Aligning Business Services with Production Services: The Case of REA and ISA-95

Bernhard Wally*, Christian Huemer* and Alexandra Mazak*,†
*Business Informatics Group and †CDL-MINT, TU Wien
Favoritenstraße 9–11/188-3, 1040 Vienna, Austria
{wally, huemer, mazak}@big.tuwien.ac.at

***Abstract*—"Industrie 4.0" aims at flexible production networks that require horizontal integration across companies. Evidently, any production related information exchanged in the network must be vertically forwarded to the corresponding service endpoints of the local production system. Accordingly, there is a need to align information that flows between companies and within each company. The Resource-Event-Agent (REA) business ontology describes a metamodel for internal business activities (e.g., production) and for inter-organizational exchange constellations on the enterprise resource planning (ERP) level. ISA-95 is a series of standards targeting the integration of enterprise control systems on the interface between ERP systems and manufacturing execution systems. Consequently, we align elements of REA and ISA-95 and define conversion rules for the transformation of elements from one system to the other. By interleaving the semantics of both standards, we formally strengthen the links between the services of the business level and the production level, and support multi-system adaptation in flexible production environments.**

***Index Terms*—REA; ISA-95; Metamodel Alignment**

I. Introduction

The digital manufacturing enterprise with fully integrated services, from inter-company logistics to intra-company workflows and data management, is a topic of high interest in both academia and industry. Various terms have been coined for a new *digital manufacturing age*, such as "fourth industrial revolution", "Industrie 4.0", and "smart production". Along have come some high level concepts and ideas on what such a service integration could achieve and provide [1]. In the depths of detailed elaboration on these high level concepts specific integration points need to be defined and justified.

One such initiative for the integration of different system levels is the international standard series for enterprise control system integration IEC 62264, based on the standard ISA-95 of the American National Standards Institute and the International Society of Automation. Part 2 of ISA-95 provides a data model for bridging two different kinds of activities in a manufacturing enterprise, namely the higher level "Business Planning and Logistics" and the lower level "Manufacturing Operations Mangement". This kind of integration is considered *vertical integration* [2], [3] and can go far beyond the manufacturing operations management layer, e.g., down to programmable logic controllers and even single sensors or actuators. The importance of vertical integration in production processes is emphasized in [4] and [5], where a domain specific modeling language, derived from the Business Process Model and Notation (BPMN), is introduced.

Another dimension of service integration is *horizontal integration*, which applies to both internal systems integration [2] and integration among different parties that are directly or indirectly involved in production (such as suppliers, customers, sub-contractors, logisticians, etc.) [6], [7]. With regards to inter-company horizontal integration the following international standard has been defined: IEC 15944-4 is part 4 of the business operational view of information technology systems and it is based on the Resource-Event-Agent (REA) ontology.

In this work, we are presenting the alignment of two standards: (i) IEC 15944-4 (from now on we use the more readable term *REA*) is a business metamodel that can resemble an enterprise resource planning (ERP) or other management information system [8], and that enables the modeling of the inter-enterprise collaboration and (ii) IEC 62264 (we will use its alias name *ISA-95* in the remainder of this contribution) is a technology for internal vertical integration, from the ERP layer down to manufacturing operations.

By aligning the conceptual models of these standards, (i) it becomes feasible to reuse existing information from one environment as a skeleton or base structure for building a corresponding model in the other environment and (ii) it fosters transforming static and dynamic (speak: runtime) artifacts between different services. With these two aspects supported, enterprise services can e.g., more agile respond to changes in an enterprise's production configuration by adapting the model of one IT system based on changes made in the other system.

For further motivation of our approach, let us consider the following example: an additional production step is required for producing a product with specific properties. In this case, on the MES layer a new process segment is defined. By having the ERP and MES aligned, the new process segment can be synchronized to the ERP system by automatically creating a corresponding business model. Ultimately, this helps keeping systems in sync, reduces manual typing errors, and saves time.

By providing a generic alignment, systems of different vendors can inter-operate with much less effort, given that they support these standards. Instead of defining $2 \times (n \times m)$ mappings between n ERP tools and m MES tools, only $2 \times (n+m)$ mappings are required (import and export facilities for the two standards of each tool). In case they do not directly implement the standards discussed in this paper, the work presented here might resemble a good starting point for the integration of proprietary business and manufacturing systems.

The remainder of this paper is structured as follows: in Section II, condensed background information about the involved standards and their conceptual models is given. Then, after related work is presented in Section III, our alignment approach and implementation is described in Section IV. In Section V we will provide an application scenario that shows a possible use case of the alignment of these two standards.

II. Background

A. Resource-Event-Agent (REA)

The term "Resource-Event-Agent" (REA) was coined in the early 80's by McCarthy, who consolidated the then current ideas of accounting research in a unified framework [9]. In its initial form, REA describes three concepts: economic resources, economic events, and economic agents. Following accounting theory, an economic event resembles something that has actually happened and that causes a record in the general ledger, such as paying for raw material, receiving money for selling finished products, or renting offices.

The initial REA model was extended and refined to a more complete business ontology comprising new types of events for production and a planning layer that allows the specification of commitments, schedules, policies, etc. [10], [11], [12], [13]. E.g., commitments were introduced as planning layer equivalent of events: they store information about what has been agreed on to be happening in the future, while events resemble what has actually happened. Commitments are "fulfilled" by events (if the course of action follows the plans).

As a modeling pattern, the type-object (or power type) pattern [14], [15], [16] is often used in REA related literature to enable further structuring of concepts and configuration at runtime [17], [13], [8]. In short, the type-object pattern proposes to provide a type class and an object class for the specification of a specific entity type and its instances. E.g., in order to support multiple types of agents and instances of these agents, a class `Agent Type` and a class `Agent` would be defined. Each `Agent` instance (e.g., a person called "John Smith" who is employed as a salesman at a company) would be associated with the specific `Agent Type` instance "Salesman". A new type of agent (e.g., "Cashier") could be added at runtime by creating a new instance of `Agent Type`; then, `Agent` instances could impersonate this type of agent.

