Entwining Plant Engineering Data and ERP Information: Vertical Integration with AutomationML and ISA-95

Bernhard Wally, Christian Huemer and Alexandra Mazak

Business Informatics Group, Institute of Software Technology and Interactive Systems, TU Wien

Favoritenstraße 9-11/188, 1040 Vienna, Austria

{wally, huemer, mazak}@big.tuwien.ac.at

***Abstract*—IT systems' integration in manufacturing companies is currently investigated in both academia and industry. While there can be found specialized systems and standards that tackle specific, e.g., production relevant problems, little has been done in the alignment of and transformation between such industrial standards. We will present the alignment of two specialized international standards, which will foster vertical system integration through detailed mapping of related concepts: (i) the Automation Markup Language (AML) standardizes the modeling of factory shop floors on top of the XML-based Computer Aided Engineering Exchange (CAEX) data format and (ii) ISA-95 is a series of standards targeting the integration of enterprise control systems, most prominent enterprise resource planning systems and manufacturing execution systems. In order to provide higher level semantics to lower level system descriptions, we have (i) aligned elements from AML and ISA-95 in order to make explicit both overlaps and complementary concepts and (ii) defined a ruleset for referencing external ISA-95 documents/elements from AML documents. Finally, we have developed a scenario that shows the potential use case for such an entwined use of AML and ISA-95.**

Keywords-AutomationML; ISA-95; Metamodel Alignment

I. Introduction

Manufacturing companies usually run a number of different software systems to program and control their operations from business aspects to their production systems. A common method to describe the different levels of detail is a functional hierarchy model [1] (cf. Fig. 1). Usually, each of these levels is configured and run by domain specific software, e.g., (i) process control is realized by programmable logic controllers (PLCs) that are programmed through dedicated tools that are used during design time, but not during runtime, (ii) production control systems are implemented as human-machine-interfaces, automated control systems, etc., (iii) manufacturing operations management is supported by a manufacturing execution system (MES), and (iv) business planning and logistics is usually run by a kind of enterprise resource planning (ERP) system [1]. Each of these systems might be designed by a different vendor and cater specific industries, brands, products, etc., which leads to technical barriers in the integration of such industrial systems.

In order to overcome some of these barriers, international standards have been developed—e.g., under the umbrella of the International Electrotechnical Commission (IEC) or by industrial associations—that provide semantics (i) for within specific system levels (e.g., IEC 62714), (ii) for information exchange between system levels (e.g., IEC 62264), (iii) for the description of information within a certain industry (e.g., Office Furniture Modeling Language (OFML) [2]), or (iv) for the communication protocol between systems of different vendors (e.g., IEC 62541 [3]).

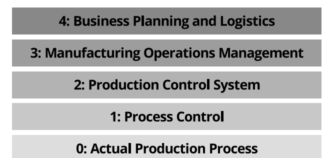

Figure 1. Functional hierarchy model for a manufacturing enterprise, depicting the five hierarchy levels (Level 0 – Level 4) in industrial automation [1].

In this work we will provide an alignment of two partly overlapping international sets of standards: IEC 62714 and IEC 62264. While the former provides a data exchange format and modeling framework for the shop floor, and thus targets levels 1 and 2 of the functional hierarchy model, the latter defines a data model for objects on levels 3 and 4 and for data exchange between these levels. With the alignment of these two standards, it becomes easier to migrate data between software tools of different levels in an automated, semantics-preserving manner. We will provide an application scenario that shows a possible use case of the alignment of these two standards, based on the running example given in [4].

While IEC 62264 (also known as ISA-95) is an established standard, IEC 62714 is an upcoming new standard that acts as an integration instrument for various engineering tools in the workflow of the design and ramp-up of manufacturing systems. It is developed and promoted by a consortium of notable international companies and academic institutions.

II. Background

A. *IEC 62424 and IEC 62714*

Computer Aided Engineering Exchange (CAEX) is a data format that has been defined in the scope of IEC 62424:2008 and provides structures (i) for information exchange between Piping and Instrumentation Diagram (P&ID) tools and Process Control Engineering (PCE) related

Computer Aided Engineering (CAE) tools, as well as (ii) for the representation of PCE requests in P&I diagrams [5]. CAEX is based on XML and enables the metamodeling and modeling of e.g., the hierarchical architecture of a plant, including involved machines and controllers and their physical and logical connections.

For the purpose of this paper, the following six main concepts of CAEX are of relevance:

Role Classes are used to define a hierarchy of concepts that are relevant for the given use case, where each role class may declare (and define) attributes. The hierarchical structure (i.e., the parent-child relationship of role classes) has no semantics, but has organizational character only.

Interface Classes allow the definition of interfaces (i) that can be pairwise linked within CAEX by internal links, or (ii) that can reference external data through syntax and semantics described in separate documents (e.g., geometry and kinematics [6], or logic descriptions [7]). Like role classes, also interface classes are modeled hierarchically, where the hierarchy is of organizational character only. Interface classes may declare (and define) attributes.

Internal Elements are the main concept for the definition of the actual model, which in CAEX is typically the equipment hierarchy ("instance hierarchy") of a plant's shop floor. Internal elements may refer to role classes or system unit classes that they instantiate, they can hold instances of external interfaces that they provide, and they also define values for corresponding attribute declarations.

