From business functions to control functions: Transforming REA to ISA-95

Alexandra Mazak
TU Vienna
Austria, Vienna
Email: mazak@big.tuwien.ac.at

Christian Huemer
TU Vienna
Austria, Vienna
Email: huemer@big.tuwien.ac.at

***Abstract*—In the context of smart factories, a seamless information exchange between information systems on the same layer (horizontal integration) and between information systems on different layers (vertical integration) is a key issue. For this purpose we aim for an integrated modeling framework spanning over production chains and value networks. In building this framework, we first concentrate on the layers realizing the business functions and the manufacturing control functions. Thereby, we build up on the Resource Event Agent (REA) business ontology (ISO/IEC 15944-4) to describe external activities requiring horizontal integration with business partners and internal activities serving as a hook for vertical integration within a manufacturing enterprise. Furthermore, we base our framework on the ISA-95 industry standard (ANSI/ISA-95; IEC 62264) to describe the vertical integration within an enterprise. In this paper, we demonstrate how information given in REA models is transformed to corresponding ISA-95 skeletons. In other words, we show how a model describing the main business functions of an enterprise is used to derive essential concepts relevant to the manufacturing execution system.**

I. Introduction

The German working committee for Industrie 4.0[1] has identified among others the following research issues [1]:

- horizontal integration through value networks
- vertical integration of networked manufacturing systems
- end-to-end digital integration of engineering across the entire value chain

Industrie 4.0 use case scenarios relating, e.g., to networked manufacturing, self-organizing adaptive logistics, and customer-integrated engineering will require business models that will primarily be implemented by what could be a highly dynamic network of businesses rather than by a single company (e.g., to link products of a manufacturing company with appropriate services provided by another company) [1]. On the one hand—for realizing a *horizontal integration through value networks*—we need appropriate language constructs to describe business relationships between companies also taking their different business views into account. On the other hand—to enable a seamless *vertical integration of networked manufacturing systems*—we need a fundamental understanding of activities and information flows within manufacturing companies. However, information flows between the horizontal layer (business partner networks) and the vertical layer (from an ERP system to a Manufacturing Execution System) are very limited or not even possible at all [1]. IT systems still tend not to cross company or factory boundaries. The German initiative for Industrie 4.0 points out that the use of information technology in this context has largely failed to reflect the existence of manufacturing networks. One problem is, among others, that value chains (from customer requirements to production and distribution) tend to be relatively static since they often have been created over many years.

From a technical as well as an economic perspective an *end-to-end digital integration* will be a key issue to realize smart factories. This integration will enable all parts of a manufacturing company (enterprise level, shop floor control level, and shop floor level) to be connected to each other through a global information system with customers, suppliers, and other external participating parties. The potential of an end-to-end integration is huge. For example, this will allow in future to individual, customer-specific criteria to be included in the design, configuration, ordering, planning, manufacture and operation phase. This will enable last-minute changes to be incorporated and very low production volumes (batch size of 1). The realization of this ambitious goal requires appropriate interfaces for integrating the individual subsystems [2]. However, it is still common practice that IT systems exchange information through extensive interfaces, but can only utilize specific pieces of that information. The situation is further worsened by the problem that many different interfaces introduce dependencies whose management can become complex and hard to achieve. Thus, the system complexity will rise drastically.

There is still a lack of appropriate concepts for interface integration by which different operational layers can be connected for communication. However, to provide a universal infrastructure for a seamless information exchange is crucial for a successful implementation of the Industrie 4.0 initiative. Modeling can act as an enabler for managing this integration. Models are representations of real and hypothetical scenarios that only include those aspects that are relevant to the issue under consideration. The working group of the German initiative points to the fact that *"the use of models constitutes an important strategy in the digital world and is of central importance in the context of Industrie 4.0"* [1]. For this purpose appropriate language constructs are required to formally describe the increasing functionality, increasing

[1]Please note, that the approach introduced in this paper is aligned with the German initiative "Industrie 4.0", and therefore, we do not translate it to the English term "Industry".

product customization, dynamic delivery requirements, and the rapidly changing forms of cooperation between different companies in order to provide end-to-end transparency.

II. Approach

The approach presented in this paper is based in its orientation on the recommendations of the German working committee for Industrie 4.0 which was released in 2013. Amongst other things, the working committee points out that production systems are to be linked vertically with business processes within decentralized production sites and enterprises, and that they are to be distributed horizontally among suppliers, distributors and customers. In order to meet these requirements, we aim for an *integrated modeling framework* spanning over the horizontal layer (value networks) and the vertical layer (production chains). For this purpose we do not intend to start from scratch by defining our own all-encompassing modeling language. In contrary, we want to build up on existing well-accepted modeling languages.

The German working group defines the *vertical integration* as *"the integration of the various IT systems at the different hierarchical levels (e.g., the actuator and sensor, control, production management, manufacturing and execution and corporate planning levels in order to deliver end-to-end solution"*, [1]. We consider the concepts and models of the industry standard *ISA-95* (ANSI/ISA-95; IEC 62264) [3], [4] as appropriate to model the vertical integration of information flows between the different levels within an enterprise. ISA-95 is an international standard released by the *International Society of Automation* for developing an automated interface between *Enterprise Resource Planning Systems (ERP)* on the enterprise level and *Manufacturing Excecution Systems (MES)* on the shop floor (control) level. Based upon this standard, which consists of five parts, the standard IEC 62264 was established.

In analogy, we apply concepts of the *Resource-Event-Agent business ontology (REA)* (ISO 15944-4) [5] which allows describing the interfaces between the systems of different business partners as a horizontal integration of information flows. The German working group defines the *horizontal integration* as referring to *"the integration of the various IT systems used in the different stages of the manufacturing and business planning processes that involve an exchange of materials, energy and information both within a company (e.g. inbound logistics, production, outbound logistics, marketing) and between several different companies (value networks)"*, [1]. In a business environment, REA is used to identify the value adding activities of the company. In general, value adding activities are either *transformations* of resources by producing something or *transfers* of resources by exchanging something with an external party. In other words, REA is able to provide the binding clue between the internal production processes requiring vertical integration and the external trading activities requiring horizontal integration.

