IEC 62264-2 for AutomationML

Bernhard Wally
Christian Doppler Laboratory for
Model-Integrated Smart Production,
TU Wien, 1040 Vienna, Austria
wally@big.tuwien.ac.at

Christian Huemer
Business Informatics Group,
Information Systems Engineering,
TU Wien, 1040 Vienna, Austria
huemer@big.tuwien.ac.at

Alexandra Mazak and Manuel Wimmer
Christian Doppler Laboratory for
Model-Integrated Smart Production,
TU Wien, 1040 Vienna, Austria
{mazak,wimmer}@big.tuwien.ac.at

***Abstract*—IEC 62264-2 and AutomationML can co-exist as separate views on the same production system; the former with a slight bias towards the upper levels, the latter slightly biased towards the lower levels of the automation hierarchy. Still, there is quite some semantic and structural overlap between IEC 62264-2 and AutomationML. Therefore, a semantic and structural alignment of their entities on a metamodel and a model level seems approrpate. In this work, we will present such an alignment together with two combinable methods for integration: (i) tagging AutomationML elements with IEC 62264-2 roles and (ii) referencing external IEC 62264-2 data.**

***Keywords*—AutomationML, IEC 62264, ISA-95, B2MML.**

I. INTRODUCTION

IEC 62264 (also known as ISA-95) part 2 is an international standard describing structures that can be used to link the enterprise resource planning (ERP) layer to the manufacturing execution system (MES) layer [1]. Specifically, among other entities, it provides classes dealing with the machinery available in production environments such as robots, conveyer belts, etc., as well as structures depicting processes, production steps and operational data. For data exchange, an XML-based serialization of IEC 62264 has been defined in a separate standard, the Business to Manufacturing Markup Language (B2MML) [2].

AutomationML is an XML-based standard targeting the modeling of production environments [3]. Several extensions exist that, e.g., enable annotating AutomationML models with information about the geometry and kinematics of machines, as well as their internal behavior. AutomationML is based on the Computer Aided Engineering Data-Exchange-Metamodel (CAEX), which is an XML dialect that has been standardized in IEC 62424 [4]. AutomationML adds certain rules and restrictions to it, as well as a set of basic instance elements, in order introduce computer-readable semantics and to make it more usable for the modeling of automated production systems.

IEC 62264-2 and AutomationML can co-exist as separate views on the same production system, but there is some overlap with respect to the definitions of the entities in IEC 62264-2 and AutomationML. Therefore, a semantic alignment of entities as well as two methods for integration are proposed: (i) tagging AutomationML elements with IEC 622664-2 roles and (ii) referencing external IEC 62264-2 data encoded in B2MML documents.

II. RELATED WORK

Alignment and integration of related standards has been investigated intensively in the model-driven engineering community, where one way of dealing with partly overlapping information from different domains is through model transformations between instances of different metamodels [5], [6]. However, before the actual transformations can be written, the semantic alignment of metamodel elements needs to be accomplished. A way to automatically generate this alignment is described in [7], where schema matching is achieved through similarity flooding. While the general idea behind this approach is very promising and definitely a step in the right direction, it has two drawbacks: (i) the alignment results need to be checked manually for correctness and (ii) the resulting alignment may differ based on certain user-defined threshold values. For our work, we have therefore not used an automated alignment approach, but we have manually defined the alignment of (i) metamodel elements and (ii) certain predefined model elements.

This work builds on previous work: an initial metamodel alignment between IEC 62264-2 and AutomationML has been proposed in [8]. While most of the concepts and mappings defined there can be used as a basis for a complete alignment of IEC 62264-2 and AutomationML, some specifications need to be reconsidered (e.g., the mapping of equipment capability test results).

Another view on the rationale behind the linking of AutomationML and IEC 62264-2 has been given in [9]. There, background information and related activities in the AutomationML consortium are presented, together with further use cases about using AutomationML at higher levels of the automation hierarchy.

Using IEC 62264-2 to connect to business models has been presented in [10]. The business model language that has been used in this alignment is called “Resource-Event-Agent” (REA) [11]. Further information about REA can be found, e.g., in [12], [13] and [14]. In [15] it was argued, that IEC 62264 could further be used as a specification language for the task layer of REA when it comes to the modeling of transformations (which really means “production” in this sense, as opposed to *transfers* that deal with buying and selling products and material).