In Fig. 1, the metamodel of the object layer of the REA business model language is depicted. Relations (duality, stockflow, participation, etc.) have been reified by correspondingly named classes. The type layer has been largely excluded in order to make the diagram more clear—basically, it resembles a conceptual copy of the object model. Classes suffixed with `␣Type` (`Resource Type` and `Agent Type` are two example classes of the type layer presented here) belong to the type layer. `Types` provide a concept for the configuration of the system at runtime, including the declaration of attributes through `Attribute Types`. Attributes and attribute types support nested attributes.

B. Enterprise-Control System Integration: ISA-95

ISA-95 is a series of standards that addresses the integration of the enterprise domain with the manufacturing and control domains. It provides a standard terminology and set of concepts for system integration [18]. The relevant part of ISA-95 for this work is part 2, as it is specified in IEC 62264-2:2013 [19].

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). All-in-all, ISA-95 part 2 differentiates between ten different models that are briefly introduced below.

Personnel comprises the classes of personnel and individuals that are required to operate manufacturing processes.

Equipment represents the equipment of an organization in form of a role based model. Specific examples include welders, titration testers, reactors, etc. Equipment may define a hierarchy (sites, areas, work centers, etc.), and it resembles a role model that is instantiated through physical assets.

Physical Assets represent 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. A dedicated relation (equipment asset mapping) records the time frame in which a physical asset was associated with an equipment. Physical assets have an impact on the ERP layer, as they are usually of value and need to be tracked financially.

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

Process Segments resemble 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.

Operations Definitions define the resources required to perform a specified operation (production, maintenance, quality test, and inventory). One operations definition corresponds to one or more process segments.

Operations Schedules represent requests for specific operations (production, maintenance, quality test, and inventory) to be performed within a defined date and time frame.

Operations Performances are reports on requested operations (production, etc.) that have actually been performed.

Operations Capabilities are collections of information about all resources for operations for a defined date and time frame (past and future), including terms, statuses, and quantities.

Process Segment Capabilities are logical groupings of resources that are committed, available, or unavailable for a defined process segment for a specific time.

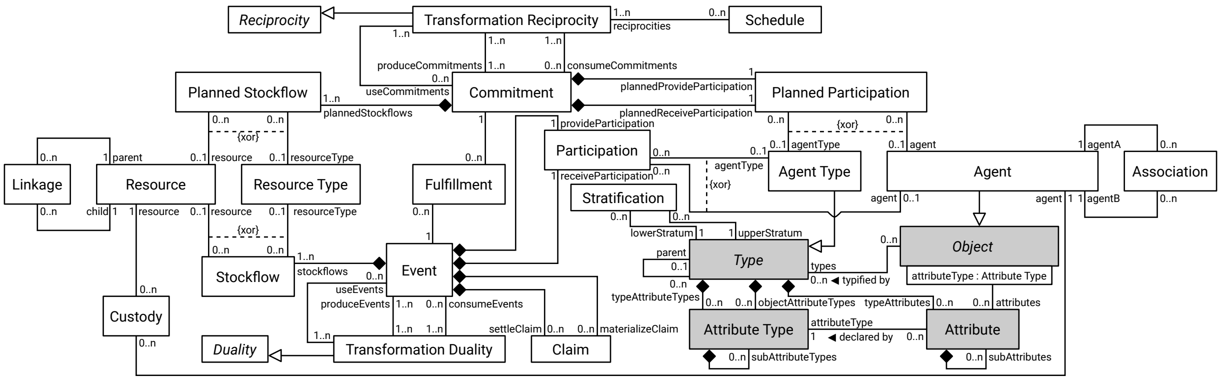

Fig. 1. Class diagram of the REA object layer—all object classes (i.e., all white shaded classes that do not end with "Type") inherit from class `Object`, while type classes inherit from `Type`. For better readability, only the the generalization relationships for `Agent` and `Agent Type` have been considered in this diagram. The type-object sub model is depicted with gray shading.

III. Related Work

This work resembles an improvement over work presented in [20] and [21], where an alignment of REA and ISA-95 has first been proposed and described based on a very reduced set of elements of both ISA-95 and REA. In this paper, we will discuss a much more complete metamodel of REA (including the *typification* concept and the *planning layer*) and all metamodels of ISA-95 part 2. Still, a special focus will be put on the *process segment* metamodel—because an exhaustive description of all ISA-95 metamodels would exceed the space provided in this paper by far.

Another difference of this work compared to [20] and [21] is the basic approach: integration of REA and ISA-95 in [20] and [21] was realized by extending the metamodel of REA in order to reflect elements of ISA-95 (and thus provide a way to model ISA-95 elements within REA, but not the other way round), whereas in this paper we will provide a stringent alignment on the metamodel level that is direction agnostic. An early approach thereof was presented in [22], where even production facility models where taken into account, but the alignment of REA and ISA-95 has not been defined in detail.

A different kind of entanglement between REA and ISA-95 has been proposed in [23], exploiting the fact that REA leaves open how tasks that are of a lower granularity than what is usually captured with "events" are designed or realized. As such, the concepts defined in [24] (which are situated at a level of higher detail than REA) are used for the implementation of the task specification layer. This kind of arrangement is complementary to what is presented here (the alignment of semantically similar concepts).

ISA-95 has been aligned with another standard further down the automation hierarchy, AutomationML[1]: [25] provides a mapping of concepts between these two standards and facilitates the integration of MES and ERP specific data into information models of plant topologies. Given this mapping, and the alignment provided in this work, a vertical information integration from the ERP level to the manufacturing level can be achieved, as has also been stated in [26].

Key challenges for software evolution in aPS are collected in [27], including co-evolving of interdisciplinary engineering models, which is one of the challenges addressed in this paper. One of the research goals stated in [27] is the development of automatic consistency mechanisms for domain specific systems. In this paper, we provide a model-driven approach for this challenge and evaluate it in a concrete scenario.

