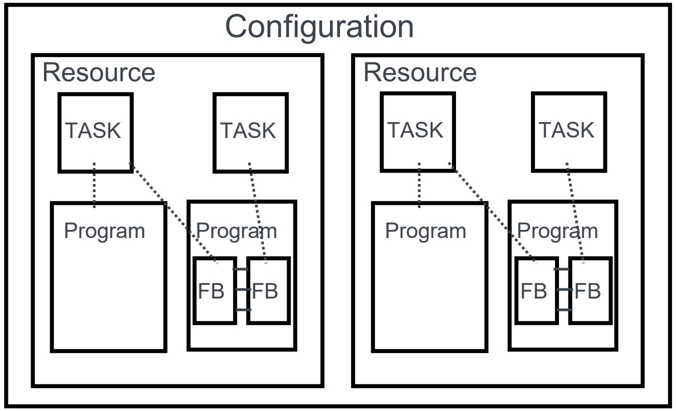


- T-Box (Taxonomy Box) : The knowledge base of all properties standard and its relationships between concepts.
- A-Box (Assertion Box) : The instances (system configuration, function block, code) are modelled. It consists of knowledge which is specific to the individual system design.

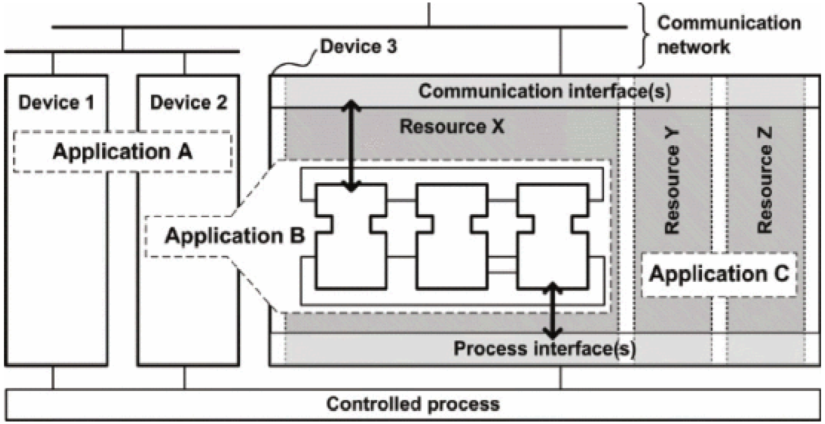
Migration to IEC 61499

Migration Rules

IEC 61131



IEC 61499



Cited from [10]

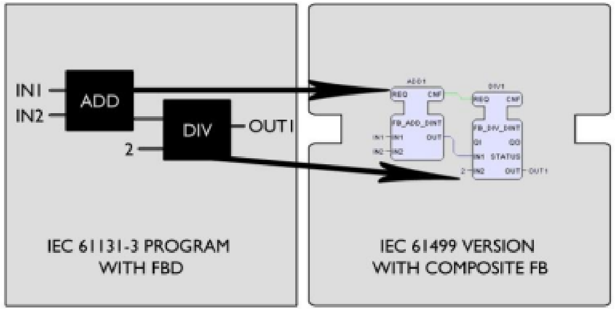
- Rules :

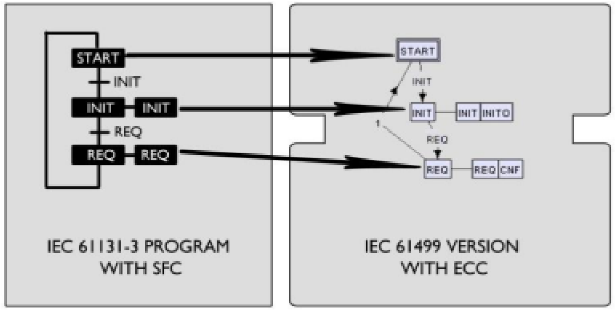
  - IEC 61131 **Resource** ➡ IEC 61499 **Device**
  - IEC 61131 **Task** ➡ IEC 61499 **Resource**
  - IEC 61131 **Program** ➡ IEC 61499 **Application**