Following this, an alignment of AutomationML with IEC 62264-2 *and* with REA has been presented, that showed conceptual congruence between sequentially stacked standards. This alignment showed that single pieces of information can be tracked “transitively” between different domains, given that a proper alignment of the metamodels has been made, and that this alignment can be formalized by model transformations and validation techniques [16].

Integrating existing standards into AutomationML has been exemplarily explained in [17], by providing rule-sets for the explicit linking of AutomationML elements and COLLADA[1] elements. With that it is possible to provide geometric and kinematic information to AutomationML entities.

The general architecture for referencing externally stored data and information has been defined in a best practice recommendation for AutomationML edition 1 [18] and in part 1 of AutomationML edition 2 [3]. We are basing our work on the rules defined in these documents.

A more general way of specifying cross-domain knowledge has been presented in [19] by providing a “linking metamodel”. Specific kinds of links can be used to explicitly relate a left side and a right side “object”. This model-driven approach has been brought to the OPC Unified Architecture (UA) domain in [20], where an extensible information model for the specification of cross-domain relations has been proposed, based on OPC UA reference types.

The AutomationML consortium provides an interactive environment for the discussion and definition of application-specific use cases through a number of domain-specific working groups and a global workshop team. Results of these internal consultations are published as *best practice recommendations* (that might become part of the AutomationML standard in later revisions) as well as *application recommendations*. One application recommendation, e.g., deals with the configuration of automation projects and how to represent cross-domain information in AutomationML [21]. Another one deals with the representation of information from IEC 62264-2 in AutomationML, *AR-MES-ERP* [22]. We are going to explain the basic concepts of AR-MES-ERP in this work, in Sec. III. Like all other *best practice* and *application recommendations*, AR-MES-ERP is available on the website of the AutomationML e.V.: https://www.automationml.org/.

III. IEC 62264-2 for AutomationML

In a first step, part 2 of IEC 62264 has been investigated for integration with AutomationML, as it sports the most immediate overlap of concepts and features. Also, part 2 declares the basic models that are reused in later parts and thus it is the natural way to start.

We have approached the alignment and integration of IEC 62264-2 with AutomationML in two steps: (i) alignment of IEC 62264-2 and AutomationML metamodel elements (cf. Sec. III-A) and (ii) alignment of IEC 62264-2 metamodel elements with elements of AutomationML that have been defined in standardized libraries (cf. Sec. III-B).

[1]cf. https://www.khronos.org/collada/

A. Alignment of Metamodels

We have been approaching the metamodel alignment from the point of view of IEC 62264-2. We wanted to find zero, one or more metamodel elements of CAEX, whose instances would be the most appropriate ones in order to represent a specific IEC 62264-2 instance.

We have collected all classes of IEC 62264-2 that are represented in corresponding class diagrams and matched them with semantically and/or structurally equivalent entities of the CAEX metamodel. An excerpt of the overall matching is is given in Fig. 1, with the most important alignment decisions as follows: (i) all kinds of properties are realized as CAEX Attributes, (ii) only a few elements are aligned with System Unit Classes and Role Classes and (iii) most of the IEC 62264-2 elements are realized in terms of CAEX Internal Elements.