A mapping of the core parts of two modeling languages has been performed in [28], namely between AutomationML and SysML[2]. Similar to the approach given here, corresponding elements of the metamodels are explicitly being related to each other. Like in this work, the generic alignment of metamodel elements resembles a good way to transform model elements between two systems, which might be required through the use of different software tools in different engineering domains.

Mapping elements between different metamodels with explicit links has been presented in [29]. A linking metamodel has been defined that allows to map model elements of different metamodels without intruding into the models—the linking model is kept separate. We are not following this approach, although it should be applicable to our use case. Instead, we are relying on semantic similarities between the metamodels involved and use modeling conventions to make the mappings between model elements explicit.

A manufacturing information system that uses an enterprise service bus to integrate various systems from a manufacturing enterprise is presented in [30] and [31]. It stresses the fact that future manufacturing systems need to be very flexible in order to be able to provide quick time-to-market. We believe that the restructuring of a manufacturing system (and corresponding business applications) is well supported if these systems are tightly integrated with each other—this is a property that is explicitly fostered by the approach presented here.

In [32] an abstract classification of heterogeneities that can occur between two different metamodels is described, extending the "correspondence patterns" given in [33]. While

[1] cf. http://www.automationml.org/

[2] cf. http://www.omg.org/spec/SysML/

the given classification is not exhaustive, it contains some of the more common "solveable" mapping classes. Where appropriate, we will tag the mapping strategy with the abbreviation of the heterogeneity class of [32]. The heterogeneity feature model presented in [32] does not include one mapping schema, even though it is mentioned in the text of [32]: when there is no corresponding metamodeling concept. In [32], this is referred to as cardinality "1:0" and tagged as "Information Loss". We add corresponding features to the feature model and name the heterogeneity "Metamodeling Concept Missing", as this term goes well along the other related heterogeneity classes. The modified and condensed (only features resembling heterogeneity classes that are relevant for this publication are included) feature model is depicted in Fig. 2.

The work presented in [34] provides a manufacturing intelligence system based on ISA-95 and other, proprietary tools. In contrast to the work presented here, focus is laid on incorporation of production data for analytics and not on specific alignment of production information with ERP data.

An OPC UA information model for ISA-95 is defined in [35]. This mapping of ISA-95 components to the OPC UA protocol increases the reach of ISA-95 in industrial settings. With the mapping between REA and ISA-95 presented in this paper, OPC UA data containing ISA-95 information could be interpreted by REA-driven ERP level systems.

In [36], a subset of ISA-95 has been implemented inside a role class library for the data exchange format AutomationML [37]. AutomationML is situated at an even lower level than manufacturing execution systems in terms of IT systems' hierarchical classification in production environments [18], [19]. Accordingly, an alignment of REA with ISA-95 would yield, through a composition with the implementation of ISA-95 in AutomationML, even further vertical integration.

IV. Alignment of the ISA-95 and REA Metamodels

As the number of metamodel items from ISA-95 is too large to be presented here in full length, the process segment model is taken as an example for the complete mapping that has been realized—the process segment model resembles the integration model of the ISA-95 basic resource models. Also, by definition, process segments represent the smallest elements of manufacturing activities that are visible to business processes (such as REA), which makes this metamodel specifically interesting for our approach. Fig. 3 depicts the process segment model: it represents the usage of resources (such as material or equipment) in a specific production process. As such it can define the amount and kind of personnel, equipment, physical assets, and material. Additionally, process segments may define dependencies to other process segments with a relation such as "Process B may not run in parallel with Process A". Since the process segment model is a structural blueprint for the operations management information models of ISA-95, their basic alignment approach is similar.

The classification of heterogeneities (cf. Fig. 2) between the process segment metamodel of ISA-95 and the REA metamodel are specified in each mapping separately.

A. Alignment of Process Segment Metamodel Classes

Process Segments represent a bundling of personnel, equipment, physical asset, and material resources to conduct a production process. They define the process segments that are available in a production facility, and they serve as a starting point for the process definitions of specific kinds of products (operations definition), for the creation of schedules (operations schedule), for the logging of work actually done (operations performance), and for generating capability reports (process segment capability).

A process segment corresponds closely to a transformation reciprocity type or transformation duality type in REA. It is situated on the REA type layer rather than on the object layer, as it provides the definition of what is possible—it does not e.g., provide information about a specific production run. In a manufacturing environment it makes sense that the transformation reciprocity types and the transformation duality types are very synchronous: the former describes what kinds of schedules are possible, while the latter describes what kinds of production operations can be logged. A production schedule should only plan operations that can actually be conducted. As such, a process segment is aligned with both a *transformation reciprocity type* and a *transformation duality type*. The heterogeneity classes involved are (1:n) and (C2C).

Process segments in ISA-95 can be made up of other process segments; this capability is not available in classic REA reciprocity and duality types. Consequently, only process segments that are not made up of other process segments ("atomic" process segments) are aligned with the corresponding REA concepts. As such, process segments that comprise further process segments are decomposed into a set of such "atomic" process segments.

ISA-95 process segments do not define the intermediate layer that is defined in REA: commitment types and event types. Instead, a process segment directly links to the resources required. This level of indirection has to be created manually in case of a transformation from ISA-95 to REA: corresponding increment and decrement commitment and event types need to be defined, and the stockflow and participation types (see below) are then added to the corresponding commitment and event types.

Personnel Segment Specifications define the required personnel resources to conduct the corresponding process segment.

This corresponds to *planned participation types* (for a reciprocity types' commitment types) and *participation types* (in case of a duality types' event types). The attributes `Personnel Class` and `Person` of class personnel segment specification that hold the IDs of the referenced personnel resources need to be resolved to the actual entities and are then aligned with the `agent` and `agentType` references of class (planned) participation type.