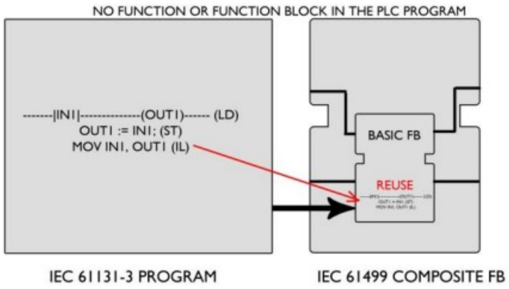
Migration to IEC 61499

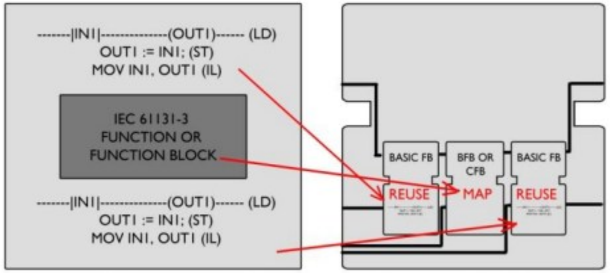
Migration Rules

- IEC 61131 Program in FBD → IEC 61499 CFB
- IEC 61131 Program in SFC → IEC 61499 ECC
- IEC 61131 Program → IEC 61499 BFB if no FC / FBD calls in original function block
- IEC 61131 Program → IEC 61499 CFB if one / more FC / FBD calls in original funtion block


IN1
IN2
ADD
DIV
2
OUT1
IEC 61131-3 PROGRAM WITH FBD
IEC 61499 VERSION WITH COMPOSITE FB

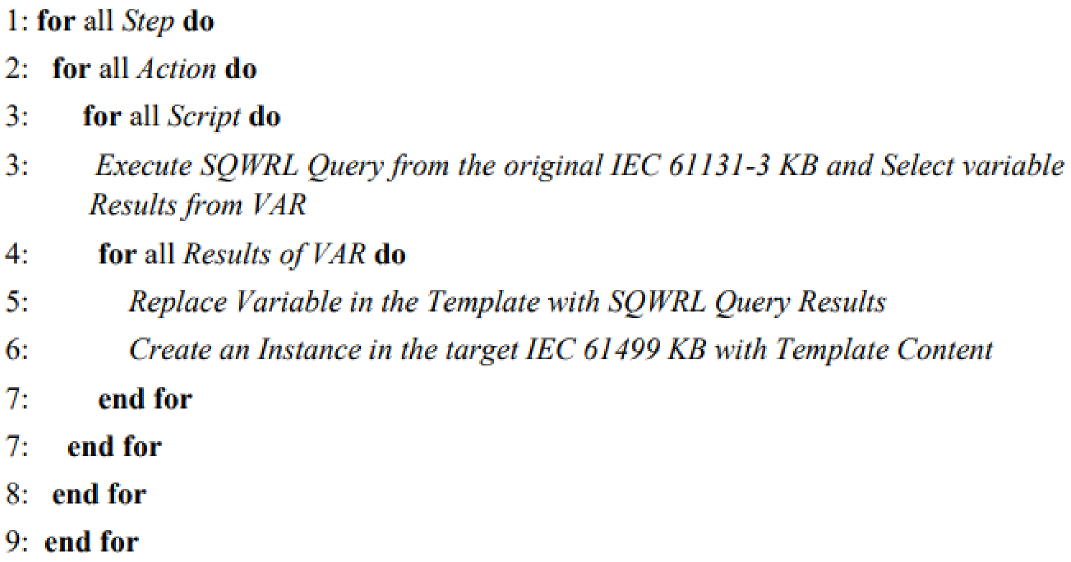



NO FUNCTION OR FUNCTION BLOCK IN THE PLC PROGRAM
-------|IN1|--------------(OUT1)------ (LD)
OUT1 := IN1; (ST)
MOV IN1, OUT1 (IL)
BASIC FB
REUSE
IEC 61131-3 PROGRAM
IEC 61499 COMPOSITE FB