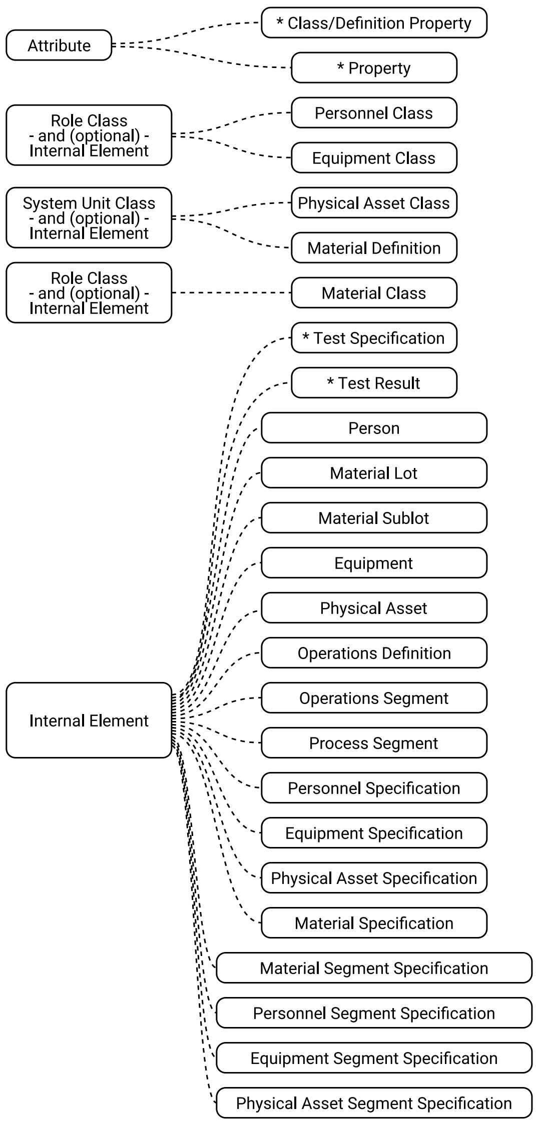

Figure 1. Excerpt of the alignment of CAEX and IEC 62264-2.

1) X → Attribute: The different properties that occur in IEC 62264-2 can be well matched with the Attribute type that is provided by CAEX. Most importantly, both metamodels allow them to have sub-Attributes/Properties, which makes them very compatible.

2) X → System Unit Class: Semantically, System Unit Classes are meant to represent vendor-specific reusable entities, such as types of robots. Structurally, System Unit Classes act like class descriptions that can be instantiated through Internal Elements. An important aspect of the class-instance relation is, as it is realized in CAEX, that an Internal Element might only be the instance of *one* (or zero) System Unit Class. In IEC 62264-2 there are two class-instance relations with such an $1 : n$ cardinality: Physical Asset Class to Physical Asset and Material Definition to Material Lot. Physical Asset Classes are an ideal candidate for System Unit Classes, as they, too, represent vendor-specific reusable entities, such as a specific type of robot (which has its own model number). Also Material Definitions can be understood as vendor-specific entities, as they might represent a specific type of raw material, such as a coating powder of a specific color from a certain vendor.

3) X → Role Class: Role Classes enable the classification of both System Unit Classes and Internal Elements, and multiple Role Classes can be referenced by a single System Unit Class or Internal Element. This $m : n$ cardinality is exactly what is needed for relating IEC 62264-2 Personnel Classes, Equipment Classes and Material Classes to their respective instances. This means, that, e.g., realizing personnel classes in AutomationML is realized in one or more Role Class Libraries.

4) X → Internal Element: All other entities of IEC 62264-2 are realized as Internal Elements, including, e.g., Equipment, Process Segments and Operations Definitions. This goes well together with the definition of the product-process-resource (PPR) concept as it is available in AutomationML [23]. More on this specific alignment will follow in the next sub-section.

5) X → External Interface and Internal Link: References between IEC 62264-2 metamodel entities that cannot be handled by super-sub or class-instance relations need to be realized using External Interfaces (EI) and Internal Links (IL) in AutomationML. Among these are the references that follow the blueprint of Equipment Segment Specification → Equipment/Equipment Class. These kinds of references are realized with `ResourceConnectors`, as depicted in Fig. 2. `ResourceConnectors` are also used for linking Operations Segments to Process Segments and in further, similar cases.

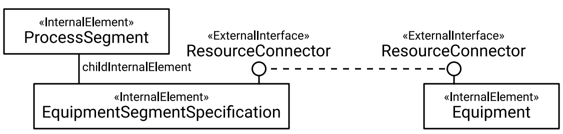

Figure 2. Realizing a reference from an Equipment Segment Specification to an Equipment through two interlinked `ResourceConnectors`.

Other EI-IL-EI implementations are used for (i) connecting resources to Hierarchy Scopes through `HierarchyScopeConnectors`, (ii) Equipment to Physical Assets through `EquipmentAssetConnectors`, (iii) Process Segments to other Process Segments through `DependencyConnectors` and (iv) resources classes to Test Specifications to Test Results to resources through `TestConnectors`. Some of these references realize association classes and thus feature an intermediate Internal Element as a reification of that reference that can be further equipped with Attributes, such as the Equipment Asset Mapping depicted in Fig. 3.