Personnel segment specifications are directly referenced from a process segment, but in REA there exist the intermediate classes "commitment type" and "event type". These intermediate classes need to be instantiated and then reference

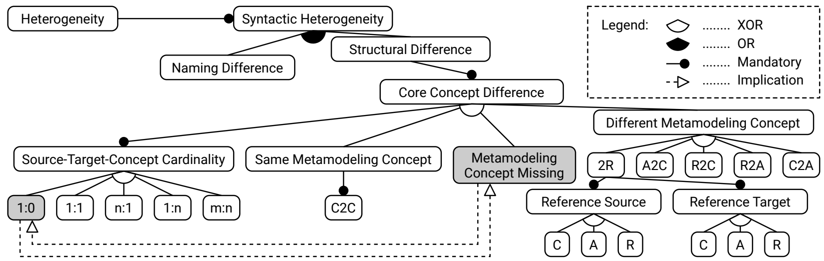

Fig. 2. Feature model for heterogeneity classes (from [32], condensed). Two additional heterogeneity classes have been added (shaded in gray), together with two feature dependencies (expressing that the two heterogeneity classes are essentially synonyms). In this work, references to feature nodes (in order to identify the heterogeneity classes of a given mapping) are realized like so: C2C.

to the corresponding (planned) participation types. The same procedure is required for the equipment, physical asset, and material segment specifications (see below) as well.

This structural difference comprises mainly the heterogeneity classes 1:n and C2C (personnel segment specification to [planned] participation type).

For incorporating the intermediate classes, the following heterogeneity classes are involved: (i) 1:n and R2C (intermediate class), (ii) 1:n and CR2R (reference from reciprocity/duality type to intermediate class), and (iii) 1:n and RC2R (reference from intermediate class to (planned) give/take participation type).

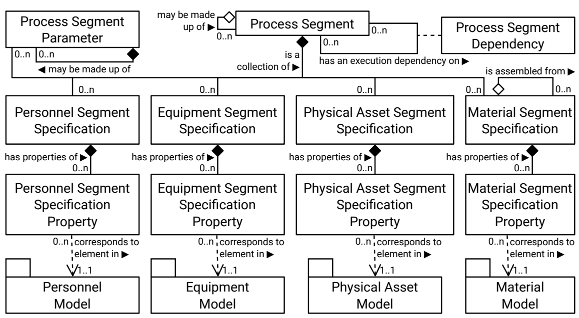

Fig. 3. ISA-95 process segment model (from [18]).

Equipment Segment Specifications define the required equipment resources to conduct a specific process segment.

Since the resource is an equipment, this corresponds to a *planned stockflow type* and a *stockflow type* that is referenced from a decremental *use* commitment/event type. The attributes `Equipment Class` and `Equipment` of class equipment segment specification that hold the IDs of the referenced equipment resources need to be resolved to the actual entities and are then aligned with the `resource` and `resourceType` references of class (planned) stockflow type.

Heterogeneity classes: 1:n and C2C. With regards to the intermediate "commitment type" or "event type" class, the same heterogeneity classes apply as specified in the personnel segment specification mapping.

Physical Asset Segment Specifications define the required physical assets to conduct the corresponding process segment.

Since the resource in question is a physical asset, this corresponds to a *planned stockflow type* and a *stockflow type* that is referenced from a decremental *use* commitment/event type, just as in the case of equipment segment specifications. The attributes `Physical Asset Class` and `Physical Asset` of class physical asset segment specification that hold the IDs of the referenced physical asset resources need to be resolved to the actual entities and are then aligned with the `resource` and `resourceType` references of class (planned) stockflow type.

Heterogeneity classes: 1:n and C2C. With regards to an intermediate "commitment type" or "event type" class, the same heterogeneity classes apply as specified in the personnel segment specification mapping.

Material Segment Specifications define the required material resources to conduct the corresponding process segment.

Since the resource is a material, this corresponds to a *planned stockflow type* and a *stockflow type* that is referenced from an incremental *produce* or a decremental *consume* commitment/event type. The kind of commitment/event type depends on the value of the ISA-95 "Material Use" property (allowed values are "Consumable", "Material Consumed", and "Material Produced"—the corresponding mapping is trivial). The attributes `Material Class` and `Material Definition` of class material segment specification that hold the IDs of the referenced material resources need to be resolved to the actual entities and are then aligned with the `resourceType` reference of class (planned) stockflow type.

Heterogeneity classes: 1:n and C2C. With regards to an intermediate "commitment type" or "event type" class, the same heterogeneity classes apply as specified in the personnel segment specification mapping.

B. Visualization of the Alignment

The alignment of metamodel elements of REA with corresponding metamodel elements of ISA-95 is depicted in Fig. 4 more extensively, including the operations management information models, except for the operations capability model.

The operations capability model is used to express capability information (such as committed, available, or unattainable) about personnel, equipment, physical asset, and material resources as well as process segments for a given slice of time (past, current, or future). It can be understood as a snapshot of the production system and is considerably important with respect to e.g., the generation of key performance indicators (for analytical applications) or production scheduling (for planning applications).

While it does not directly align with REA metamodel elements, operations capability information might be generated from REA fragments, such as from logged REA events for past slices of time. Vice versa, operations capability information—eg, sourced from the ERP or other IT systems—might be used to generate REA fragments, such as a REA schedule corresponding to a production schedule after production planning has been completed on the MES layer.