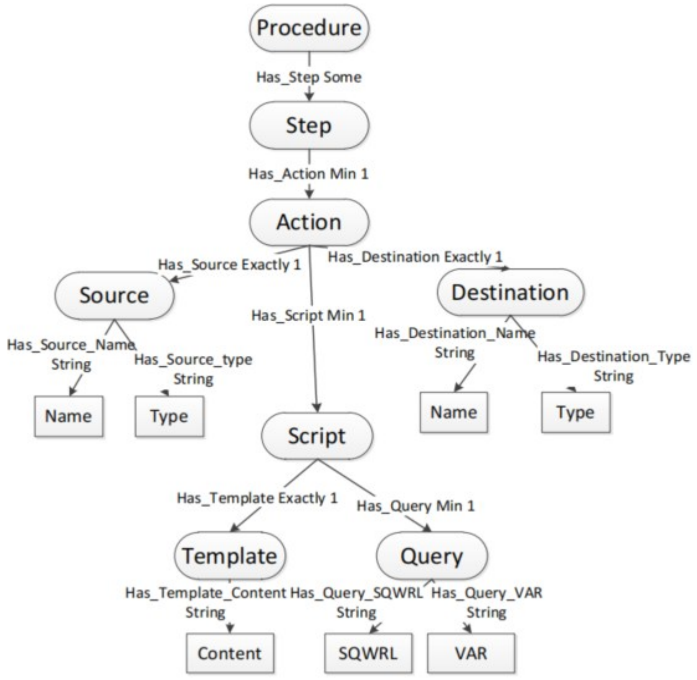

-------|IN1|--------------(OUT1)------ (LD)
OUT1 := IN1; (ST)
MOV IN1, OUT1 (IL)
IEC 61131-3 FUNCTION OR FUNCTION BLOCK
-------|IN1|--------------(OUT1)------ (LD)
OUT1 := IN1; (ST)
MOV IN1, OUT1 (IL)
BASIC FB
BFB OR CFB
BASIC FB
REUSE
MAP
REUSE

Cited from [10]

Ontology Mapping



```
for all Step do
  for all Action do
    for all Script do
      Execute SQWRL Query from the original IEC 61131-3 KB and Select variable
      Results from VAR
      for all Results of VAR do
        Replace Variable in the Template with SQWRL Query Results
        Create an Instance in the target IEC 61499 KB with Template Content
      end for
    end for
  end for
end for
```



Cited from [10]

Implementation

Migration to IEC 61499

Creation of target IEC 61499 XML File

Document Type Definition (DTD) of IEC 61499

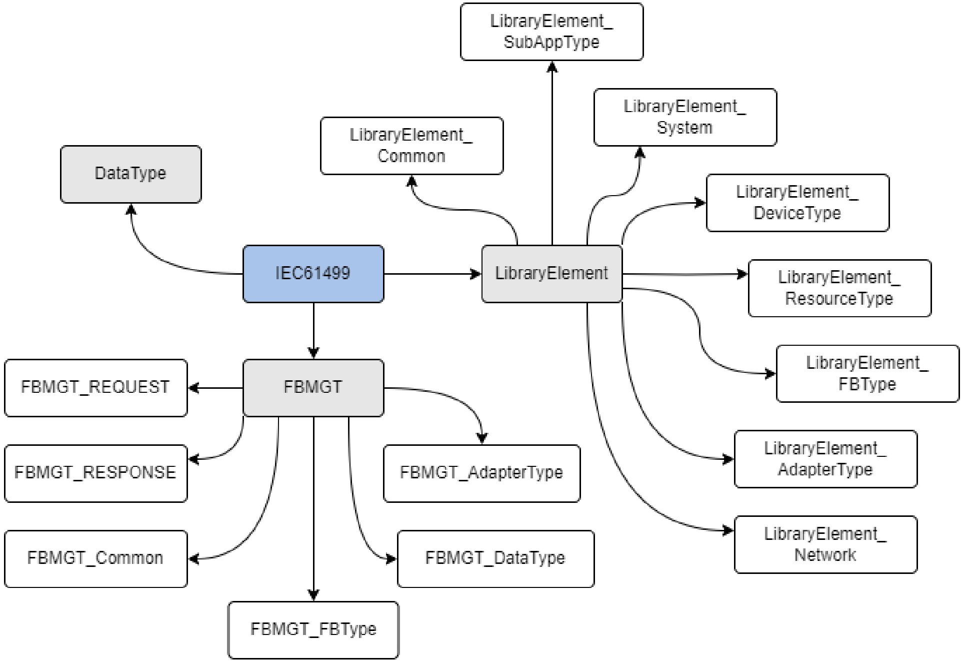
- DTD defines the structure and the legal elements and attributes of an XML.
- XML document validated against DTD : "Well formed" and "Valid".

- IEC 61499 has 3 DTD file :

  1. Data Type allowed in IEC 61499

  2. FBMGT – Function Block Management Commands, defines protocol used for communication.

  3. Library Elements



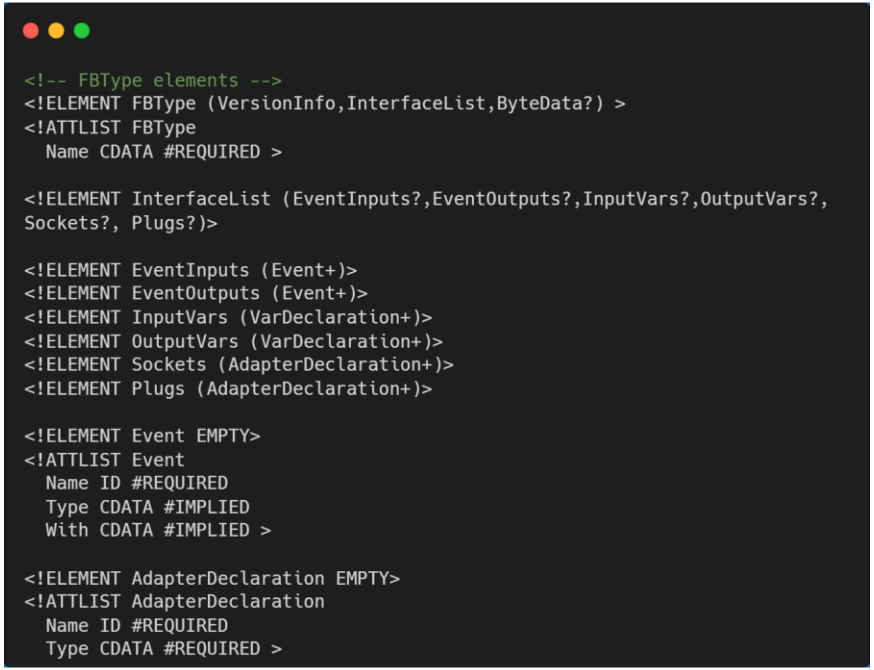
Mapping the target DTD file into Ontology Knowledge Base