System Unit Classes are an intermediate layer between role classes and internal elements—they are used to store reusable (e.g., vendor specific) AML classes, such as a specific robot along with its attributes, internal elements and external interfaces. They can be seen as "blueprints" for internal elements; however, their use is not mandatory: internal elements can also be direct instances of role classes.

External Interfaces can be defined as children of internal elements and system unit classes—they represent instantiations of interface classes. External interfaces can be used to interlink different parts of an instance hierarchy as instance-instance relations using "internal link" objects.

Attributes resemble properties of role classes, interface classes, and internal elements. Usually, attributes of role classes and interface classes declare the attribute, i.e. they define the property type and possibly a value range and/or default value. Attributes of internal elements define values for properties—it is not necessary that an attribute has been declared by a role class or interface class, instead internal elements may define their own attributes just as required.

IEC 62714 is based on CAEX and defines sets of role classes and interface classes with certain restrictions regarding their application [4], [8]—it is also known as Automation Markup Language (AutomationML, AML), which is the term we will use in the remainder of this paper. AML defines an abstract interface class `ExternalDataConnector` which is to be used to reference external documents and elements therein. Two use

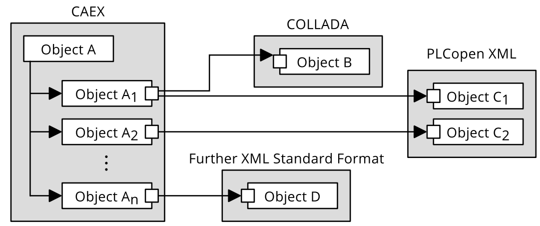

Figure 2. Referencing external data from AML documents (from [4]).

cases of this external data connector have been defined so far in separate whitepapers (cf. Fig. 2): `COLLADAInterface` specifies how external COLLADA [1] documents are referenced [6] and `PLCopenXMLInterface` defines how PLCopen[2] XML documents can be referenced from AML documents [7]. It is these whitepapers that provide a rough guideline on the referencing and integration of external data into AML documents and that serve as a starting point for the work presented here.

B. IEC 62264 and B2MML

IEC 62264 is a series of standards that addresses the integration of the enterprise domain with the manufacturing and control domains—which information is normally exchanged and how can this be accomplished in a robust, secure, and cost-effective manner. It provides a standard terminology and set of concepts for system integration, based upon a standard that has been standardized by the American National Standards Institute (ANSI) and the International Society of Automation (ISA): ANSI/ISA-95 [1]. The relevant part of IEC 62264 for this work is part 2, as it is specified in IEC 62264-2:2013 [10]. For its better readability, we will stick to the term ISA-95 for the remainder of this paper, when we are referring to concepts of IEC 62264. Specifically, our work is based on the latest edition of this standard: IEC 62264-2:2013.

Part 2 of ISA-95 specifies common objects and attributes, mainly by a set of commented UML class diagrams, that can be roughly differentiated between (i) basic resources that depict the static definitions of an enterprise with regards to its production facilities (e.g., personnel, equipment, and material) and (ii) operations management information that resembles operational data (e.g., operations capabilities, schedules, and performance). The former five basic resources are briefly described below, as they are the main objects involved in the alignment of AML and ISA-95:

Personnel comprises actors for the operation of manufacturing processes—both, classes of personnel and individuals. AML has not been designed with a focus on the definition of human resources, ISA-95 personnel information can therefore be seen as complementary information to AML. However if required, this kind of information could be modeled in AML as well.

[1] COLLADA—Collaborative Design Activity: an XML based exchange format for 3D assets (cf. https://www.khronos.org/collada/).

[2] PLCopen is a vendor- and product-independent association active in industrial control (cf. http://www.plcopen.org/). PLCopen XML is a data exchange format for the storage of PLC program information according to IEC 61131-3 [9].

Equipment represents the equipment of an organization in form of a role based model. One example is the organizational structure with regards to physical, geographical or logical features, such as the enterprise and its sites, areas, work centers, and work units. More specific examples include welders, titration testers, lift trucks, reactors, etc. Equipment may define a hierarchy, and it resembles an abstract model that is instantiated through physical assets.

Physical Asset represents the physical pieces of equipment, i.e. while equipment defines the roles for certain items, a physical asset is the real physical item that implements that role. Physical assets have an impact on the ERP layer, as they are usually of value and need to be tracked financially. Machines for instance have a unique serial number—as such the serial number of a machine resembles a valid value for a physical asset's ID. A dedicated equipment asset mapping represents the relationship between a physical asset and an equipment. It records the time frame in which a physical asset was associated with an equipment.

Material represents raw, finished, and intermediate materials, as well as consumables. A uniquely identified amount of material is referred to as material lot.

Process Segment resembles the smallest elements of manufacturing activities that are visible to business processes. Process segments describe a hierarchical model in which multiple levels of abstraction may be defined. Process segments are also a logical grouping of personnel, equipment, physical asset, and material required for a specific manufacturing operation.