In our approach, we elaborate on a seamless integration of the horizontal and vertical layers which implies that necessary information must flow between these layers. For realizing a vertical as well as a horizontal integration through value networks appropriate language constructs are needed

Fig. 1. ISA-95: Functional Hierarchy model [IEC 62264-1]

to describe interface integration within the company between different kinds of IT systems (ERP, MES) at different levels and between multiple enterprises and various participating parties (vendors, sub-contractors, customers). For this purpose we transform REA concepts to ISA-95 concepts. Our approach is independent of any software solution. In fact, companies may use different ERP and MES systems and still have to collaborate with each other.

Following our aim of an integrated modeling framework by transforming concepts of REA to concepts and models of ISA-95, we concentrate on these two standards in Section III on related work. Section IV presents the REA meta model and its core concepts. In Section V, we present the ISA-95 meta model. Section VI provides the core of our paper describing the transformation rules from REA to ISA-95. This transformation is illustrated by examples in Section VII and Section VII-B. We close the paper with a summary of our contribution in Section VIII.

III. Related Work

A. Industry Standard ISA-95

The ISA-95 standard has been developed for global manufacturers, i.e., a production company with decentralized, networked production plants. This standard fosters a universal communication within a manufacturing company (headquarters and distributed industrial premises). ISA-95 can be applied in all industries, and in all sorts of production processes like batch processes, continuous processes, and repetitive processes. ISA-95 was specifically developed for creating interfaces between the enterprise domain with its ERP system at Level 4 and the shop floor control domain with its MES at Level 3 and lower (Levels 2, 1, 0). It offers a fundamental understanding of activities and information flows within a manufacturing company. The standard describes hierarchy models which are based on the *Purdue Enterprise Reference Architecture (PERA)* for Computer Integrated Manufacturing (CIM) [6].

Figure 1 shows in a simplified manner the different levels of the *functional hierarchy model*. In addition, the equipment (e.g., site, area, process cell, production line, storage zone) are usually organized in a hierarchical fashion. The red cycle in Figure 1 shows the *enterprise-control interface* between Level 4 and Level 3. Between these levels the standard points to 31 information flows, as outlined in Figure 2. The wide dotted line of this *functional enterprise control model* illustrates the boundary of the enterprise-control interface. Everything that lies outside the dotted lines belongs to Level 4, and everything that lies inside the dotted lines belongs to Level 3. The

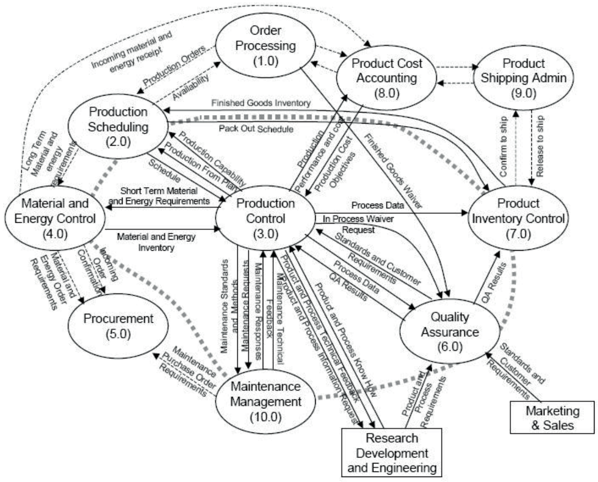

Fig. 2. ISA-95: Functional Enterprise-control model [IEC 62264-1]

labeled lines indicate the 31 information flows of importance to manufacturing control. The model contains 12 functions [3].

ISA-95 describes step by step the tasks of each of these functions. The functions shown in rectangles (e.g., research, development and engineering, marketing, sales) are external entities and as such they are not described in the functional enterprise control model. These entities are components outside the boundaries of this model that send data and receive data from the functions. The basic data to be exchanged in this model are information flows which are defined by ISA-95 for the sectors *personnel, material, equipment, physical asset* and *process segment*. The process segment is a logical group of equipment, physical asset, personnel, material required to carry out a specific part of a process (e.g, mixing, sawing, etc.). These sectors are defined as *object models* in ISA-95 which constitute basic building blocks with which the information flows of the functional hierarchy model are constructed (cf. Figure 1). In order to standardize the 31 information flows between Level 4 and Level 3 ISA-95 groups them into four categories: (i) production capability information, (ii) production definition information, (iii) production schedule information, and (iv) production performance information [3].

B. *Resource-Event-Agent Business Ontology*

The Resource-Event-Agent business ontology (REA) was developed by William McCarthy [7] for the application-independent description of *economic phenomena* (i.e., exchanges which can either be transfers or transformations of resources). The acronym REA stands for the three main concepts of the ontology *Resource*, *Event*, and *Agent*. Agents are persons, companies, or organizational units capable of having control over resources, who/which participate in an *economic exchange*. Resources are transferred or transformed during an economic exchange. Resources can be goods, material, rights, labor, equipment, physical assets or services which agents have control of and which should be monitored and controlled in a business environment. An event is considered as a class of phenomena reflecting exchanges of resources. REA has its roots in the accounting discipline and is based on strong concepts of the literature in economic theory [8]. Additionally, REA focuses on IT implementation issues and follows a conceptual modeling approach [9]. This makes it a good choice for being used in a business model-driven engineering approach. Moreover, the REA business ontology is a wide accepted language in the academic world to design enterprise information systems. For instance, in the ISO/IEC 15944-4 Open-edi standard [5]—which addresses business communications between enterprises—REA is used as an ontological framework for specifying concepts and relationships involved in business transactions and scenarios. REA initially focuses on concepts of economic exchanges of the present and the past.

IV. The REA Meta Model

In this section, we elaborate on the REA meta model. Thereby, we build up on previous work [10], [11], [12]. In these papers we developed a domain specific language (DSL) for the REA ontology called REA-DSL. The REA-DSL provides a formal definition of the REA language concepts by means of Object Management Group's (OMG) meta-modeling architecture called Meta-Object Facility (MOF) [13]. MOF comes with a meta-meta model (M3 layer) that allows us to define the REA concepts as a meta-model (M2 layer). In this section, we introduce the existing REA concepts by means of meta-models and also show some additional extensions required for this work.

REA consists of three different layers concerning entrepreneurial logic and details at a different level of granularity. The three layers from top down are:

1) value chain specification layer
2) duality specification layer
3) task specification layer

In the following subsections, we explain the meta-models of these REA layers.