- Each DTD document : as a domain in the ontology
- Each DTD element : mapped to an OWL class
- The hierarchies of the DTD elements : mapped to object properties
- The attributes of an element : mapped to data properties
- Translate the occurrence of an element symbol :

DTD Document	OWL Code
(*) declaring 0 or more occurance of an Element	Owl:minQualifiedCardinality = 0;
(+) declaring minimum 1 occurance of Element	Owl:minQualifiedCardinality = 1;
(?) declaring 0 or 1 occurance of an element	Owl:maxQualifiedCardinality = 1;

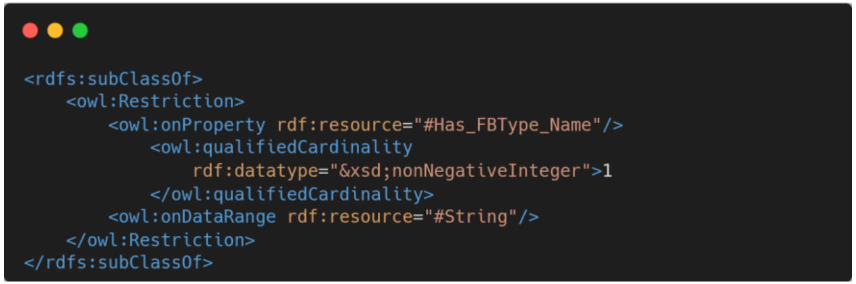
Mapping DTD file for metamodel definition into OWL Ontology Knowledge

- Snapshoot IEC 61499 DTD code



```
<!-- FBType elements -->
<!ELEMENT FBType (VersionInfo,InterfaceList,ByteData?) >
<!ATTLIST FBType
  Name CDATA #REQUIRED >

<!ELEMENT InterfaceList (EventInputs?,EventOutputs?,InputVars?,OutputVars?,
Sockets?, Plugs?)>

<!ELEMENT EventInputs (Event+)>
<!ELEMENT EventOutputs (Event+)>
<!ELEMENT InputVars (VarDeclaration+)>
<!ELEMENT OutputVars (VarDeclaration+)>
<!ELEMENT Sockets (AdapterDeclaration+)>
<!ELEMENT Plugs (AdapterDeclaration+)>

<!ELEMENT Event EMPTY>
<!ATTLIST Event
  Name ID #REQUIRED
  Type CDATA #IMPLIED
  With CDATA #IMPLIED >

<!ELEMENT AdapterDeclaration EMPTY>
<!ATTLIST AdapterDeclaration
  Name ID #REQUIRED
  Type CDATA #REQUIRED >
```

Example creating OWL format :
FBType must has a name of data type String.