The Business to Manufacturing Markup Language (B2MML) is an XML serialization of ISA-95 created by the Manufacturing Enterprise Solutions Association (MESA) International. The latest version of this standard is V0600 which has been specified in [11]—this version is the basis for our contribution. V0600 is up to date with the latest versions of the ISA-95 standard, which in turn is the basis for the latest version of IEC 62264.

III. Related Work

A. Alignment and Integration of Semantic Models with AutomationML

AML has been successfully aligned with other standards in the past: COLLADA and PLCopen XML. Support for COLLADA enables AML documents to provide concise geometric and kinematic information through referencing respective external COLLADA documents that can be authored by professional tools and used for e.g., visualization or simulation of plants [6].

Integration of PLCopen XML elements enables AML documents to get a hold on behavioral information, such as logic programs alongside with detailed information about data points and their usage across machines. With that information, it is possible to provide both a structural (CAEX) and a behavioral (PLCopen XML) view of the plant from an integrated set of sources [7].

AML has also been aligned with IEC 62541—also known as OPC Unified Architecture (UA)—in order to enable the communication of AML elements throughout industrial networks: (i) the CAEX metamodel has been mapped to the OPC UA specification and has been specified as an OPC UA information model and (ii) the standardized base libraries of AML have been defined as OPC UA information model that is to be used as a starting point for AML conforming OPC UA configurations [12], [13], [14].

An application recommendation has been compiled that describes a workflow and method of hardware configuration modeling using AutomationML as an exchange format between Electronic Computer-Aided Design (ECAD) and PLC tools [15].

B. Alignment and Integration of Semantic Models in the Industrial Engineering Domain

An OPC UA information model has been created for IEC 61131-3 architecture models, which makes controllers as the main component of automation systems accessible in vertical information integration [16].

Another OPC UA information model has been defined for concepts of ISA-95. This mapping of ISA-95 components to the OPC UA protocol increases the vertical reach of ISA-95 in industrial settings [17].

A semantic representation of the administrative shell of smart production systems has been described in [18]. An administrative shell is the digital representation of any kind of information or service, be it physical or virtual, a concept that has been presented e.g., in [19].

ISA-95 has been aligned with a modeling language targeting level 4 activities, the Resource-Event-Agent (REA) ontology (standardized in IEC 15944-4) [20]. This work enables integration of information from an MES into an ERP (as REA has been successfully used for the implementation of a prototypical ERP system [21]).

IV. Alignment of ISA-95 and AutomationML

The alignment of AML and ISA-95 is realized (i) on a metamodel level between ISA-95 and CAEX and (ii) between ISA-95 metamodel classes and specific AML elements. Furthermore, a mechanism is described for explicitly referencing ISA-95 elements that are stored within B2MML documents from AML documents and thus allowing the parallel use of AML and B2MML.

A. Mapping of ISA-95 Elements to the CAEX Metamodel

The relevant part of the CAEX metamodel corresponds to the elements described in Sec. II.A: Role Class, Interface Class, Internal Element, External Interface, System Unit Class, and Attribute. The elements of ISA-95 need to be mapped to these metamodel elements in a corresponding way. As the number of metamodel items from ISA-95 is too large to be presented here in full length, the equipment model is taken as an example for the complete mapping that has been realized. Fig. 3 depicts the equipment model, which resembles information about classes of equipment, specific equipment, and equipment capability tests. Equipment classes and specific equipment may define properties [1].

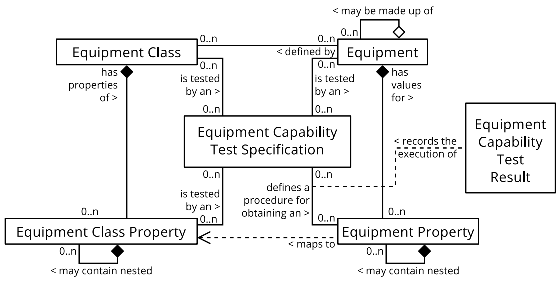

Figure 3. ISA-95 equipment model (from [1]).

The mapping of instances of the ISA-95 equipment model to CAEX model elements is discussed below and visualized on the left side of Fig. 4.

Equipment Classes are a representation of a grouping of equipment with similar characteristics. An equipment class may represent one of multiple definitions for an equipment.

This corresponds to the way role classes are specified in CAEX, however role classes are limited in their functionality, as they can only specify one parent role class. Through that, they cannot refer to both the AML role class "Resource" (sematic match) and to the role class "B2MMLData" (see Sec. IV.D).

Accordingly, an ISA-95 equipment class corresponds to a CAEX *system unit class*, for the following reasons: (i) system unit classes are external data aware, i.e. they provide the structural features required to reference externally stored data, and (ii) semantically, system unit classes are a good fit for a "classification" element.

On the negative side, only one system unit class can be referenced by an internal element. In order to provide the multi-typing feature that is provided by ISA-95, we propose the definition of sub internal elements within a multi-typed internal element, where each of the sub internal elements refers to one of the system unit classes the parent internal element is defined by.

Equipment is the representation of the elements of the equipment hierarchy model. It shall be defined by zero to multiple equipment classes.

Equipment is semantically best mapped to CAEX *internal elements*: ISA-95 equipment defines the elements of the equipment hierarchy, including production lines and process cells—this is exactly what the instance hierarchy in CAEX does: it defines the actual plant model by (possibly nested) internal elements.