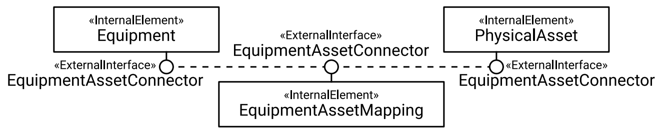

Figure 3. Realizing a reference from a piece of Equipment to a Physical Asset over an Equipment Asset Mapping through three interlinked `EquipmentAssetConnectors`.

B. Alignment with AutomationML Libraries

For IEC 62264-2, the most important Role Classes in AutomationML are `Product`, `Process` and `Resource`, as well as their `*Structure` versions. Fig. 4 shows an alignment of IEC 62264-2 metamodel elements with AutomationML-specific CAEX instance elements with the following proerties: a relation between an IEC 62264-2 element and an AutomationML element means, that a Role Class has been defined in AutomationML that is a sub-class of the identified pre-existing AutomationML Role Class. In case there exists no such Role Class, `AutomationMLBaseRole` is chosen as the super-class. It turns out that most of the elements of IEC 62264-2 have no direct relation to AutomationML elements, however in some cases there are very specific overlaps in terms of semantics.

For instance, Equipment, Equipment Class, Physical Asset and Physical Asset Class are all possible instances of the `Resource` Role Class of the AutomationMLBaseRoleClassLib. Therefore, we have made the Role Classes of these elements sub-classes of `Resource`. Similar reasoning can be applied to the Material* entities and `Product` as well as to the *Segment entities and `Process`.

Other elements, such as Operations Definitions match well with `ProcessStructures`, as the later is "an abstract role type for a process oriented object hierarchy" [24], which is also what an Operations Definition is: it subsumes a set of Operations Segments.

Other related Role Classes are defined in [25] and include elements that can mainly be used as additional Role Classes for pieces of Equipment, such as `BatchManufacturingEquipment`, `Transport`, `Storage`, `Machine`, `Communication`, `Controller`, `NC`, `PLC` and `ContManufacturingEquipment`.

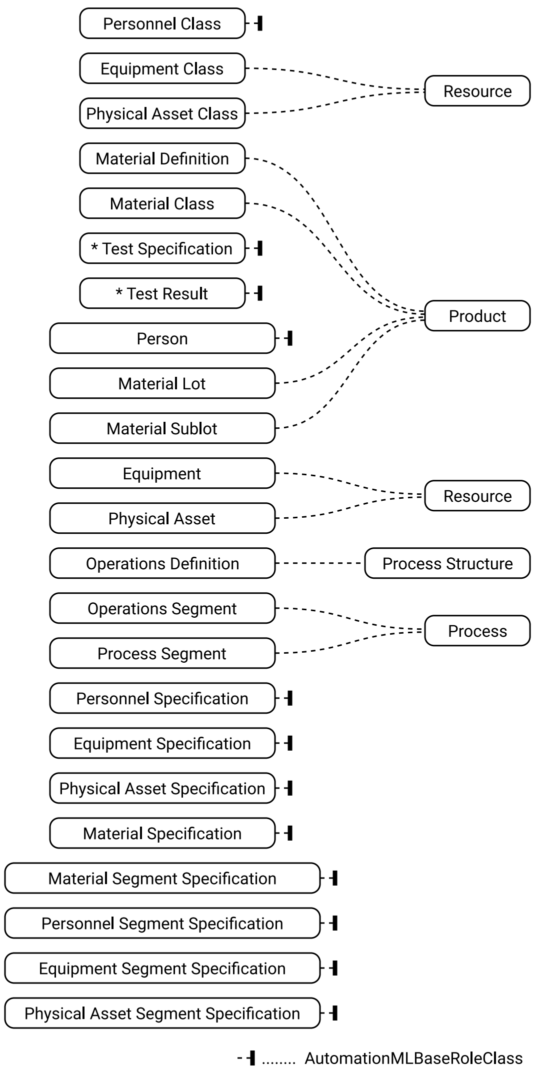

Figure 4. Excerpt of the alignment of IEC 62264-2 with AutomationML standard Role Classes.