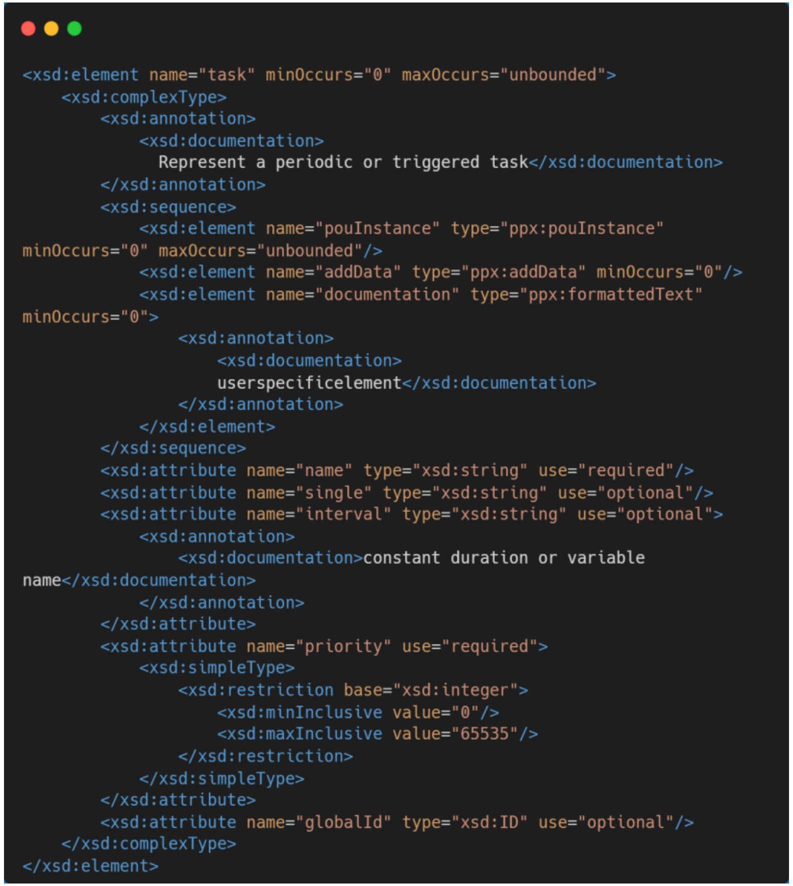


```
<rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty rdf:resource="#Has_FBType_Name"/>
            <owl:qualifiedCardinality
                rdf:datatype="&xsd;nonNegativeInteger">1
            </owl:qualifiedCardinality>
        <owl:onDataRange rdf:resource="#String"/>
    </owl:Restriction>
</rdfs:subClassOf>
```

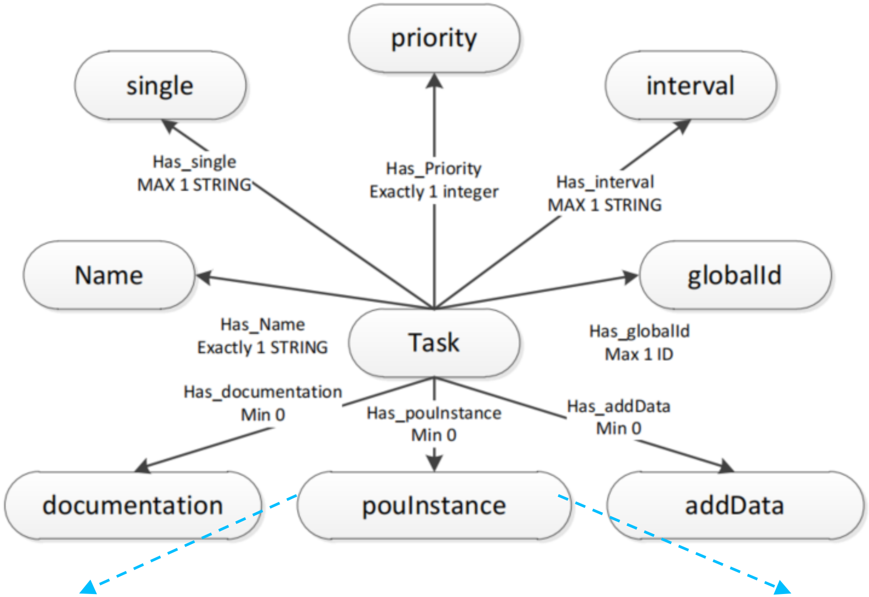
Note :
Repeat for each element in 1 DTD file.

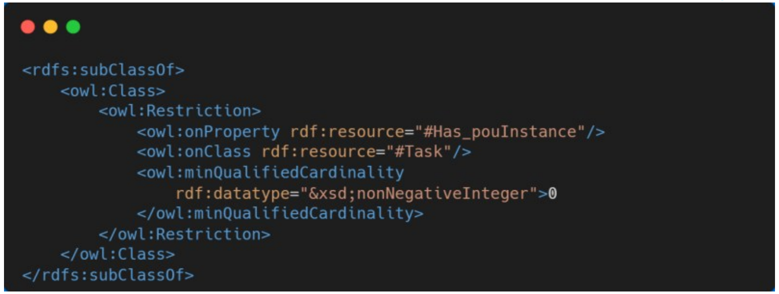
Mapping XSD file for metamodel definition into OWL Ontology Knowledge

Example creating OWL Knowledge Base of IEC 61131 "Task"



```
<xsd:element name="task" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
        <xsd:annotation>
            <xsd:documentation>
              Represent a periodic or triggered task</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="pouInstance" type="ppx:pouInstance"
minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="addData" type="ppx:addData" minOccurs="0"/>
            <xsd:element name="documentation" type="ppx:formattedText"
minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>
                    userspecificelement</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="single" type="xsd:string" use="optional"/>
        <xsd:attribute name="interval" type="xsd:string" use="optional">
            <xsd:annotation>
                <xsd:documentation>constant duration or variable
name</xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="priority" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:integer">
                    <xsd:minInclusive value="0"/>
                    <xsd:maxInclusive value="65535"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="globalId" type="xsd:ID" use="optional"/>
    </xsd:complexType>
</xsd:element>
```