Equipment Class Properties and **Equipment Properties** correspond to CAEX *attributes*.

Equipment Capability Test Specifications are representations of capability tests—they may be associated with equipment classes, equipment class properties, equipment or equipment properties. It is therefore required that test specifications can reference system unit classes and internal elements. From a modeling point of view, a CAEX *internal element* is therefore the corresponding element.

Equipment Capability Test Results are the results from an equipment capability test for a specific piece of equipment. From a modeling point of view, they are very specific, as they are defined by ISA-95 by an association class that associates the test specification with an equipment property. On an instance level, the ISA-95 definition of an equipment capability test result specifies that it is the test result for a specific piece of equipment [10]. Semantically it makes sense to assign a test result for a specific equipment to the equipment. We should keep in mind that the association class associates an equipment property—consequently, the corresponding element in CAEX for an ISA-95 test result is a complex *attribute*, modeled as a subattribute of the corresponding equipment attribute under test. Furthermore, this implies that there is no need for a corresponding equipment capability test result role class.

B. *Alignment of ISA-95 Elements and AML Base Role Classes*

AML already defines a set of role classes that are related to ISA-95 concepts, spread over a few AML role class libraries. Most notable, the AutomationML base role class library implements the Process-Product-Resource (PPR) concept [22], [23] that can be used to tag an internal element as a process, a product, or a resource. Through instances of the `PPRConnector` interface class it is possible to interlink instances and clarify to some extent the production relevant relation of AML objects. In ISA-95 this concept is specified in much more detail but in a similar manner—an alignment of the respective concepts is therefore reasonable. The PPR concepts in AML are defined as follows [8]:

Process and **Process Structure** are production related processes, or process oriented object hierarchies.

Product and **Product Structure** are products, parts or materials, that are processed, or product oriented object hierarchies.

Resource and **Resource Structure** describes plants, equipment, or other production resources, or resource oriented object hierarchies.

From that, it can be inferred, that ISA-95 classes that allow nested elements relate to `*Structure` AML elements, while the other classes refer to non-structural elements. The discussed ISA-95 elements are aligned below, the alignment is further depicted in Fig. 4. What follows is not a complete alignment of all ISA-95 elements, but only a meaningful excerpt, due to the restricted space in this paper. ISA-95 elements of the operations management models follow the same alignment approach as the elements of the process segment model.

Personnel Class and **Person** have no specific corresponding targets in existing AML role class libraries and are therefore subroles of `AutomationMLBaseRole`.

Equipment Class corresponds semantically to the AML role class Resource. Since equipment classes do not allow nested elements, `Resource` is already the best match.

Equipment corresponds semantically to the AML role class `Resource`. Since equipment can contain nested equipment, it is ultimately aligned with the role class `ResourceStructure`.

Physical Asset Class is a representation of a grouping of physical assets with similar characteristics. It corresponds semantically to the AML role class `Resource`. Since physical asset classes do not allow nested elements, `Resource` is already the best match.

Physical Asset is a physical piece of equipment. It corresponds semantically to the AML role class `Resource`. Since a physical asset can contain nested physical assets, it is aligned with the role class `ResourceStructure`.

Material Class is a representation of groupings of material definitions for a definite purpose. It corresponds semantically to the AML role class `Product`. Since a material class can contain nested material classes, it is ultimately aligned with the role class `ProductStructure`.

Material Definition is a representation of goods with similar name characteristics for the purpose of manufacturing definition, scheduling, capability and performance. It corresponds semantically to the AML role class `Product`. Since a material definition can contain nested material definitions, it is ultimately aligned with the role class `ProductStructure`.

Material Lot is a uniquely identified specific amount of material, either countable or weighable. It corresponds semantically to the AML role class `Product`. Since a material lot can contain nested material lots, it is ultimately aligned with the role class `ProductStructure`.

Material Sublot is a separately identifiable quantity of the same material lot. It corresponds semantically to the AML role class `Product`. Since a material sublot can contain nested material sublots, it is ultimately aligned with the role class `ProductStructure`.

Process Segment is the listing of resources of personnel, equipment, physical assets, and material needed for a given process. It corresponds semantically to the AML role class `Process`. Since a process segment can contain nested process segments, it is ultimately aligned with the role class `ProcessStructure`.

Equipment Segment Specification specifies the equipment resources required for a process segment. It links to equipment or equipment classes and might provide additional information, such as the quantity needed and intended use for the equipment. It does however not resemble an equipment itself. As such, it is not aligned with AML role class `Resource`, but with another AML role class: `Group`. An AML group is a specific role class that is allowed to define its own properties (such as in this case the quantity of equipment required), but it is allowed to only hold pointers to (in AML: mirror objects of) other internal elements as children. In this specific case, only internal elements of role class equipment or equipment class are allowed to be referenced.

This provision (when generalized) applies to physical asset segment specification and material segment specification (and to corresponding elements of the operations management metamodels) as well.