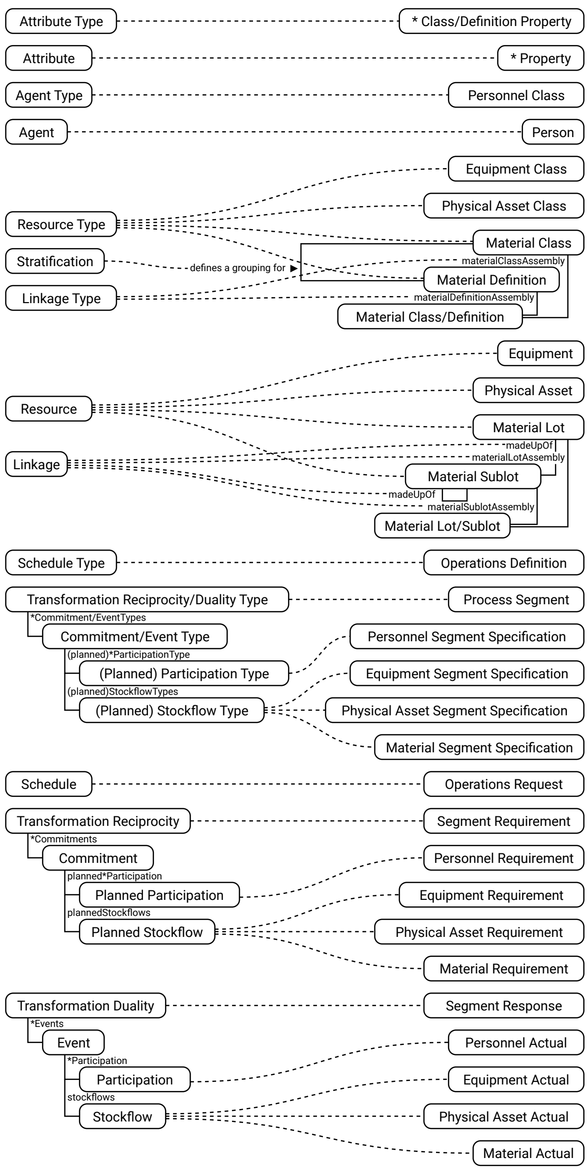

Fig. 4. Metamodel element alignment between REA and ISA-95: included is the high-level alignment of all ISA-95 metamodels except for "operations capabilities", because there is no direct equivalent in REA.

V. Application Scenario

A. Definition of the Initial Model

We provide an application scenario that represents a simplified production line of an office furniture manufacturer to discuss the main benefits of an entwined modeling view on REA and ISA-95 (cf. Fig. 5). In this scenario, the manufacturer hosts an ERP system (here abstracted through REA, similar to the one presented in [8]) that handles horizontal integration with business partners (customers and suppliers in this case), while an ISA-95 -aware MES drives the production processes required to manufacture the furniture. The production line under observation consists of five production steps: (i) **Cutting** creates a correctly sized piece of press board **P1** from a generically sized veneered press board **P** on a cutting machine **C**, (ii) **Drilling** creates a bored piece of press board **P2** on a leased drilling machine **D**, (iii) **Edge Banding** glues edging material **E** (e.g., veneer or plastic), on the bored piece of press board **P2** on an edge banding machine **B** and consequently produces a prepared piece of press board **P3** that is ready to be assembled, (iv) in a manual **Assembly** step, several components **Co** (e.g., table legs) are mounted on the prepared press boards, creating an assembled product **A**, and (v) quality assurance (**QA**) checks the product for flaws and in case everything is within valid parameters, provides a finished product **F** that is ready to be shipped. The money **M** earned by selling can in turn be used to buy more raw material, machines, consumables, etc. Each of the production steps could also produce a deficient product, subsumed as "waste" **W**.

The REA model has knowledge about the existence and cost of the material and machines involved, but it initially does not track the production processes at all, i.e., in the ERP system there is no information about when which production step has been realized in what time and which intermediate products have been on stock at a given point in time (e.g., *now*).

B. Implementation of the Mapping Approach

For an evaluation of the feasibility and usefulness of the presented approach, we choose the "Cutting" process of the application scenario as the testing subject of our implementation. Initially, the cutting process is defined in ISA-95 only, because the ERP system deals with entities of larger granularity with respect to the manufacturing processes. However, tighter integration with production facilities of customers requires the availability of production information within the ERP, because exchange of critical data with customers is realized only through electronic data interchange (EDI), due to a company policy. Therefore, the ERP system needs to keep track of these operations, which can be implemented as follows:

- Using model-driven techniques, we bootstrap the structure of the production process in the ERP system.
- Operations information is transformed into ERP information so that it can be easily stored in the ERP system.

For the implementation we are using the Eclipse Modeling Framework[3] (EMF) and the Epsilon[4] toolkit. ISA-95 and REA are both expressed in Ecore. The REA metamodel has been built manually, while the ISA-95 metamodel has been generated from the publicly available XML schema files of