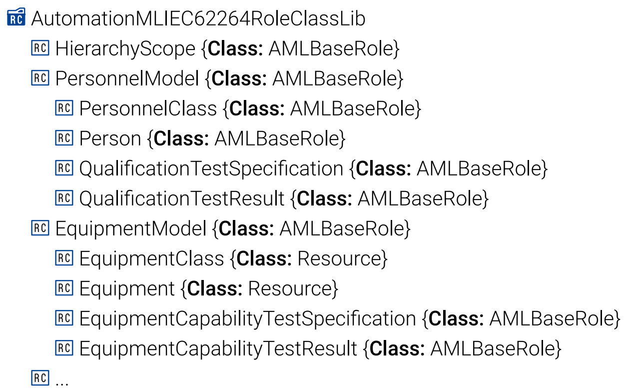

Figure 5. Excerpt of the IEC 62264 Role Class Library. . has been abbreviated as `AMLBaseRole` due to obvious width constraints.

In the non-normative AutomationMLExtendedRoleClassLib, there exist even several role classes that directly implement elements coming from IEC 62264 vocabulary, such as `Enterprise`, `Site` and `Area`. These role classes can be used as additionally supported role classes in order to further tag certain elements (usually Equipment).

C. Application

1) IEC 62264 Role Class Library: In order to enable the modeling of AutomationML entities as instances of certain IEC 62264-2 concepts, the AutomationMLIEC62264RoleClassLib has been defined, an excerpt of which is depicted in Fig. 5. It defines a Role Class for each of the IEC 62264-2 concepts and structures them in several folders, one for each information model. Whenever an AutomationML entity corresponds to an IEC 62264-2 concept, the according Role Class should be added to that entity in order to mark it as an instance thereof (and to inherit pre-defined Attributes that stem from the declarations available in [1]).

The Role Classes in the AutomationMLIEC62264RoleClassLib each define all the properties that are defined in [1] for the IEC 62264-2 entities. It can therefore be used to realize IEC 62264-2-compliant models in pure AutomationML. No hard dependencies on other data are defined.

However, it is also possible to model only necessary parts of IEC 62264-2-compliant models in AutomationML and refer to external sources for the details. This approach is further explained in Sec. III-C3; it relies on the referencing of data that is stored in separate B2MML documents.

As an example, let us consider the equipment information model that is provided under the `EquipmentModel` Role Class. One of the Role Classes defined there is `EquipmentClass`; as its name suggests, it corresponds to the IEC 62264-2 entity Equipment Class. Given the alignment discussed in the first few sub-sections of Sec. III, an implementor needs to instanciate Equipment Classes in terms of Role Classes in some Role Class Library. Such an Equipment Class instance needs to specify `EquipmentClass` as its parent Role Class. If an instance of an `EquipmentClass` (i.e., an Equipment) is to be modeled, it must be realized as an Internal Element that needs to relate to two Role Classes: (i) it must refer to the `Equipment` Role Class of the AutomationMLIEC62264RoleClassLib and (ii) it must refer to the user generated Role Class that represents an Equipment Class.

Since the `EquipmentClass` and the `Equipment` Role Classes both have `Resource` es their parent class, the user generated Equipment Class and Equipment are also indirectly tagged as `Resource`.

2) IEC 62264 Interface Class Library: The AutomationMLIEC62264InterfaceClassLib defines the required Interface Classes to model dependencies between IEC 62264-2-related entities in AutomationML, such as the ones described in Sec. III-A5. The complete Interface Class Library is depicted in Fig. 6. All Interface Classes are subclasses of `AutomationMLBaseInterface`, even though the `ResourceConnector` is very much aligned to the standardized `PPRConnector` Interface Class. There are two reasons why the former is not a sub-class of the latter: (i) `ResourceConnector` connects slightly different entities with each other (not the `Process` itself, but a child-element of it) and (ii) it is designed to also connect other entities than just PPR (such as personnel).

- AutomationMLIEC62264InterfaceClassLib
  - HierarchyScopeConnector {**Class:** AutomationMLBaseInterface}
  - ResourceConnector {**Class:** AutomationMLBaseInterface}
  - EquipmentAssetConnector {**Class:** AutomationMLBaseInterface}
  - DependencyConnector {**Class:** AutomationMLBaseInterface}
  - TestConnector {**Class:** AutomationMLBaseInterface}

Figure 6. The IEC 62264 Interface Class Library.