C. *Creating Meaningful AML Role Classes*

For each suitable ISA-95 metamodel element (i.e., one that is not mapped to an AML attribute) we define an AML role class, that is used by other AML elements to identify them as e.g., equipment class or equipment capability test specification. Since (i) role classes support inheritance and (ii) valid AML role classes must directly or indirectly inherit from `AutomationMLBaseRole` we take the following approach: ISA-95 elements that correspond to an existing AML role class are defined as a subrole of that role class, while ISA-95 elements that have no AML counterpart, such as "person", inherit from the AML base role (cf. Sec. IV.B). With this approach, it is possible to apply the AML PPR pattern directly to internal elements that instantiate these role classes, if the relevant classes inherit from AML PPR role classes. Refer to the right side of Fig. 4 for a visualization of the discussed alignments.

D. *Referencing B2MML Documents from AML*

AML provides a standardized mechanism to reference data stored in external files, such as COLLADA or PLCopen XML. A derived mechanism has been developed in order to reference arbitrary kinds of files, collected in a "best practice recommendation" (BPR) for external data reference by the AutomationML consortium [24]. The definition of an external data reference from AML documents to B2MML documents, as explained below, conforms to this recommendation and is an early instantiation of this BPR.

Two elements need to be defined: (i) role class `B2MMLData` (a subrole class of [24]'s `ExternalData`) and (ii) interface class `B2MMLReference` (a subinterface class of [24]'s `ExternalDataReference`). B2MMLReference inherits two attributes of type "xs:string" (cf. https://www.w3.org/TR/xmlschema11-2/#string): `refURI` stores the Uniform Resource Identifier (URI, cf. https://tools.ietf.org/html/rfc3986) of the B2MML file to be referenced, and `MIMEtype` holds the corresponding MIME type information, which is in this case fixed to a default value of "application/x.b2mml+xml", following the recommendations given in [25].

Using this mechanism, it is possible to map AutomationML elements to ISA-95 elements stored externally. Applications that are able to interpret both types of documents, may generate a more complete view (e.g., by adding information found in a B2MML document to an AML document) on a production system than is stored within the AML or B2MML documents alone. The basic idea is that an AML internal element or system unit class may correspond to a specific ISA-95 element in a B2MML document. The external data reference enables making such a mapping explicit.

The concrete procedure for referencing ISA-95 elements stored in external B2MML files from AML documents is as follows:

- An internal element or system unit class that corresponds to an ISA-95 element stored in an external B2MML document must implement a reference to role class `B2MMLData`.

- In order to reference a specific B2MML file, an external interface of type `B2MMLInterface` is added to that internal element. This external interface defines in the `refURI` attribute the URI'fied path to the external file.
- Referencing elements within a B2MML document is modeled by applying the global ID of the respective element to the fragment part of the URI (separated by a hash "#"). If no fragment part is defined in the URI, then the root element is referenced. B2MML does not support the common XML notation of an `id` attribute, but instead uses a sub-element named `ID` of type "xs:normalizedString" (cf. https://www.w3.org/TR/xmlschema11-2/#normalizedString). The content of that element represents the identifier for the higher level B2MML element to be referenced.

V. Application Scenario

The following application scenario (taken from [4], but reduced to a subset in order to keep it short and clear) shows how the integration of IEC 62714 and IEC 62264 can provide meaningful information for the processing of level 3 and 4 information on lower automation hierarchy levels. The scenario is depicted in Fig. 5 and reads as follows: in a car manufacturing plant, "cars-without-wheels" become "cars-with-wheels" in an assembly process realized by a robot that attaches wheels.

The scenario is realized in a (document-wise) distributed way: some of the information is modeled in AML (cf. Lst. 1), while additional information is modeled in ISA-95 and serialized in a separate B2MML file (cf. Lst. 2). This B2MML file is referenced from the AML file and allows an interpreter to generate a more complete AML file depicted in Lst. 3. Specifically, in Lst. 1 the process segment "Assemble" is not modeled completely in AML, but only a stub internal element is provided (cf. line 3)—the details are outsourced in a separate B2MML file. Line 8 holds the URI that references the B2MML file that further describes the corresponding process segment (for enhanced readability and reduced space requirements, we omit closing tags and use indent style for the visualization of XML documents in this paper—also we replace Universal Unique Identifiers (UUIDs) [26] with better readable string identifiers).

Lst. 2 provides ISA-95 information about the process segment "Assemble": it specifies one equipment segment ("Robot") and three material segments ("Car-without-Wheels", "Wheel", and "Car-with-Wheels") that can be traced back to elements in the AML file through string equality of the respective identifiers in the "ID" attributes. An interpreter can use the information stored in the B2MML file in order to enrich the AML file with information that has not been modeled there, yet. We show the enrichment of PPR information in Lst. 3: (i) AML elements that are (according to the B2MML file) interlinked are provided a newly generated UUID and a "PPRConnector" external interface and (ii) in the "Assemble" internal element, internal links are created accordingly in order to relate the process to