A. *REA Value Chain*

A business model defines how a company creates value. It specifies a competitive strategy by looking at those activities that create value for the company. A seminal work in this respect has been Michael E. Porter's book "Competitive Advantage" [14] in which he first introduces the concept of the value chain. A *value chain* is a set of activities that an organization carries out to create value. Porter proposes the concept of a value chain to examine all of a company's activities, and see how these are connected.

The *REA value chain* is based on Porter's definition. It is built by a number of value activities. A *value activity* takes some resources as input and creates some resources as output. From an economic perspective it is important that the output is considered to be of higher value than the input. On a high level of abstraction there are two ways to create additional value by an activity: firstly, one may use and/or consume some input resources in order to produce some output (e.g., a finished good),—this is called a *transformation* in REA. Secondly, in a trading relationship with external business partners one may receive resources (e.g., material, equipment, transport service, etc.) and give resources (e.g., cash) in return,—this is called a *transfer* in REA.

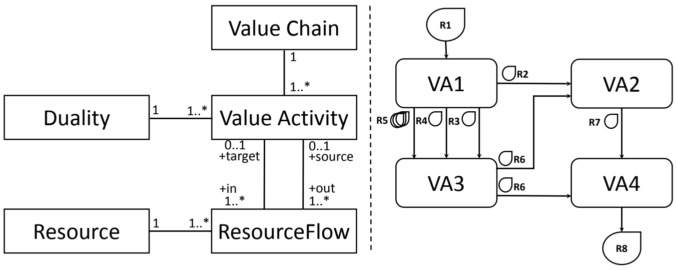

Fig. 3. REA: Value Chain Meta Model and its Instantiation

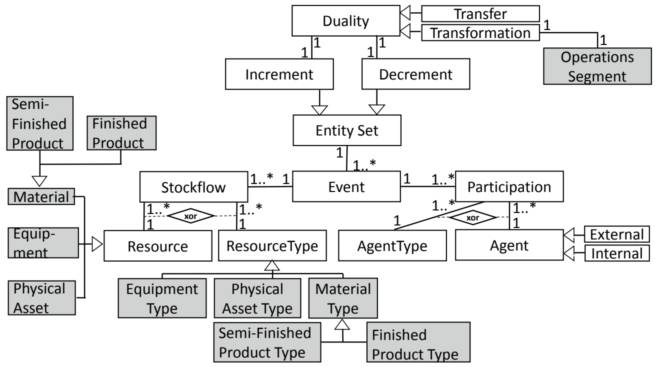

Fig. 4. REA: Duality Meta-Model

Furthermore, REA is built on the economic principle that any output by one value activity serves as input to another value activity. It follows that it is the resources which connect the different value activities. Thus, a REA value chain contains a number of value activities and specifies the resource flows amongst them—nothing more, nothing else [15]. More details are available on the second layer—the duality specification layer—where we find duality models for each of the value activities (cf. Figure 4).

The left hand side of Figure 3 presents the meta-model of the REA value chain. A value chain includes one to many value activities that are depicted by rectangles with rounded corners (cf. right hand side of Figure 3). A value activity is used only once in one distinctive value chain. A value activity points to exactly one duality (described in the next subsection). A *duality* is usually the basis of one value activity, but may be referred to by multiple value activities.

Resource flows tie the value activities together. A resource flow is a directed association that usually starts from a source value activity and ends at a target value activity (cf. right hand side of Figure 3). When analyzing a whole company, there is in theory no final output and no input that is not based on an output of another value activity. For the purpose of a partial analysis, we permit resource flows that have either no source value activity or no target value activity. It follows that a value activity has at least one, but up to many *outgoing* resource flows. Similarly, a value activity has at least one, but up to many *ingoing* resource flows. Each resource flow points to exactly one resource. This resource is depicted by the symbol of a drop next to the directed arc of the information flow. A resource may be included in many resource flows. The right hand side of Figure 3 shows an abstract example model of a value chain which is a valid instance of the meta-model on the left hand side.

B. REA Duality

In the previous subsection, we learned that value activities receive some input resources to create output recources of higher value. Each value activity is further detailed by a duality on the second REA layer. A *duality* is a core economic principle that says that it is impossible to get something for nothing ("there is no free lunch"). Accordingly, a duality consists of two parts: the *decrement entity set* covers events executed by some agents leading to a decrease of some resources. It is compensated by the *increment entity set* that covers events executed by some agents leading to an increment of some (other) resources. By definition the increment in resources is considered of higher value than the decrement in resources. Again, the duality concept applies to transfers (exchanges with external agents) and transformations (value creation inside the enterprise).

Figure 4 shows the meta-model for a duality. The meta classes with white background describe the existing REA concepts, the ones with gray background represent our proposed extensions described further below. A *duality* has two specializations: a *transfer* and a *transformation*. Independent of the specialization a duality is composed of exactly one *increment entity set* and one *decrement entity set*. Both are specializations of the general entity set. Each entity set is represented in a specific swimlane (cf. Figure 5). According to the REA meta-model, an entity set covers at least one but up to multiple events. An *event*—depicted as a hexagon—is specific to the entity set it belongs to (cf. Figure 5). Following the principles of duality, all events in the decrement entity set (give/consume/use) are counterbalanced by the events in the corresponding increment entity set (take/produce) of the same duality (cf. Figure 4).

The relationship between an event and a resource is described by the concept of *stockflow* [15]. A stockflow is represented as a directed arc between exactly one event (hexagon) and one resource (drop) (cf. Figure 5). In the increment set the direction of the arc goes from the resource to the event, in the decrement set in the reverse direction. An event will affect most of the time one resource only, but it may affect multiple ones. Thus, an event may have one up to many stockflows connected. A resource usually is affected by many different events (in different entity sets of different duality models). At a minimum a resource is affected by one event—otherwise it would not be worth considering the resource at all. Consequently, a resource is connected to one up to many stockflows.