3) B2MML Role Class Library: With the AutomationMLB2MMLRoleClassLib the required Role Class for referring to external data is defined (cf. Fig. 7). The rationale behind this is the following: with B2MML, an XML-implementation for IEC 62264 models has been defined that can be used to serialize IEC 62264-compliant models from (industrial) tools and applications. The `B2mmlData` Role Class allows the modeling of a document node that references an external B2MML document through an `ExternalDataReference` that has been defined in [18]. For that, it specifies the `refURI` Attribute with a path to the document in question. For the referencing of B2MML documents, the `MIMEtype` of the `ExternalDataReference` must be "`application/x.b2mml+xml`".

- AutomationMLB2MMLRoleClassLib
  - B2mmlData {**Class:** ExternalData}

Figure 7. The B2MML Role Class Library.

Fig. 8 presents the referencing of a B2MML document in a pictorial way, while highlighting the main features: (i) a single document (here: `pas.b2mml`) can be referred to several times and (ii) a specific sub-entry of the B2MML document can be referred to by appending the ID of the element to the fragment part of the URI, separated by a hash (#).

Here, two Internal Elements representing Physical Assets are modeled, each of which refers to an entity that is available in an externally filed B2MML document. Additional information about this entity, such as further Properties may be available in this external document and thus is not required to be modeled in AutomationML.

IV. Example

In this section we give a simple example of an AutomationML document that (i) encodes IEC 62264-2 information and (ii) references an external B2MML document. The system to be modeled comprises three pieces of Equipment: a `Turntable`, a `Conveyer` and a `Robot`. For the system under observation, only the `Turntable` is of interest and so only this element is modeled in AutomationML. However, in the information system, which is able to output B2MML-encoded IEC 62264-2 data, more pieces of Equipment are listed.

The instance hierarchy excerpt of the AutomationML document is presented in Listing 1. In line 12 an External Interface of type `ExternalDataReference` is created that refers to the Equipment with ID `Turntable` in file `Equipment.b2mml` (line 23).

Listing 1. InstanceHierarchy excerpt of an AutomationML document that references IEC 62264-2 data encoded in an external B2MML document. UUIDs have been replaced with integer values, class references have been truncated to the last segment of the path, i.e., the name of the class.

```
1 ...
2 <InstanceHierarchy Name="B2mmlElement">
3   <Description>
4     Example for referencing an IEC 62264 element
5     that is one of many elements within a B2MML
6     document.
7   </Description>
8   <Version>1.0.0</Version>
9   <InternalElement Name="Turntable" ID="1">
10    <InternalElement
11      Name="TurntableB2mmlDocument" ID="2">
12      <ExternalInterface Name="B2mml" ID="3"
13        RefBaseClassPath="ExternalDataReference">
14        <Attribute Name="MIMEType"
15          AttributeDataType="xs:string">
16          <Value>
17            application/x.b2mml+xml
18          </Value>
19        </Attribute>
20        <Attribute Name="refURI"
21          AttributeDataType="xs:anyURI">
22          <Value>
23            ./Equipment.b2mml#Turntable
24          </Value>
25        </Attribute>
26      </ExternalInterface>
27      <RoleRequirements
28        RefBaseRoleClassPath="B2mmlData" />
29    </InternalElement>
30    <SupportedRoleClass
31      RefRoleClassPath="WorkCell" />
32    <RoleRequirements
33      RefBaseRoleClassPath="Equipment" />
34  </InternalElement>
35 </InstanceHierarchy>
36 ...
```

The B2MML document in turn is instantiated in a file named `Equipment.b2mml`. It holds information about various pieces of equipment (cf. Listing 2). The Equipment that is referred to in the AutomationML document is defined starting with line 15; the ID of this Equipment is stated in line 16.

Listing 2. Document `Equipment.b2mml`, holding IEC 62264-2 information about equipment.

```
1 <EquipmentInformation
2   xmlns="http://www.mesa.org/xml/B2MML-V0600">
3   <Equipment>
4     <ID>Conveyer</ID>
5     <Description>
6        Delivers products from one work cell
7        to another.
8     </Description>
9     <EquipmentLevel>
10      <EquipmentElementLevel>
11        WorkCell
```