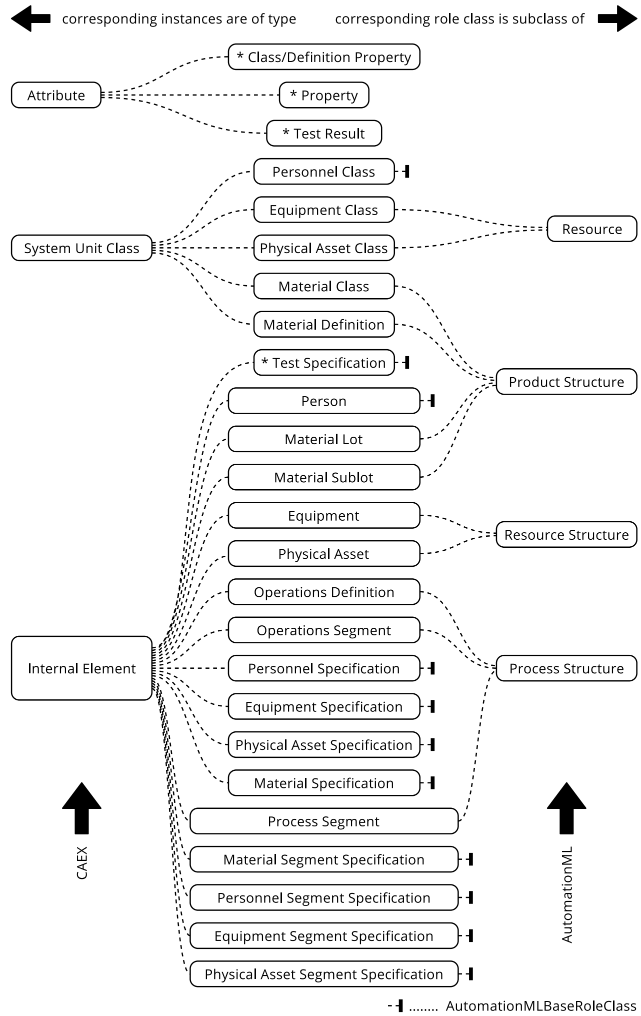

Figure 4. Mapping from ISA-95 (center) to elements of the CAEX metamodel (left) and to existing AutomationML elements (right).

its products ("Wheel", "Car-without-Wheels", and "Car-with-Wheels") and resource ("Robot").

In this case, an AML document has been enriched with data that has been defined in an ISA-95 context already. Conversely, it would be possible to generate ISA-95 information (e.g., a B2MML document) from compatibly structured AML information by extracting (i) PPR information and (ii) internal elements and system unit classes with references to relevant role classes.

VI. Conclusion

We have defined an alignment of two industrial standards that typically operate on different operational layers with regards to IT systems in the production industry. This alignment enables a conversion of information from one standard to the other, and it provides means for the modeling of higher system levels into lower level systems. We have also described a mechanism for referencing higher level information that is stored in external files from lower level documents. That way, high level systems can integrate information from both sources in order to infer information that has not been explicitly modeled, to propagate changes throughout the operational hierarchy (vertical integration).

We have evaluated our approach by an application scenario, were we have applied the alignment to concrete AML and B2MML documents. We have used the information stored in a separate B2MML document for information enrichment in the AML document, following the PPR approach. However, the alignment of AML and ISA-95 is only one step in our efforts to integrate systems of different hierarchy levels in the manufacturing domain. We plan to further align related standards in order to achieve tighter vertical integration, but also we want to consider standards that enable horizontal integration, e.g., by integrating the Resource-Event-Agent business ontology.

Listing 1. InstanceHierarchy and SystemUnitClassLib excerpts of document Example-B2MML.aml.

```
<InstanceHierarchy Name="ICCAR-InstanceHierarchy">
   <Version>1.0.0
   <InternalElement Name="ProcessSegments" ID="ProcessStructures">
      <InternalElement Name="Assemble" ID="Assemble">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Assemble
         <ExternalInterface Name="B2MMLReference" RefBaseClassPath="B2MMLReference" ID="B2MML-1">
            <Attribute Name="MIMEType" AttributeDataType="xs:string">
               <Value>application/x.b2mml+xml
            <Attribute Name="refURI" AttributeDataType="xs:anyURI">
               <Value>./Assemble.b2mml
         <SupportedRoleClass RefRoleClassPath="B2MMLData" />
         <RoleRequirements RefBaseRoleClassPath="ProcessSegment" />
   <InternalElement Name="Equipment" ID="Equipment">
      <InternalElement Name="Robot" ID="Robot">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Robot
         <Attribute Name="equipmentClasses">
            <RefSemantic CorrespondingAttributePath="ListType" />
            <Attribute Name="equipmentClass1" AttributeDataType="xs:string">
               <Value>Library/EquipmentClasses/RobotClass
         <RoleRequirements RefBaseRoleClassPath="Equipment" />
<SystemUnitClassLib Name="Lb">
   <Version>1.0.0
   <SystemUnitClass Name="MaterialDefs">
      <SupportedRoleClass RefRoleClassPath="ProductStructure" />
      <SystemUnitClass Name="Wheel">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Wheel
         <SupportedRoleClass RefRoleClassPath="MaterialDefinition" />
      <SystemUnitClass Name="Car-without-Wheels">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Car-without-Wheels
         <SupportedRoleClass RefRoleClassPath="MaterialDefinition" />
      <SystemUnitClass Name="Car-with-Wheels">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Car-with-Wheels
         <SupportedRoleClass RefRoleClassPath="MaterialDefinition" />
   <SystemUnitClass Name="EquipmentClasses">
      <SupportedRoleClass RefRoleClassPath="Resource" />
      <SystemUnitClass Name="RobotClass">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>RobotClass
         <SupportedRoleClass RefRoleClassPath="EquipmentClass" />
```