In REA, resources can be goods, material, rights, labor, equipment, physical assets, or services. REA does not make any particular differentiation and all of these resources are denoted by the icon of a drop (cf. Figure 5). Due to its dedicated focus on the production domain, ISA-95 differentiates between *material*, *equipment*, and *physical asset* as special kinds of resources. When aiming for an integrated approach the differentiation of these special resources should be reflected in the REA ontology as well. Accordingly, we define *material*, *equipment*, and *physical asset* as specializations of the REA

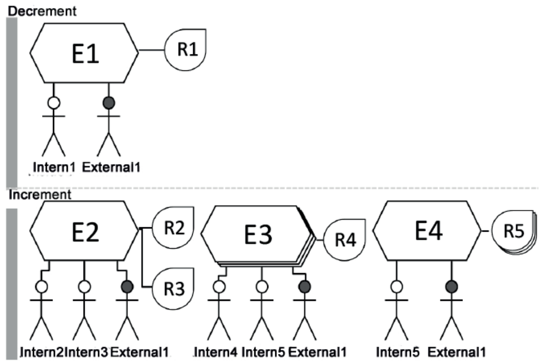

Fig. 5. Duality Example

resource (see classes with gray background in Figure 4). In addition, we define specializations for the corresponding typification concepts, i.e., *material type*, *equipment type*, and *physical asset type* are defined as specializations of the REA resource type. We also define dedicated icons for them. A material is denoted by a cuboid, an equipment by a white gear wheel, and a physical asset by a gray gear wheel. All of these specializations may also be used whenever a resource is expected in REA, i.e., as part of a duality and as classifiers assigned to resource flows. In addition, we introduce two specializations of the (non-abstract) material concept, i.e., *semi-finished product* presented by a white cube and *finished product* presented by a gray cube.

An event involves *agents* depicted as stickfigures. We distinguish between *external agents* (denoted with black heads), e.g., trading partners outside the company, and *internal agents* (denoted with white heads), who are accountable inside the company (cf. Figure 5). The involvement of agents in events is denoted by the concept of *participation*. A participation is an undirected association that connects exactly one event with one agent. An event is associated to at least one, but up to many agents. Hence, an event has one to many participation associations. An agent participates in at least one, but up to many events (in the same, but also in different entity sets of the same or different dualities). Thus, an agent has one to many participations connected. In addition, there are further constraints assigned to the meta-model to handle specifics of transfers. In case of a transfer, each event must be assigned to exactly one outside agent and, in addition, to at least one inside agent [16]. All events of the same transfer (both in the decrement and the increment entity set) must involve one and the same outside agent. Additionally, REA provides concepts for the *typification* of resources and agents [17]. *Resource types* and *agent types* display a small `T` in their icon. It should be noted that due to space limitations, we do not elaborate on the details of event series, resource series and agent series, which are denoted by a staple of hexagons/drops/stickfigures. The interested reader is referred to the paper of Sonnenberg et al. [10]. Figure 5 shows an abstract example model of the REA concept duality which is a valid instance of the meta-model presented in Figure 4.

C. REA task specification layer

In the first two subsections, we elaborated on the top two layers of REA (value chain specification layer and duality specification layer). One may expect that we do the same for the third layer—the *task specification layer*—describing the process to transform the input to the output as defined in the layers above. However, the REA literature does not concentrate on the task specification layer, instead it suggests to use activity diagrams or state machines to describe the task specification layer. REA does not provide any language concepts for linking identified tasks with agents, resources, etc. Accordingly, one may consider either extending the REA ontology for this purpose or specifying transformations to another language. In the context of the production domain, we are confident that ISA-95 is a perfect candidate langauge for the latter case. Accordingly, we propose that each REA duality model points to exactly one ISA-95 *operations segment* (see upper right corner of Figure 4). The relevant ISA-95 meta models with respect to an operations definition are described in the following section.

V. THE ISA-95 META MODEL

Production operations are defined by the ISA-95 *operations definition model* that is depicted in key parts in Figure 6. An *operations definition* represents the resources required to perform a specified operation. The operations definition references a *work definition*, which defines the information used to instruct a manufacturing operation (i.e., how to perform the operation) [18]. An operations definition is associated to one to many *operations segments*. *Operations segments* may be recursively structured. An *operations segment* encapsulates the information needed to quantify a segment for a specific operation. It corresponds to one to many *process segments* [18]. Process segments are the smallest elements of manufacturing activities that are visible to business processes.

An operations segment provides a logical grouping of personnel resources, equipment resources, physical asset resources, and material required to perform a specific operations segment. Consequently, it includes different kinds of resource specifications: *personnel specifications*, *material specifications*, *equipment specifications*, and *physical asset specifications* [4]. These resource specifications identify the

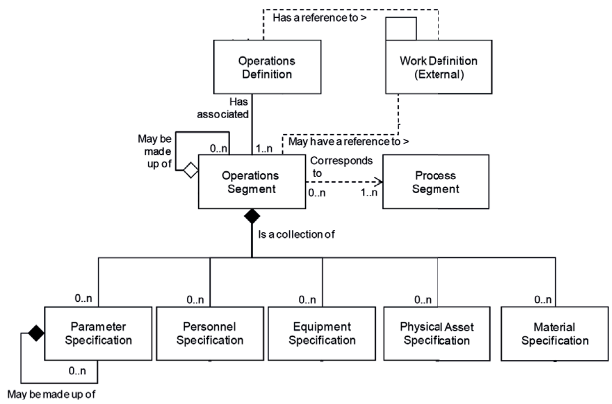

Fig. 6. ISA-95: Part of the Operations Definition Information Model [IEC 62264-2]

resource types and/or concrete resources, their quantity and the unit of measure of the quantity needed to perform an operations segment. For instance, to perform a frame production, we need—amongst other things—a specified quantity of a certain material (e.g., carbon crossbar) or material type (e.g., crossbar). In addition, the operations segment may include one to many *parameter specifications* containing the names and types of values that may be sent to the manufacturing execution systems at Level 3 to parametrize an operation.

Each of the above mentioned resource specifications within an operations segment are references to corresponding ISA-95 models [4]. The *material model* defines the actual materials, material definitions, and information about classes of material definitions. Material information includes the inventory of raw, finished, intermediate materials, and consumables [18]. The role-based *equipment model* contains information about specific equipment, the equipment class, and their particular properties. Role-based means that the equipment model is used to construct hierarchy models used in manufacturing scenarios (enterprise, site, area, work center, work units, process cells, etc) [18]. Due to this role-based view the equipment model is related to the *physical asset model* [4]. This model contains information about the physical piece within the manufacturing enterprise, i.e., a specific equipment. The *personnel model* contains information about specific personnel (class Person), classes of personnel (class Personnel Class) as well as their properties [4].