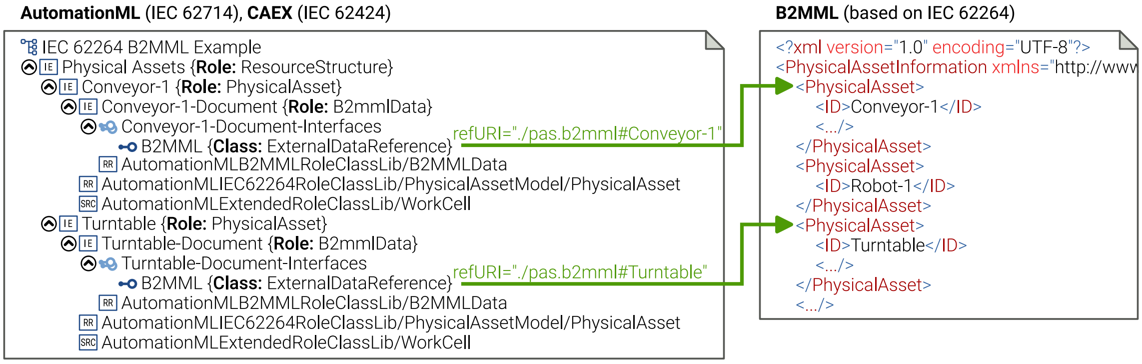

Figure 8. Referencing externally stored Physical Asset information through `ExternalDataReferences`.

```
      </EquipmentElementLevel>
    </EquipmentLevel>
  </Equipment>
  <Equipment>
    <ID>Turntable</ID>
    <Description>
      Turns products towards distinct work cells
      for further production.
    </Description>
    <EquipmentLevel>
      <EquipmentElementLevel>
        WorkCell
      </EquipmentElementLevel>
    </EquipmentLevel>
  </Equipment>
  <Equipment>
    <ID>Robot</ID>
    <Description>
      Assembles products from intermediate
      products.
    </Description>
    <EquipmentLevel>
      <EquipmentElementLevel>
        WorkCell
      </EquipmentElementLevel>
    </EquipmentLevel>
  </Equipment>
</EquipmentInformation>
```

V. Conclusion

We have presented work that has been recently published as an AutomationML application recommendation (AR-MES-ERP [22]) and provided some of the rationale behind the main design decisions. With this application recommendation it is possible to model IEC 62264-2-compliant information using standard AutomationML methods.

Users can choose between (i) modeling in pure AutomationML or (ii) modeling in AutomationML and B2MML. The extent to which information is modeled in AutomationML is completely up to the user or tool creating such models.

It should further be noted that integration with OPC Unified Architecture (UA) has been considered, since both, AutomationML [26] and IEC 62264-2 [27] have corresponding companion specifications for the application in OPC UA. In cases where `ExternalDataReferences` are used, an alignment of AutomationML and IEC 62264-2 is realized through the `HasAMLUAReference` Reference Type [26]. In cases where IEC 62264-2 information is modeled in AutomationML without external references, another Reference Type has been defined [22]: `HasAMLIEC62264UAReference`. It needs to be added in a separate processing step, as there is no standard way to link an AutomationML entity to a IEC 62264-2 entity in an unambiguous way; scenario-specific knowledge is required for creating such References.

Currently, the application recommendation only includes information of part 2 of IEC 62264. The incorporation of other parts, specifically part 4 would allow even more advanced models. Part 4 not only provides much more detailed modeling capabilities when it comes to the description of processes (e.g., workflow specifications, job orders and work definitions), but it also provides some generally useful models, such as resource relationship networks that allow the modeling of gas supply, material flow routing, etc.

Acknowledgment

The authors would like to thank the participants of the AutomationML sub-working-group on the “Application of AutomationML at Higher Levels” as well as the “AutomationML Workshop” team for the fruitful discussions that helped shaping the application recommendation AR-MES-ERP that is the basis of this document.

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged.

References

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

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

[3] 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. 2.0, 2018, IEC 62714-1:2018.

[4] 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. 2.0, 2016, IEC 62424:2016.