[3]cf. https://www.eclipse.org/emf/

[4]cf. https://www.eclipse.org/epsilon/

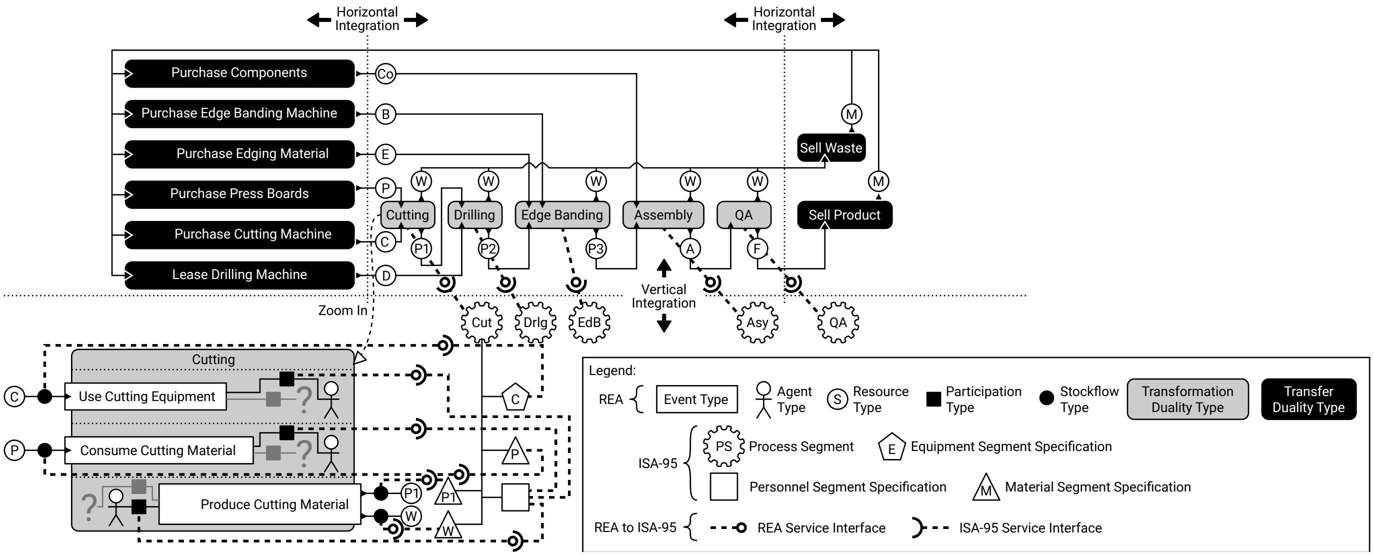

Fig. 5. Application scenario for the alignment of REA and ISA-95. The top compartment depicts REA structures, the middle compartment shows ISA-95 process segments and their mapping to REA concepts, the bottom compartment (shaded in light gray) visualizes the physical production process. The realization of the REA model of the manufacturing processes (gray shaded transformation duality types) is created from existing ISA-95 models in a semi-automatic manner, as described in Sec. V-B.

the business to manufacturing markup language[5] (B2MML), which is an XML based implementation of ISA-95.

In order to make the mapping between model elements of the different metamodels explicit, the following convention is used: a model element in REA defines an attribute with name `ISA-95 ID` that correlates with the `ID` attribute of a corresponding ISA-95 model element.

The bootstrapping of a REA model that corresponds to the process segment defined in ISA-95 is realized with the Epsilon Validation Language (EVL). An excerpt of the EVL code is depicted in Lst. 1, the underlying approach is as follows: for each "atomic" (line 6) process segment, the REA model is searched for a corresponding transformation duality/reciprocity type (lines 7–11). If none is found, a message is displayed (line 12–13) and the user can choose to create a stub model (lines 14–15) whose creation process is explained below:

1) Initiate (lines 17–18) a new transformation duality type (transformation reciprocity types are created analogously) and set the `ISA-95 ID` attribute to the `ID` of the process segment (lines 20–24).
2) In case only one personnel segment specification is defined in the process segment (lines 26-27), it is assumed that it refers to the machine operator. Thus, a corresponding agent type is searched in the REA model and stored in a variable to be used later (lines 28–33).
3) Check whether the process segment defines at least one equipment segment specification (lines 26–27), if so continue with (4) and (5). The physical asset and the material segment specifications are handled correspondingly but are not shown in Lst. 1 (cf. line 68).
4) ISA-95 has no equivalent to REA events, however in REA they are core features, so an "anonymous" event type needs to be created (lines 38–41). In the case of an equipment segment specification, the event type is added to the duality type as a *use* event type, since it will most likely resemble the usage of a machine (line 42). The same logic applies to physical assets. Material segment specifications, however, will trigger the creation of a *consume* or *produce* event type, depending on the value of the `Material Use` attribute: `Consumable` and `Material Consumed` lead to a consume event type, `Material Produced` a produce event type. If an agent type has been found in the previous step, create a participation type with that agent type and add it to the event type (lines 43–48).
5) Each of the equipment segment specifications (lines 49–50) maps to a stockflow type of the event type created in the previous step, therefore a stockflow type is created (line 54–56). Next, the correct resource type needs to be added: the available resource types of the REA model are searched for one that has a matching `ISA-95 ID` attribute value (lines 57–64).
   Note 1: in Lst. 1 only equipment *classes* are considered. Line 66 depicts that here similar code has been cut out that deals with equipment.
   Note 2: line 68 resembles the place where physical asset and material entities would be handled, but have been cut out due to space constraints.

The created **Cutting** transformation duality type is provided in the middle compartment of Fig. 5 in more detail: it consists of three event types that either use, consume, or produce a certain resource type through the given stockflow type. E.g., the event type **Use Cutting Equipment** uses a resource type

[5] cf. http://www.mesa.org/en/B2MML.asp

C through a corresponding stockflow type. In REA, each event type requires the definition of at least two participating agents or agent types: one is providing the resources of the event, the other one is receiving them. In terms of transformation events it could be that e.g., a manufacturing manager provides resources to a machine operator. It is likely, that the "providing" participant is not modeled in the MES—in that case, only the machine operator participation type is aligned between ISA-95 and REA. However, the other participation type *could* be added to the ISA-95 model to provide a more complete view on the production system's side.

The roles of the providing and receiving agents in intra-company transformations have not been settled on in REA—only recently it has been proposed to have only one participant (the operator) [38], which would then align very well with ISA-95. For that reason the second participation in the "cutting" duality type is depicted in gray. Another way to express the second participant in transformation dualities would be implicitly, through organizational relations (e.g., the line manager of the operator or the shift supervisor).

C. Evaluation and Critical Discussion

The presented approach relies on a modeling convention that has been implemented in the fictitious "Furniture, Inc." company: elements of the REA and the ISA-95 domain are explicitly linked to each other by implementing a `ISA-95 ID` string attribute in the REA elements. Other approaches, such as the one presented in [29], might be used in other companies. This linking process needs to be done in a pre-processing step before the algorithm presented in Sec. V-B can be executed.

Also, the generated REA elements might not be structured and named in a way that a human domain expert would have modeled it. One of the problematic points is the creation of event types, because generically either there is one event type for each segment specification, or the segment specifications are bundled together and expressed through different stockflow types (the approach implemented here). Manual improvement of the created REA model is therefore needed. The benefit of the presented generic approach still is that the domain expert can start with a structural skeleton instead of starting with a "blank sheet of paper". Also, further modeling conventions might help algorithms to generate better suited structures.

The presented result of the reconciliation approach, however, looks very promising. The created REA model is "readable" to the REA domain expert and can later be manually improved by restructuring event types and renaming entities.