Accordingly, the schemata of these resource models are very similar and we do not detail all of them due to space limitations. We pick the material model as a typical representative of the resource models and present it in Figure 7. A *material class* may be defined as containing an assembly of material classes and as part of an assembly of material classes. A material class is a grouping of material definitions for an operations definition. A material class may define zero or more *material class properties*. Material class properties may contain nested material class properties. These properties often list the nominal, or standard values for the material (e.g., pH factor, material strength). A material property does not have to match material class properties. A *material definition* shall belong to zero or more material classes. Similar to material class, a material definition may be defined as containing an assembly of material definitions and as part of an assembly of material definitions. For a detailed description of the ISA-95 resource models, we refer the interested reader to the

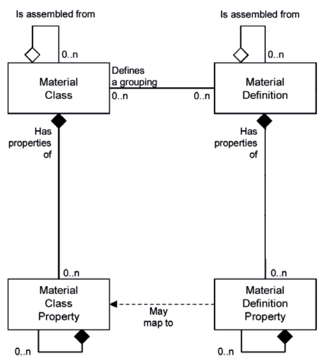

Fig. 7. ISA-95: Material Model [IEC 62264-2]

standard[4].

VI. Transformation Rules from REA to ISA-95

In the previous two sections, we described the REA meta model and the relevant parts of the ISA-95 meta model. In our integrated modeling framework, we intend to use REA for the purpose of modeling the main business functions of an enterprise. These business functions reside on Level 4 of the functional hierarchy model as depicted in Figure 1. In REA, one may distinguish business functions that require the exchange of resources with business partners, i.e., REA-transfers, and business functions that require the transformation of resources and are executed within the enterprise, i.e., REA-transformations. In an industry context, the latter ones are typically the production processes. Evidently, information about business functions describing transformations should be passed to the control functions at Level 3 of the functional hierarchy model. Accordingly, the relevant information in REA models has to be transformed to ISA-95 in order to realize the upper part of our intended integrated modeling framework. In this section, we describe the corresponding transformation rules which are depicted in Figure 8.

Each REA *duality* describing a *transformation* is transformed to an ISA-95 *operations segment* (A). The *name* of the *duality* becomes the *operations segment* ID (A1). Alternatively, one may decide to use logical, system generated identifiers, in which case a REA *duality ID* would map to the *operations segment* ID and the *name* of the *duality* to the *operations segment description*. In this paper, we have opted for "‘readable"’ IDs, also for other concepts described further below. The REA duality also links to a corresponding process definition which is carried forward to the *process segment* ID referenced by the operations segment (A2). It should be noted that each REA duality model leads to exactly one operations segment. In case that this operations segment is not fine granualar enough for control functions, one may re-work the operations segment in ISA-95 to create nested operations segments within it.

In the next steps, we have to transform the input resources for operations segments, which are personnel (B), equipment (C), physical assets (D), and materials (E). In REA the input side is described within the decrement entity set. Accordingly, calculating the input requires to access all events within the decrement entity set of a duality. The input then corresponds to the REA agents connected by *participation* associations to these events and the REA resources connected by *stockflow* associations.

It follows that each *agent* or *agent type* connected to a *decrement event* leads to a *personnel specification* within the *operations segment* (B). In the case of an *agent type* its *name* is mapped to the *personnel class ID* (B1). Whereas the *name* of a specific *agent* maps to the *person ID* of the *personnel specification* (B2). The *participation* association between an *event* and an *agent* has by default an attribute *quantity*, i.e. the number of *agent* (*types*) involved. This *quantity* is mapped to the *personnel specification quantity* (B3).

An *equipment* or *equipment type* connceted to a *decrement event* results in an *equipment specification* as part of the *operations segment* (C). In case of an *equipment type* its *name* maps to the *equipment class ID* (C1). Whereas the *name* of a specific

A	Duality ⇒ Operations Segment
A1	Duality.Name ⇒ OperationsSegment.ID
A2	Duality.ProcessDefinition ⇒ OperationsSegment->ProcessSegment.ID
B	Duality->Decrement->Event->Participation->Agent/AgentType ⇒ OperationsSegment->PersonnelSpecification
B1	Duality->Decrement->Event->Participation->AgentType.Name ⇒ OperationsSegment->PersonnelSpecification. PersonnelClassID
B2	Duality->Decrement->Event->Participation->Agent.Name ⇒ OperationsSegment->PersonnelSpecification. PersonID
B3	Duality->Decrement->Event->Participation->Quantity ⇒ OperationsSegment->PersonnelSpecification.Quantity
C	Duality->Decrement->Event->Stockflow->Equipment/EquipmentType ⇒ OperationsSegment->EquipmentSpecification
C1	Duality->Decrement->Event->Stockflow->EquipmentType.Name ⇒ OperationsSegment->EquipmentSpecification.EquipmentClassID
C2	Duality->Decrement->Event->Stockflow->Equipment.Name ⇒ OperationsSegment->EquipmentSpecification.EquipmentID
C3	Duality->Decrement->Event->Stockflow->Quantity ⇒ OperationsSegment->EquipmentSpecification.Quantity
D	Duality->Decrement->Event->Stockflow->PhysicalAsset/PhysicalAssetType ⇒ OperationsSegment->PhysicalAssetSpecification
D1	Duality->Decrement->Event->Stockflow->PhysicalAssetType.Name ⇒ OperationsSegment->PhysicalAssetSpecification.PhysicalAssetClassID
D2	Duality->Decrement->Event->Stockflow->PhysicalAsset.Name ⇒ OperationsSegment->PhysicalAssetSpecification.PhysicalAssetID
D3	Duality->Decrement->Event->Stockflow->Quantity ⇒ OperationsSegment->PhysicalAssetSpecification.Quantity
E	Duality->Decrement->Event->Stockflow->Material/MaterialType ⇒ OperationsSegment->MaterialSpecification
E1	Duality->Decrement->Event->Stockflow->MaterialType.Name ⇒ OperationsSegment->MaterialSpecification.MaterialClassID
E2	Duality->Decrement->Event->Stockflow->Material.Name ⇒ OperationsSegment->MaterialSpecification.MaterialDefinitionID
E3	Duality->Decrement->Event->Stockflow->Quantity ⇒ OperationsSegment->MaterialSpecification.Quantity
E4	Duality->Decrement->Event->Stockflow->UnitOfMeasure ⇒ OperationsSegment->MaterialSpecification.UnitOfMeasure
E5	Duality->Decrement->Event->Stockflow->Material/MaterialType ⇒ OperationsSegment->MaterialSpecification->MaterialUse = "Consumed"
F	Duality->Increment->Event->Stockflow->Material/MaterialType ⇒ OperationsSegment->MaterialSpecification
F1	Duality->Increment->Event->Stockflow->MaterialType.Name ⇒ OperationsSegment->MaterialSpecification.MaterialClassID
F2	Duality->Increment->Event->Stockflow->Material.Name ⇒ OperationsSegment->MaterialSpecification.MaterialDefinitionID
F3	Duality->Increment->Event->Stockflow->Quantity ⇒ OperationsSegment->MaterialSpecification.Quantity
F4	Duality->Increment->Event->Stockflow->UnitOfMeasure ⇒ OperationsSegment->MaterialSpecification.UnitOfMeasure
F5	Duality->Increment->Event->Stockflow->Material/MaterialType ⇒ OperationsSegment->MaterialSpecification->MaterialUse = "Produced"