[5] S. Sendall and W. Kozaczynski, "Model transformation: The heart and soul of model-driven software development," *IEEE Software*, vol. 20, no. 5, pp. 42–45, 2003.
[6] T. Mens and P. Van Gorp, "A taxonomy of model transformation," *Electronic Notes in Theoretical Computer Science*, vol. 152, pp. 125–142, 2006, proceedings of the International Workshop on Graph and Model Transformation (GraMoT 2005). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1571066106001435
[7] J.-R. Falleri, M. Huchard, M. Lafourcade, and C. Nebut, "Metamodel matching for automatic model transformation generation," in *Model Driven Engineering Languages and Systems*, K. Czarnecki, I. Ober, J.-M. Bruel, A. Uhl, and M. Völter, Eds. Springer Berlin Heidelberg, 2008, pp. 326–340.
[8] B. Wally, C. Huemer, and A. Mazak, "Entwining plant engineering data and ERP information: Vertical integration with AutomationML and ISA-95," in *Proceedings of the 3rd IEEE International Conference on Control, Automation and Robotics (ICCAR 2017)*, 2017.
[9] B. Wally, M. Schleipen, N. Schmidt, N. D'Agostino, R. Henßen, and Y. Hua, "AutomationML auf höheren Automatisierungsebenen," in *Proceedings of AUTOMATION 2017*, 2017.
[10] B. Wally, C. Huemer, and A. Mazak, "Aligning business services with production services: The case of REA and ISA-95," in *Proceedings of the 10th IEEE International Conference on Service Oriented Computing and Applications (SOCA 2017)*, 2017.
[11] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), *Business Transaction Scenarios—Accounting and Economic Ontology*, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) Std., 2007, ISO/IEC 15944-4:2007(E).
[12] W. E. McCarthy, "The REA accounting model: A generalized framework for accounting systems in a shared data environment," *The Accounting Review*, vol. 57, no. 3, pp. 554–578, 1982.
[13] G. L. Geerts and W. E. McCarthy, "The ontological foundation of REA enterprise information systems," in *Annual Meeting of the American Accounting Association, Philadelphia, PA*, vol. 362, 2000, pp. 127–150.
[14] G. L. Geerts and W. E. McCarthy, "An ontological analysis of the economic primitives of the extended-REA enterprise information architecture," *International Journal of Accounting Information Systems*, vol. 3, no. 1, pp. 1–16, 2002.
[15] B. Wally, C. Huemer, and A. Mazak, "ISA-95 based task specification layer for REA in production environments," in *Proceedings of the 11th International Workshop on Value Modeling and Business Ontologies (VMBO 2017)*, 2017.
[16] B. Wally, C. Huemer, and A. Mazak, "A view on model-driven vertical integration: Alignment of production facility models and business models," in *Proceedings of the 13th IEEE International Conference on Automation Science and Engineering (CASE 2017)*, 2017.
[17] AutomationML consortium, *AutomationML Whitepaper Part 3—Geometry and Kinematics*, Whitepaper, Rev. 2.0, 2015, AML Part 3:2015. [Online]. Available: https://www.automationml.org/
[18] AutomationML consortium, *AutomationML Best Practice Recommendation—External Data Reference*, Best Practice Recommendation, Rev. 1.0.0, 2016, AML BPR-EDR:2016.
[19] S. Feldmann, M. Wimmer, K. Kernschmidt, and B. Vogel-Heuser, "A comprehensive approach for managing inter-model inconsistencies in automated production systems engineering," in *Proceedings of the 12th IEEE International Conference on Automation Science and Engineering (CASE 2016)*, 2016, pp. 1120–1127.
[20] B. Wally, C. Huemer, A. Mazak, and M. Wimmer, "AutomationML, ISA-95 and others: Rendezvous in the OPC UA universe," in *Proceedings of the 14th IEEE International Conference on Automation Science and Engineering (CASE 2018)*, 2018.
[21] AutomationML consortium, *Automation Project Configuration*, Application Recommendation, Rev. 1.0.0, 2017. [Online]. Available: https://www.automationml.org/
[22] B. Wally, *Provisioning for MES and ERP—Support for IEC 62264-2 and B2MML*, Application Recommendation, Rev. 1.0.0, 2018. [Online]. Available: https://www.automationml.org/
[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] 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, 2014, IEC 62714-1:2014.
[25] 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, 2015, IEC 62714-2:2015.
[26] Deutsches Institut für Normung (DIN), *Combining OPC Unified Architecture and Automation Markup Language*, DIN SPEC, 2016, DIN 16592:2016-12.
[27] OPC Foundation, *OPC Unified Architecture for ISA-95 Common Object Model*, Companion Specification, Rev. 1.00, 2013.