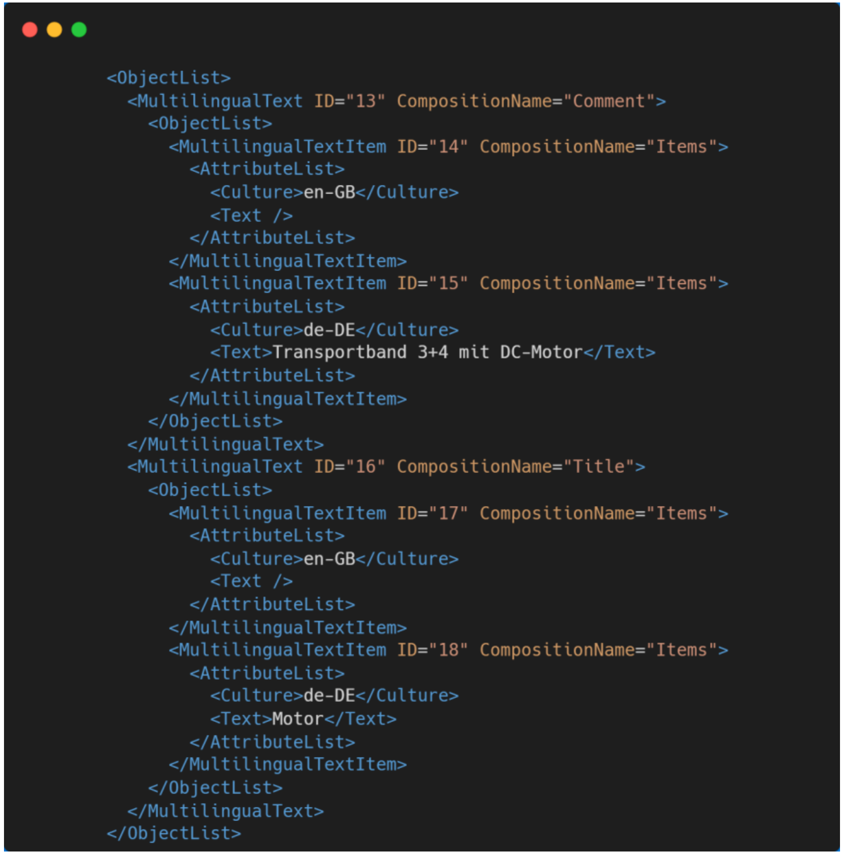




```
<rdfs:subClassOf>
    <owl:Class>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#Has_pouInstance"/>
            <owl:onClass rdf:resource="#Task"/>
            <owl:minQualifiedCardinality
                rdf:datatype="&xsd;nonNegativeInteger">0
            </owl:minQualifiedCardinality>
        </owl:Restriction>
    </owl:Class>
</rdfs:subClassOf>
```