Fig. 8. Transformation Rules for ISA-95 Operations Segments

equipment transforms to the *equipment ID* of the *equipment specification* (C2). The *stockflow* association between an *event* and an *equipment* has by default an attribute *quantity* which is mapped to the *equipment specification quantity* (C3). The transformation rules for physical assets (D) are mirroring the ones for equipment (C).

Also the transformation rules for input *materials* (E) are similar to the ones for *equipment* (C) and *physical assets* (D). Evidently, the *quantity* of *materials* is not always a number of pieces. Consequently, there is an additional transformation rule mapping the *unit of measure* of the *quantity* of a *stockflow* to the *unit of measure* of the *material specification* (E4). However, most important is the fact that a *material* connected to a *decrement event* is considered as an input and thus the attribute *material use* of *material specification* is set to the value `consumed` (E5).

The transformation rules B - E describe the input side. The transformation rules for the output side are the ones in section F. The output of an *operations segment* is by definition the produced *material* or *material type* (including the specializations *semi-finished goods* and *finished goods*). In REA, the output are *materials* or *material types* connected via *stockflow* associations to *events* that reside in the *increment* partition. Accordingly, the transformation rules for output *materials* (F) are the same as for input *materials* (E) except for the fact that they apply to the *increment* side and not to the *decrement* side. In addition, the *material use* attribute is set to `produced` (F5). Furthermore, it is worth mentioning that *semi-finished goods* and *finished goods* are specializations of materials, and thus the transformation rules in sections E and F apply as well.

The transformation rules described above are used to map REA duality models to ISA-95 operations segments. In ISA-95, operations segments are not stand-alone items, but are always part of an operations definition. At first sight, one might assume that a REA value chain maps to a single operations definition and all duality models in the value chain become part of this operations definition. However, such an approach is too naive in practice. Our practical experience has shown that usually some duality models are grouped into one operations definition, but it always requires a human decision on this grouping. Accordingly, the transformation of a value chain to an operations definition is always a semi-automatic process requiring feedback from the modeler.

In this paper, we concentrated on the transformation of duality models to operations segments (of operations definition items), because they have a high significance for our approach. Nevertheless, it is important to note that REA also offers concepts to model the attributes of resources (equipment, physical assets, and material) and of agents as well as of their typification. The underlying meta model (cf. [11]) is conceptually very similar to corresponding ISA-95 resource models. Consequently, the transformation is rather straightforward and we do not further elaborate on them due to space limitations.

VII. REA TO ISA-95 TRANSFORMATION EXAMPLE

A. The REA model of Maxi Bike

The business model of Maxi Bike is to produce and sell bicycles. Figure 9 presents Maxi Bike's value chain, which is an instantiation of the value chain meta model depicted on the left hand side of Figure 3. Keeping the example simple and easy to follow, we only present a partial analysis and do not show value activities for acquiring equipment, physical assets, raw materials and labor. The *value chain* covers five *value activities*: `Purchase`, `Transport`, and `Sale`

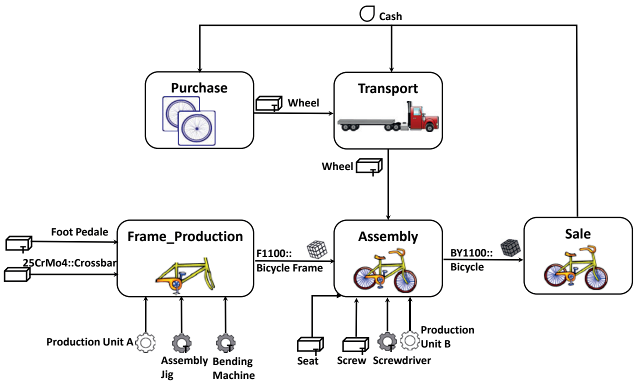

Fig. 9. REA Value Chain of Maxi Bike

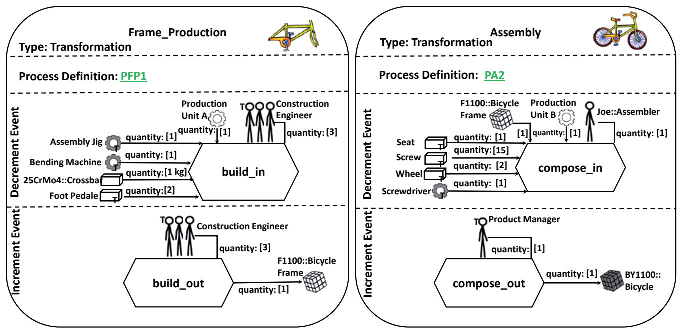

Fig. 10. Frame_Production and Assembly - Duality Models

are REA-transfers requiring horizontal integration, whereas `Production` and `Assembly` are REA-transformations requiring vertical integration. The value chain shows the flow of resources (materials/equipments/physical assets) amongst them. In `Purchase` the *resource* `Cash` is used to get the *material type* `Wheel`. `Wheel` and again `Cash` are used in `Transport` to receive the wheels at the right location (`Production Unit B`).