Listing 2. Content of document Assemble.b2mml.

```
<ProcessSegment>
<ID>Assemble
<Description>Assembling is accomplished by...
<OperationsType>Production
<HierarchyScope>
<EquipmentElementLevel>WorkCell
<EquipmentSegmentSpecification>
<EquipmentID>Robot
<MaterialSegmentSpecification>
<MaterialDefinitionID>Car-without-Wheels
<MaterialSegmentSpecification>
<MaterialDefinitionID>Wheel
<MaterialSegmentSpecification>
<MaterialDefinitionID>Car-with-Wheels
```

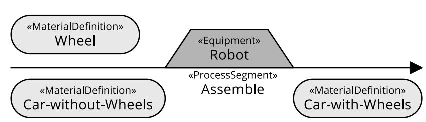

Figure 5. Schematic view of the application scenario (from [4], reduced). The dark gray trapezoid depicts a resource structure, light gray stadiums depict product structures, and the black arrow depicts a process structure.

References

[1] International Electrotechnical Commission (IEC), Enterprise-control system integration—Part 1: Models and terminology, International Standard, Rev. 2.0, May 2013, IEC 62264-1:2013.

Listing 3. InstanceHierarchy and SystemUnitClassLib excerpts of document Example-B2MML-Enriched.aml (inferred and newly generated information is highlighted in **bold**).

```
<InstanceHierarchy Name="ICCAR-InstanceHierarchy-Enriched">
   <Version>1.0.0
   <InternalElement Name="ProcessSegments" ID="ProcessSegments">
      <InternalElement Name="Assemble" ID="Assemble">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Assemble
         <ExternalInterface Name="B2MMLReference" RefBaseClassPath="B2MMLReference" ID="B2MML-1">
            <Attribute Name="MIMEType" AttributeDataType="xs:string">
               <Value>application/x.b2mml+xml
            <Attribute Name="refURI" AttributeDataType="xs:anyURI">
               <Value>./Assemble.b2mml
         <ExternalInterface Name="P" ID="PPR-1" RefBaseClassPath="PPRConnector" />
         <SupportedRoleClass RefRoleClassPath="B2MMLData" />
         <InternalLink Name="A1" RefPartnerSideA="Assemble:P" RefPartnerSideB="Wheel:P" />
         <InternalLink Name="A2" RefPartnerSideA="Assemble:P" RefPartnerSideB="Car-without-Wheels:P" />
         <InternalLink Name="A3" RefPartnerSideA="Assemble:P" RefPartnerSideB="Car-with-Wheels:P" />
         <InternalLink Name="A4" RefPartnerSideA="Assemble:P" RefPartnerSideB="Robot:P" />
         <SupportedRoleClass RefRoleClassPath="B2MMLData" />
         <RoleRequirements RefBaseRoleClassPath="ProcessSegment" />
   <InternalElement Name="Equipment" ID="Equipment">
      <InternalElement Name="Robot" ID="Robot">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Robot
         <Attribute Name="equipmentClasses">
            <RefSemantic CorrespondingAttributePath="ListType" />
            <Attribute Name="equipmentClass1" AttributeDataType="xs:string">
               <Value>Library/EquipmentClasses/RobotClass
         <ExternalInterface Name="P" ID="PPR-2" RefBaseClassPath="PPRConnector" />
         <RoleRequirements RefBaseRoleClassPath="Equipment" />
<SystemUnitClassLib Name="Library">
   <Version>1.0.0
   <SystemUnitClass Name="MaterialDefs">
      <SupportedRoleClass RefRoleClassPath="ProductStructure" />
      <SystemUnitClass Name="Wheel" ID="Wheel">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Wheel
         <ExternalInterface Name="P" ID="PPR-3" RefBaseClassPath="PPRConnector" />
         <SupportedRoleClass RefRoleClassPath="MaterialDefinition" />
      <SystemUnitClass Name="Car-without-Wheels" ID="Car-without-Wheels">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Car-without-Wheels
         <ExternalInterface Name="P" ID="PPR-4" RefBaseClassPath="PPRConnector" />
         <SupportedRoleClass RefRoleClassPath="MaterialDefinition" />
      <SystemUnitClass Name="Car-with-Wheels" ID="Car-with-Wheels">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>Car-with-Wheels
         <ExternalInterface Name="P" ID="PPR-5" RefBaseClassPath="PPRConnector" />
         <SupportedRoleClass RefRoleClassPath="MaterialDefinition" />
   <SystemUnitClass Name="EquipmentClasses">
      <SupportedRoleClass RefRoleClassPath="Resource" />
      <SystemUnitClass Name="RobotClass">
         <Attribute Name="ID" AttributeDataType="xs:string">
            <Value>RobotClass
         <SupportedRoleClass RefRoleClassPath="EquipmentClass" />
```


[2] Verband der Büro-, Sitz- und Objektmöbel e.V. (BSO), OFML—Standardized Data Description Format of the Office Furniture Industry, International Standard 2.0, Rev. 3, November 2015, OFML-2.0r3:2015.