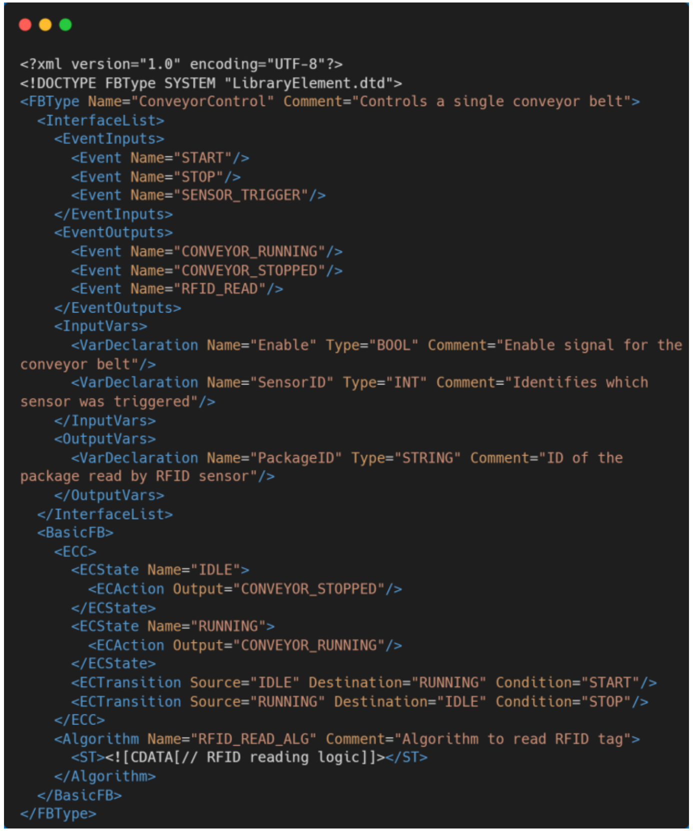
IEC 61131 file → into IEC 61499

Proof-of-Concept



```
<ObjectList>
  <MultilingualText ID="13" CompositionName="Comment">
    <ObjectList>
      <MultilingualTextItem ID="14" CompositionName="Items">
        <AttributeList>
          <Culture>en-GB</Culture>
          <Text />
        </AttributeList>
      </MultilingualTextItem>
      <MultilingualTextItem ID="15" CompositionName="Items">
        <AttributeList>
          <Culture>de-DE</Culture>
          <Text>Transportband 3+4 mit DC-Motor</Text>
        </AttributeList>
      </MultilingualTextItem>
    </ObjectList>
  </MultilingualText>
  <MultilingualText ID="16" CompositionName="Title">
    <ObjectList>
      <MultilingualTextItem ID="17" CompositionName="Items">
        <AttributeList>
          <Culture>en-GB</Culture>
          <Text />
        </AttributeList>
      </MultilingualTextItem>
      <MultilingualTextItem ID="18" CompositionName="Items">
        <AttributeList>
          <Culture>de-DE</Culture>
          <Text>Motor</Text>
        </AttributeList>
      </MultilingualTextItem>
    </ObjectList>
  </MultilingualText>
</ObjectList>
```