The *ingoing* resource flows of the *value activity* `Frame_Production` are the *material type* `Foot Pedale`, the *material* `25CrMo4::Crossbar`, the *equipment* `Production Unit A` and the *physical asset types* `Assembly Jig` and `Bending Machine`. The *outgoing* resource flow of this *value activity* is the *specialized material* `F1100::Bicycle Frame` which is a *semi-finished product*. The *value activity* `Assembly` has as *ingoing* resource flows the *material types* `Seat`, `Screw`, `Wheel` and as semi-finished product `F1100::Bicycle Frame`. The other *ingoing* resource flows are the *physical asset type* `Screwdriver` and the equipment `Production Unit B`. These resources are transformed (i.e., used and consumed) to produce the specialized *material* `BY1100::Bicycle` which is the *finished product*. In the *value activity* `Sale` the `BY1100::Bicycle` is turned into `Cash` which is used as input for the other *value activities* mentioned above.

Each of the five *value activities* presented in Figure 9 must be refined by a *duality* model. Due to space limitations we only show the *duality model* for `Frame_Production` and `Assembly` (cf. Figure 10). The left hand side of Figure 10 shows the duality `Frame_Production` which is of the REA type *Transformation*. The `build_in` *decrement* event is performed by the *agent type* `Construction Engineer`. In order to build the *semi-finished product* `F1100::Bicycle Frame`, a *quantity* of `3` engineers is needed. The frame production is carried out in `Production Unit A` and leads to a decrease of a *quantity* of `1 kg` of the input resource *material* `25CrMo4::Crossbar`. To accomplish the frame the *material type* `Foot Pedale` decreases by a *quantity* of `2`. Additionally, the *physical asset types* `Assemble Jig` and `Bending Machine`, each of which with a *quantity* of `1`, are used. In the *increment* event `build_out` the produced good is the *semi-finished product* `F1100::Bicycle Frame` with a *quantity* of `1`, received by `one` *agent type* who has to be a `Construction Engineer`.

The right hand side of Figure 10 depicts the *duality* model `Assembly` which is again a *Transformation*. The `compose_in` *decrement* event is performed by the *inside agent* `Joe::Assembler` and leads to a decrease of the input resources by consuming the *semi-finished product* `F1100::Bicycle Frame` and the *material types* `Wheel` with a *quantity* of `2`, `Screw` with a *quantity* of `15` and `Seat` with a *quantity* of `1`. In addition, the *physical asset type* `Screwdriver` (*quantity* `1`) and the *equipment* `Production Unit B` are used. This *decrement* event is compensated by the *increment* event `compose_out`, which produces the specialized *material* `BY1100::Bicycle` as final product received by the *agent type* `Product Manager`. These example models do not specify any process details on how to produce the bicycle frame or assemble the bicycle. They only provide links to the *Process Definitions* `PFP1` and `PA2`.

B. Mapping the REA Duality Models to B2MML

In contrary to the REA-DSL, ISA-95 does not come with any dedicated graphical language as a concrete syntax to represent ISA-95 compliant models. Accordingly, one may only use a corresponding object diagram as an abstract syntax. However, the Business To Manufacturing Markup Language (B2MML) [19] is an XML implementation of ISA-95. In other words, B2MML defines XML schemas that are exact equivalents of the ISA-95 meta model. Accordingly, one may use a B2MML XML file that is valid with respect to the B2MML schema to show a valid instance of the ISA-95 standard. This is our choice for illustrating the example.

In the following, we demonstrate the mapping of the two duality models `Frame_Production` and `Assembly` as depicted in Figure 10 to B2MML. This mapping uses the transformation rules of Figure 8. The resulting B2MML file is listed in Figure 11. For easier readability, we do not use closing XML tags, but use indent style instead. This B2MML file lists an operations definition with two *operations segments* (`Frame_Production` and `Assembly`), which are both one to one mappings of the REA duality models `Frame_Production` and `Assembly`. All the information in green font is a result of applying our transformation rules. It should be noted that the grouping of the two duality models or operations segements, respectively, has been done manually and, consequently, the instances in black font have to be created manually.

```xml
<OperationsDefinitionInformation>
  <ID> BY1100-ODI
  <Description> Bicycle BY1100 Production
  <OperationsType>Transformation
  <PublishedDate> 2015-03-27
  <OperationsDefinition>
    <ID> BY1100-OD
    <Version> V1
    <Description> BY1100 Bicycle Operations Definition
    <WorkDefinition> WBY1100
    <OperationsSegment>
      <ID> Frame_Production
      <ProcessSegmentID> PFP1
      <PersonnelSpecification>
        <PersonnelClassID> Construction Engineer
        <Quantity> 3
      <EquipmentSpecification>
        <EquipmentID> Production Unit A
      <PhysicalAssetSpecification>
        <PhysicalAssetClassID> Assembly Jig
        <Quantity> 1
      <PhysicalAssetSpecification>
        <PhysicalAssetClassID> Bending Machine
        <Quantity> 1
      <MaterialSpecification>
        <MaterialClassID> Crossbar
        <MaterialDefinitionID> 25CrMo4
        <MaterialUse> Consumed
        <Quantity> 1
        <UnitOfMeasure> kg
      <MaterialSpecification>
        <MaterialClassID> Foot Pedale
        <MaterialUse> Consumed
        <Quantity> 2
      <MaterialSpecification>
        <MaterialClassID> Bicycle Frame
        <MaterialDefinitionID> F1100
        <MaterialUse> Produced
        <Quantity> 1
    <OperationsSegment>
      <ID> Assembly
      <ProcessSegmentID> PA2
      <PersonnelSpecification>
        <PersonID> Joe::Assembler
      <EquipmentSpecification>
        <EquipmentID> Production Unit B
      <PhysicalAssetSpecification>
        <PhysicalAssetClassID> Screwdriver
        <Quantity> 1
      <MaterialSpecification>
        <MaterialClassID> Bicycle Frame
        <MaterialDefinitionID> F1100
        <MaterialUse> Consumed
        <Quantity> 1
      <MaterialSpecification>
        <MaterialClass> Seat
        <MaterialUse> Consumed
        <Quantity> 1
      <MaterialSpecification>
        <MaterialClass> Screw
        <MaterialUse> Consumed
        <Quantity> 15
      <MaterialSpecification>
        <MaterialClass> Wheel
        <MaterialUse> Consumed
        <Quantity> 2
      <MaterialSpecification>
        <MaterialClassID> Bicycle
        <MaterialDefinitionID> BY1100
        <MaterialUse> Produced
        <Quantity> 1
```

Fig. 11. B2MML example: Maxi Bike

The first *operations segment* presented in the B2MML example in Figure 11 with the ID `Frame_Production` is a one to one mapping to the REA *duality model* `Frame_Production`. This *operations segment* contains a *process segment ID* `PFP1` that corresponds to the link specified in the REA *duality* model. The *personnel specification* of the *operations segment* contains the *personnel class ID* `Construction Engineer` which is results from the REA agent type `Construction Engineer` who participates in the *decrement* event `build_in` attributed by a *quantity* of `3`. The *equipment specification* with its ID `Production Unit A` and the *physical assets specifications* `Assembly Jig` and `Bending Machine` are mapped according to the equipment and physical asset types connected to the `build_in` *decrement* event. Each of them has a *quantity* of `1`.