[3] International Electrotechnical Commission (IEC), OPC Unified Architecture—Part 1: Overview and Concepts, International Standard, Rev. 1.0, February 2010, IEC 62541-1:2010.

[4] International Electrotechnical Commission (IEC), Engineering data exchange format for use in industrial automation systems engineering—Automation markup language—Part 1: Architecture and general requirements, International Standard, Rev. 1.0, June 2014, IEC 62714-1:2014.

[5] International Electrotechnical Commission (IEC), Representation of process control engineering—Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools, International Standard, Rev. 1.0, August 2008, IEC 62424:2008.

[6] AutomationML consortium, AutomationML Whitepaper Part 3—Geometry and Kinematics, Whitepaper, Rev. 2.0, August 2015, AML Part 3:2015.

[7] AutomationML consortium, AutomationML Whitepaper Part 4—AutomationML Logic Description, Whitepaper, Rev. 2.0, May 2010, AML Part 4:2010.

[8] International Electrotechnical Commission (IEC), Engineering data exchange format for use in industrial automation systems engineering—Automation markup language—Part 2: Role class libraries, International Standard, Rev. 1.0, March 2015, IEC 62714-2:2015.

[9] PLCopen Technical Committee 6, XML Formats for IEC 61131-3, Technical Paper, Rev. 2.01, May 2009, PLCopen XML:2009.

[10] International Electrotechnical Commission (IEC), Enterprise-control system integration—Part 2: Objects and attributes for enterprise-control system integration, International Standard, Rev. 2.0, June 2013, IEC 62264-2:2013.

[11] Manufacturing Enterprise Solutions Association (MESA) International, Business To Manufacturing Markup Language, Standards and Tools, Rev. V0600, May 2013.

[12] R. Henßen and M. Schleipen, "Interoperability between OPC UA and AutomationML," in Proceedings of the 8th International Conference on Digital Enterprise Technology (DET 2014), March 2014.

[13] AutomationML e.V. and OPC Foundation, OPC UA Information Model for AutomationML, Companion Specification, Rev. 1.00.00, February 2016, AML-OPC:2016.

[14] DIN, Combining OPC Unified Architecture and Automation Markup Language, DIN Standard, December 2016, DIN 16592:2016-12.

[15] AutomationML consortium, "Automated project configuration," Application Recommendation, April 2016.

[16] PLCopen and OPC Foundation, OPC UA Information Model for IEC 61131-3, Companion Specification, Rev. 1.00, March 2010, PLCopen-OPC:2010.

[17] OPC Foundation, OPC Unified Architecture for ISA-95 Common Object Model, Companion Specification, Rev. 1.00, October 2013, OPC-ISA-95:2013.

[18] I. Grangel-González, L. Halilaj, G. Coskun, S. Auer, D. Collarana, and M. Hoffmeister, "Towards a semantic administrative shell for industry 4.0 components," in Proceedings of the 10th IEEE International Conference on Semantic Computing (ICSC 2016), 2016.

[19] B. Dong, B. Wally, and A. Ferscha, "Tokenized interaction architecture," in Proceedings of the 2nd International Workshop on Pervasive Advertising, 2009.

[20] A. Mazak and C. Huemer, "From business functions to control functions: Transforming REA to ISA-95," in Proceedings of the 17th IEEE Conference on Business Informatics (CBI 2015), D. Aveiro, U. Frank, K. J. Lin, and J. Tribolet, Eds., vol. 1. IEEE, July 2015, pp. 33–42.

[21] B. Wally, A. Mazak, B. Kratzwald, and C. Huemer, "Model-driven retail information system based on REA business ontology and Retail-H," in Proceedings of the 17th IEEE Conference on Business Informatics (CBI 2015), D. Aveiro, U. Frank, K. J. Lin, and J. Tribolet, Eds., vol. 1, July 2015, pp. 116–124.

[22] A. F. Cutting-Decelle, R. I. M. Young, J. J. Michel, R. Grangel, J. L. Cardinal, and J. P. Bourey, "ISO 15531 MANDATE: A product-process-resource based approach for managing modularity in production management," vol. 15, no. 2, pp. 217–235, 2007.

[23] M. Schleipen and R. Drath, "Three-view-concept for modeling process or manufacturing plants with AutomationML," in Proceedings of the 14th International Conference on Emerging Technologies and Factory Automation, September 2009.

[24] AutomationML consortium, AutomationML Best Practice Recommendation—External Data Reference, Best Practice Recommendation, Rev. 1.0.0, July 2016, AML BPR-EDR:2016.

[25] N. Freed, J. Klensin, and T. Hansen, Media Type Specifications and Registration Procedures, Internet Engineering Task Force (IETF) Best Current Practice, January 2013, RFC6838. [Online]. Available: https://tools.ietf.org/html/rfc6838

[26] T. H. Dineen, P. J. Leach, N. W. Mishkin, J. N. Pato, and G. L. Wyant, "The network computing architecture and system: An environment for developing distributed applications," in Proceedings of the 1987 Summer USENIX Conference, 1987, pp. 385–398.