VI. Conclusion

We have presented an alignment of a business application metamodel (REA) and an enterprise-control system integration metamodel (ISA-95). The usefulness and feasibility of this approach is evaluated by an application scenario in which techniques from model-driven engineering are used in order to synchronize and co-evolve models of these two domains.

The alignment contributes to resolving a number of issues in vertical integration scenarios:

- It helps keeping MES and ERP models in sync and enables a service interface for information exchange by providing a starting point for model-driven co-evolution and inter-model validation of ERP and MES systems.
- Production services are aligned with business services which supports data integrity and facilitates data analysis.
- A formally modeled system can support the model-driven definition and calculation of production-related (key) performance indicators.

Future work will explore the modeling of key performance indicators based on ISO 22400 and integration into the current testbed. ISO 22400 is particularly well suited since it partly relies on type definitions and modeling approaches of IEC 62264 [39]. This would enable the implementation of KPIs from within a model-driven environment.

Acknowledgment

This work was supported through the Christian Doppler Research Association by the Austrian Federal Ministry of Science, Research and Economy and the Austrian National Foundation for Research, Technology and Development.

References

[1] J. Gausemeier, F. Klocke, C. Dülme, D. Eckelt, P. Kabasci, M. Kohlhuber, N. Schön, and S. Schröder, "Industrie 4.0," Heinz Nixdorf Institut, Universität Paderborn, RWTH Aachen, Technical Report, 2016.

[2] M. Schleipen, "Adaptivität und semantische Interoperabilität von Manufacturing Execution Systemen (MES)," PhD Thesis, January 2012.

[3] J. Imtiaz and J. Jasperneite, "Scalability of OPC-UA down to the chip level enables "Internet of Things"," in *Proc. 11th IEEE Int. Conf. Industrial Informatics*, 2013, pp. 500–505.

[4] M. Witsch and B. Vogel-Heuser, "Towards a formal specification framework for manufacturing execution systems," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 311–320, May 2012.

[5] M. Witsch, "Funktionale Spezifikation von Manufacturing Execution Systems im Spannungsfeld zwischen IT, Geschäftsprozess und Produktion," Ph.D. dissertation, Technische Universität München, 2013.

[6] M. Brettel, N. Friederichsen, M. Keller, and M. Rosenberg, "How virtualization, decentralization and network building change the manufacturing landscape: an industry 4.0 perspective," *Int. J. of Mechanical, Industrial Science and Engineering*, vol. 8, no. 1, pp. 37–44, 2014.

[7] J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. de Amicis, E. B. Pinto, P. Eisert, J. Döllner, and I. Vallarino, "Visual computing as a key enabling technology for industrie 4.0 and industrial internet," *IEEE computer graphics and applic.*, vol. 35, no. 2, pp. 26–40, 2015.

[8] B. Wally, A. Mazak, B. Kratzwald, and C. Huemer, "Model-driven retail information system based on REA business ontology and Retail-H," in *Proc. 17th IEEE Conf. Business Informatics*, vol. 1, 2015, pp. 116–124.

[9] 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, July 1982.

[10] G. L. Geerts and W. E. McCarthy, "Modeling business enterprises as value-added process hierarchies with resource-event-agent object templates," in *Business Object Design and Implementation*, 1997.

[11] ——, "An accounting object infrastructure for knowledge-based enterprise models," *IEEE Intelligent Systems*, vol. 14, no. 4, pp. 89–94, 1999.

[12] ——, "The ontological foundation of REA enterprise information systems," in *Annual Meeting of the AAA*, vol. 362, 2000, pp. 127–150.

[13] ——, "Policy-level specifications in REA enterprise information systems," *J. of Information Systems*, vol. 20, no. 2, pp. 37–63, Fall 2006.

[14] P. Coad, "Object-oriented patterns," *Communications of the ACM*, vol. 35, no. 9, pp. 152–159, September 1992.

[15] R. Johnson and B. Woolf, "Type object," in *Pattern Languages of Program Design 3*, 1997, ch. Type Object, pp. 47–65.

[16] Object Management Group, Inc., *OMG Unified Modeling Language (OMG UML)*, Specification, Rev. 2.5, March 2015.

[17] G. L. Geerts and W. E. McCarthy, “An ontological analysis of the economic primitives of the extended-REA enterprise information architecture,” *Int. J. of Accounting Information Systems*, vol. 3, no. 1, 2002.

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

[19] ——, *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.

[20] A. Mazak and C. Huemer, “From business functions to control functions: Transforming REA to ISA-95,” in *Proc. 17th IEEE Conf. Business Informatics*, vol. 1. IEEE, July 2015, pp. 33–42.

[21] ——, “HoVer: A modeling framework for horizontal and vertical integration,” in *Proc. 13th IEEE Int. Conf. Industrial Informatics*, 2015.

[22] B. Wally, C. Huemer, and A. Mazak, “A view on model-driven vertical integration: Alignment of production facility models and business models,” in *Proc. IEEE Int. Conf. Autom. Science and Engineering*, 2017.

[23] ——, “ISA-95 based task specification layer for REA in production environments,” in *Proc. 11th Int. Workshop Value Modeling and Business Ontologies*, 2017.

[24] International Electrotechnical Commission (IEC), *Enterprise-control system integration—Part 4: Object model attributes for manufacturing operations management integration*, International Standard, Rev. 1.0, December 2015, IEC 62264-4:2015.

[25] B. Wally, C. Huemer, and A. Mazak, “Entwining plant engineering data and ERP information: Vertical integration with AutomationML and ISA-95,” in *Proc. IEEE Int. Conf. Control, Automation and Robotics*, 2017.

[26] B. Wally, M. Schleipen, N. Schmidt, N. D’Agostino, R. Henßen, and Y. Hua, “AutomationML auf höheren Automatisierungsebenen,” in *Proc. AUTOMATION 2017*, 2017.