The *decrement* event `build_in` expects a *material type* `Foot Pedale` and a *material* `25CrMo4` which is of *material type* `Crossbar`. Accordingly, we have two *material specifications*. The first one is for the *material class* ID `Crossbar` and the exact *material definition* ID `25CrMo4`, whereas the second one only mentions the *material class* ID `Foot Pedale` without any more detailed *material definition*. These *material specifications* are considered as input resources and thus the attribute *material use*, of both of them, is set to the value `Consumed`. The *material class* ID `Bicycle Frame` with the *material definition* ID `F1100` has the status `Produced` with a quantity of `1`, which is a mapping result of the *increment* event `build_out`. The transformation of the duality `Assembly` to the second *operations segment* is done in the exactly same manner, and thus, is not described in further detail.

VIII. Conclusion

It is our overall goal to develop a universal model-driven approach towards the horizontal and vertical integration in the context of smart factories. For this purpose we strive for an *integrated modeling framework* based on existing modeling approaches. Thereby, we built up on the REA business ontology to identify, both, activities requiring horizontal integration with business partners and activities serving as hooks into the internal systems requiring vertical integration. The latter activities have then to be further detailed by means of the ISA-95 standard. Accordingly, it is of crucial importance to transform concepts of REA to concepts of ISA-95.

First of all, this requires an alignment of concepts that appear to be similar in REA and ISA-95. In this respect, we have extended the resource concept in REA by similar concepts from ISA-95. In particular, we introduce specializations of the concept resource, namely equipment, physical asset, and material. Evidently, these extensions also apply to the REA type level.

Most importantly, we have developed dedicated transformation rules for the purpose of transforming a REA model into an ISA-95 one. In particular, we map REA duality models to ISA-95 operations segments. Thereby, we are able to convert information about the input and output of business functions to the control functions. Nevertheless, it is important to note that later on this information needs to be further detailed on the shop floor control level.

For the evaluation of our approach, we have first implemented the proposed REA extensions into our REA DSL tool. In a next step, we added the transformation rules to our tool. For the moment these rules have been hard coded, but it is planned to use a dedicated transformation language in the future. Accordingly, we demonstrated the technical feasibility of our approach by mapping from REA-DSL to B2MML (the XML equivalent of ISA-95). The syntactical correctness of the transformation has been checked by the proof of valid B2MML XML instances. More extensive case studies are planned for the future, once the overall modeling framework spanning over all hierarchical layers has been realized.

REFERENCES

[1] Industrie 4.0 Working Group, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0," Final report, April 2013.

[2] G. Spur, *Optionen zukünftiger industrieller Produktionssysteme (in German)*. Akademie Verlag, 1997.

[3] *Enterprise system integration – Part 1: Models and terminology*, IEC ISO/IEC 62 264-1, 2012.

[4] *Enterprise-control system – Part 2: Objects and attributes for enterprise-control system integration*, IEC ISO/IEC 62 264-2, 2013.

[5] *Open-edi Part 4: Business transaction scenarios – Accounting and economic ontology*, ISO/IEC ISO 15 944-4, 2007.

[6] T. J. Williams, "The Purdue Enterprise Referene Architecture - A Technical Guide for CIM Planning and Implementation," Institute for Interdisciplinary Engineering Studies, Purdue University, Tech. Rep., 1992.

[7] 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.

[8] C. Y. Yu, *The Structure of Accounting Theory*. The University Press of Florida, 1976.

[9] P. Chen, "The Entity-Relationship Model - Towards a Unified view of Data," *ACM Transactions on Database Systems*, 1976.

[10] C. Sonnenberg, C. Huemer, B. Hofreiter, D. Mayrhofer, and A. Braccini, "The REA-DSL: A Domain Specific Modeling Language for Business Models," in *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011)*. Springer LNCS 6741, 2011, pp. 252–266.

[11] D. Mayrhofer and C. Huemer, "REA-DSL: Business Model Driven Data-Engineering," in *Proceedings of the 14th IEEE International Conference on Commerce and Enterprise Computing (CEC 2012)*. IEEE CS, 2012, pp. 9–16.

[12] ——, "Extending the REA-DSL by the Planning Layer of the REA Ontology," in *Advanced Information Systems Engineering Workshops (CAiSE 2012)*. Springer LNBIP 112, 2012, pp. 543–554.

[13] *Ontology Definition Metamodel (ODM)*, 1st ed., Third Revised Submission to OMG/RFP, OMG Object Management Group, May 2009, available at http://www.omg.org/spec/ODM/1.0/ (checked online July-11-2011).

[14] M. E. Porter, *Competitive Advantage: Creating and Sustaining Superior Performance*. The Free Press, 1985.

[15] G. L. Geerts and W. E. McCarthy, "Modeling business enterprise as value-added process hierarchies with resource-event-agent object templates," in *In Business Object Design and Implementation*. Springer Verlag, 1997, pp. 94–113.

[16] ——. (2005, March) The Ontological Foundation of REA Enterprise Information Systems. [Online]. Available: http://www.msu.edu/user/mccarth4/tulane.doc

[17] ——, "Type-Level Specification in REA ESystems Information Systems." American Accounting Association, August 2005.

[18] B. Scholten, *The Road to Integration - A Guide to Applying the ISA-95 Standard in Manufacturing*. ISA - Instrumentation, Systems, and Automation Society, 2007.

[19] Business To Manufacturing Markup Language (B2MML). World Batch Forum (WBF). [Online]. Available: http://isa-95.com/b2mml/