Translation



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE FBType SYSTEM "LibraryElement.dtd">
<FBType Name="ConveyorControl" Comment="Controls a single conveyor belt">
  <InterfaceList>
    <EventInputs>
      <Event Name="START"/>
      <Event Name="STOP"/>
      <Event Name="SENSOR_TRIGGER"/>
    </EventInputs>
    <EventOutputs>
      <Event Name="CONVEYOR_RUNNING"/>
      <Event Name="CONVEYOR_STOPPED"/>
      <Event Name="RFID_READ"/>
    </EventOutputs>
    <InputVars>
      <VarDeclaration Name="Enable" Type="BOOL" Comment="Enable signal for the
conveyor belt"/>
      <VarDeclaration Name="SensorID" Type="INT" Comment="Identifies which
sensor was triggered"/>
    </InputVars>
    <OutputVars>
      <VarDeclaration Name="PackageID" Type="STRING" Comment="ID of the
package read by RFID sensor"/>
    </OutputVars>
  </InterfaceList>
  <BasicFB>
    <ECC>
      <ECState Name="IDLE">
        <ECAction Output="CONVEYOR_STOPPED"/>
      </ECState>
      <ECState Name="RUNNING">
        <ECAction Output="CONVEYOR_RUNNING"/>
      </ECState>
      <ECTransition Source="IDLE" Destination="RUNNING" Condition="START"/>
      <ECTransition Source="RUNNING" Destination="IDLE" Condition="STOP"/>
    </ECC>
    <Algorithm Name="RFID_READ_ALG" Comment="Algorithm to read RFID tag">
      <ST><![CDATA[// RFID reading logic]]></ST>
    </Algorithm>
  </BasicFB>
</FBType>
```

Result and Discussion

Result and Discussion

- The migration from IEC 61131 PLC code to IEC 61499 is achieved using ontology mapping.
- IEC 61499 is the system model of distributed automation, it need a Run Time Environment to be pre-installed and run before deployment of IEC 61499 code.

Future Reasearch

- Deployment of IEC 61499 code.
- Comprehensive testing
- IT/OT application based on IEC 61499 Code for better code utilization and modularity

University of Stuttgart
Institut of Industrial Automation
and Software Engineering

Thank you!

Beryl Wicaksono

e-mail	beryl.wicaksono@gmail.com
phone	+49 (0) 711 685-
fax	+49 (0) 711 685-

University of Stuttgart
Institute of Industrial Automation
and Software Engineering

References

[1] https://blog.orientalmotor.com/what-is-modular-automation

[2] Process INDUSTRIE 4.0: The Age of Modular Production On the doorstep to market launch, ZVEI –Zentralverband Elektrotechnik und Elektronikindustrie e. V. March 2019

[3] P. Gsellmann, M. Melik-Merkumians, A. Zoitl and G. Schitter, "A Novel Approach for Integrating IEC 61131-3 Engineering and Execution Into IEC 61499," in IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5411-5418, Aug. 2021, doi: 10.1109/TII.2020.3033330.

[4] C. Sünder, M. Wenger, C. Hanni, I. Gosetti, H. Steininger and J. Fritsche, "Transformation of existing IEC 61131-3 automation projects into control logic according to IEC 61499", Proc. 13th IEEE Int. Conf. Emerg. Technol. Factory Autom., pp. 369-376, Sep. 2008.

[5] C. Sünder, A. Zoitll, J. H. Christensen, H. Steininger and J. Fritsche, "Considering IEC 61131-3 and IEC 61499 in the context of component frameworks", *Proc. 6th IEEE Int. Conf. Ind. Informat.*, pp. 277-282, Jul. 2008.

[6] M. Wenger, A. Zoitl, C. Sünder and H. Steininger, "Transformation of IEC 61131-3 to IEC 61499 based on a model driven development approach", *Proc. 7th IEEE Int. Conf. Ind. Informat.*, pp. 715-720, Jun. 2009.

[7] W. Dai and V. Vyatkin, "Redesign distributed PLC control systems using IEC 61499 function blocks", *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 390-401, Apr. 2012.

References

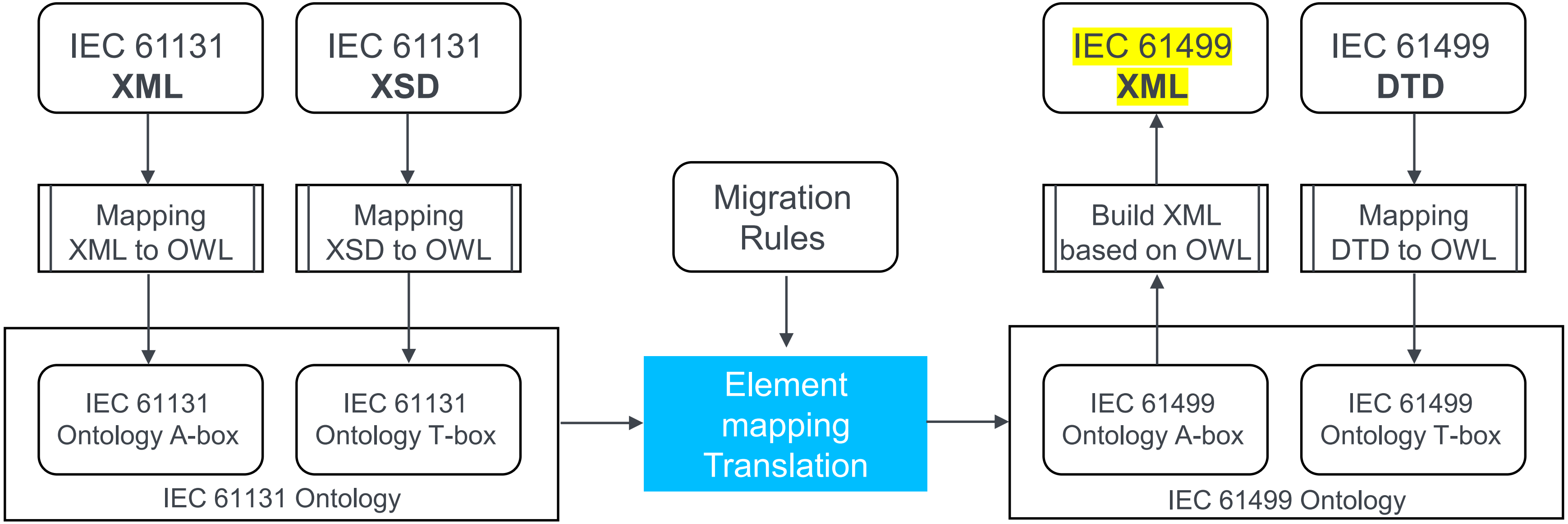
[8] W. Dai and V. Vyatkin, "Transformation from PLC to distributed control using ontology mapping", *Proc. 10th IEEE Int. Conf. Ind. Informat.*, pp. 436-441, Sep. 2012.

[9] W. Dai, V. Vyatkin and V. Dubinin, "Ontology-based design recovery and migration between IEC 61499-compliant tools", *Proc. 37th Annu. Conf. the IEEE Ind. Electron. Soc.*, pp. 4332-4337, Nov. 2011.

[10] W. Dai, V. N. Dubinin and V. Vyatkin, "Migration From PLC to IEC 61499 Using Semantic Web Technologies," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44, no. 3, pp. 277-291, March 2014, doi: 10.1109/TSMCC.2013.2264671

Migration to IEC 61499 Strategy

Concept of Ontology Mapping



- T-Box (Taxonomy Box) : The knowledge base of all properties standard and its relationships between concepts.
- A-Box (Assertion Box) : The instances (system configuration, function block, code) are modelled. It consists of knowledge which is specific to the individual system design.