[27] B. Vogel-Heuser, S. Feldmann, J. Folmer, J. Ladiges, A. Fay, S. Lity, M. Tichy, M. Kowal, I. Schaefer, C. Haubeck, W. Lamersdorf, T. Kehrer, S. Getir, M. Ulbrich, V. Klebanov, and B. Beckert, “Selected challenges of software evolution for automated production systems,” in *Proc. IEEE Int. Conf. Industrial Informatics*, 2015, pp. 314–321.

[28] L. Berardinelli, S. Biffl, A. Lüder, E. Mätzler, T. Mayerhofer, M. Wimmer, and S. Wolny, “Cross-disciplinary engineering with AutomationML and SysML,” *at - Automatisierungstechnik*, vol. 64, no. 4, 2016.

[29] S. Feldmann, M. Wimmer, K. Kernschmidt, and B. Vogel-Heuser, “A comprehensive approach for managing inter-model inconsistencies in automated production systems engineering,” in *Proc. 12th IEEE Int. Conf. Automation Science and Engineering*, 2016, pp. 1120–1127.

[30] A. Theorin, K. Bengtsson, J. Provost, M. Lieder, C. Johnsson, T. Lundholm, and B. Lennartson, “An event-driven manufacturing information system architecture,” in *Proc. 15th IFAC Symp. Information Control Problems in Manufacturing*, 2015, pp. 547–554.

[31] ——, “An event-driven manufacturing information system architecture for industry 4.0,” *Int. J. of Production Research*, vol. 55, no. 5, 2017.

[32] M. Wimmer, G. Kappel, A. Kusel, W. Retschitzegger, J. Schoenboeck, and W. Schwinger, “Towards an expressivity benchmark for mappings based on a systematic classification of heterogeneities,” in *Proc. 1st Int. Workshop on Model-Driven Interoperability*. ACM, 2010, pp. 32–41.

[33] F. Scharffe and D. Fensel, “Correspondence patterns for ontology alignment,” in *Proc. 16th Int. Conf. Knowledge Engineering and Knowledge Management*, ser. LNAI, no. 5268, 2008, pp. 83–92.

[34] H. O. Unver, “An ISA-95-based manufacturing intelligence system in support of lean initiatives,” *Int. J. of Advanced Manufacturing Technology*, vol. 65, no. 5, pp. 853–866, 2013.

[35] OPC Foundation, *OPC Unified Architecture for ISA-95 Common Object Model*, Std., Rev. 1.00, October 2013, OPC-ISA-95:2013.

[36] International Electrotechnical Commission (IEC), *Automation markup language—Part 2: Role class libraries*, International Standard, Rev. 1.0, March 2015, IEC 62714-2:2015.

[37] ——, *Automation markup language—Part 1: Architecture and general requirements*, International Standard, Rev. 1.0, June 2014, IEC 62714-1:2014.

[38] W. E. McCarthy, G. L. Geerts, and G. Gal, “The economic structures of exchanges vs. conversions in the REA enterprise ontology,” 2016, presented at 10th Int. Workshop Value Modeling and Business Ontology.

[39] International Organization for Standardization (ISO), *Automation systems and integration—Key performance indicators (KPIs) for manufacturing operations management—Part 2: Definitions and descriptions*, International Standard, Rev. 1.0, January 2014, ISO 22400-2:2014.

Lst. 1. Excerpt of a cross-model validation between REA and ISA-95, expressed in EVL (transformation duality type compared to process segment).

```
pre {
  var rea = REA!Model.all.first();
  var isaId = "ISA-95 ID"; }
context ISA!ProcessSegmentType {
 constraint TransformationDualityTypeExists {
   guard: self.processSegment.isEmpty()
   check : REA!TransformationDualityType.all
     .exists( tdt | tdt.attributes.exists( a |
       a.key = isaId ) and tdt.attributes
       .selectOne( a | a.key = isaId ).value
       .stringValue = self.iD.value )
   message : "No Transformation Duality Type" +
     " found for \"" + self.iD.value + "\""
   fix {
     title : "Add missing Transf. Duality Type"
     do {
       var tdt = new REA!
         TransformationDualityType;
       tdt.name = self.iD.value;
       var idAttr= new REA!StringToAttributeMap;
       idAttr.key = isaId;
       idAttr.value = new REA!Attribute;
       idAttr.value.stringValue = self.iD.value;
       tdt.attributes.add( idAttr );
       var at = null;
       if( self.personnelSegmentSpecification
         .size() = 1 ) {
         at = rea.agentTypes.selectOne( at | at
           .attributes.exists(a | a.key = isaId)
           and at.attributes.selectOne( a |
           a.key = isaId ).value.stringValue
           = self.personnelSegmentSpecification
           .first().personnelClassID.value);
       }
       var et = null;
       if( self.equipmentSegmentSpecification
         .size() > 0 ) {
         et = new REA!EventType;
         et.name = "Use " + self.iD.value +
           " Equipment";
         rea.eventTypes.add( et );
         tdt.useEventTypes.add( et );
         if( at <> null ) {
           var pt = new REA!ParticipationType;
           pt.name = "Receive " + self.iD.value
             + " Equipment";
           pt.agentType = at;
           et.receiveParticipationType = pt; }
         for( es in self
           .equipmentSegmentSpecification ) {
           if( (es.equipmentClassID <> null)
             and( es.equipmentClassID
             .value <> null) ) {
             var sft = new REA!StockflowType;
             sft.name = "Use " + es
               .equipmentClassID.value;
             var rt = rea.resourceTypes
               .selectOne( rt | rt.attributes
               .exists( a | a.key = isaId )
               and rt.attributes.selectOne( a
               | a.key= isaId).value.stringValue
               = es.equipmentClassID.value );
             if( rt <> null ) {
               sft.resourceType = rt; }
             et.stockflowTypes.add( sft ); }
         //SOCA: Skipping Equipment
         } }
       //SOCA: Skipping Phy. Assets and Material
       rea.transformationDualityTypes
         .add( tdt ); } } } }
```