



Towards an Integrated Architecture Model of Smart Manufacturing Enterprises

Thijs Franck¹, Maria-Eugenia Iacob²(✉), Marten van Sinderen²,
and Andreas Wombacher¹

¹ Aurelius Enterprise, Amsterdam, Netherlands
{thijs.franck, andreas.wombacher}
@aureliusenterprise.com

² University of Twente, Enschede, Netherlands
{m.e.iacob, m.j.vansinderen}@utwente.nl

Abstract. With the introduction of smart manufacturing, the scope of IT expands towards physical processes on the shop floor. Enterprise architects, software engineers and process engineers will have to work closely together to build the information systems that are connected to the shop floor and aligned with the business needs of smart manufacturers. However, it is unclear whether they have the means to do so. This research aims to provide enterprise architecture modelling support for smart manufacturers by investigating ArchiMate 3.0's fitness for this purpose. ArchiMate 3.0 meta-model is compared to the ISA-95 standard for enterprise systems and control systems integration. Modelling patterns are introduced, along with some new modelling concepts, to compensate for deficiencies found. The patterns proposed are validated as part of a case study.

Keywords: ArchiMate 3.0 · Enterprise architecture · ISA-95
Smart manufacturing · Industry 4.0

1 Introduction

Manufacturing companies worldwide are facing the need to improve productivity and quality, as well as implement new products, while shortening innovation cycles. To this end, the manufacturing industry is currently in the process of adopting the new Smart Manufacturing paradigm, also known as the Industry 4.0 paradigm. Smart Manufacturing promises smart machine line operations, high-fidelity models of production processes and improved decision-making support [3].

For the benefits of Smart Manufacturing to materialize, manufacturers will need some way to maintain alignment between their business needs and the information systems that permeate increasingly through all levels of their operations [4, 13]. Maintaining alignment between a company's strategy (the business domain) and its supporting IT is one of the main benefits of enterprise architecture (EA) [1].

The management of processes at the shop floor and the systems used to operate the industrial control devices have traditionally fallen under the Operations Technology (OT) domain of process engineers. As OT increasingly starts to overlap with IT, it makes sense to consider the physical domain from an IT perspective. As a result, the dichotomy between IT and OT fades, in favour of a single EA for the manufacturing domain.

To make this integration between business, IT and OT successful, enterprise architects and process engineers must have a shared modelling language that can express all concepts required for modelling the EA of the manufacturing domain. One of the major requirements introduced by Smart Manufacturing is the modelling of cyber-physical systems (ISCPs). CPS is a type of information system that integrates computational and physical processes and allows these processes to interact [9]. For example, an oven may report real time its temperature curve. If this curve is sub-optimal, the oven wastes energy. Such an insight could be used as input for operational excellence programs, or preventive maintenance.

The modelling of such systems will involve not just viewpoints and concepts related to applications and IT infrastructure, but also to the physical environment (i.e., conditions on the shop floor) [10].

For this research, we adopt the international open standard ArchiMate as our EA modelling language of choice. The most recently published version of the standard, ArchiMate 3.0 [11], already includes several concepts for modelling the physical environment of enterprises. Being a new release, however, it has not been seriously validated or applied in the manufacturing domain.

To ensure that ArchiMate enables the modelling of a smart manufacturer's EA, the standard needs to be validated for that particular purpose. We adopt a process framework and a common object model published as part of the standards suite ANSI/ISA-95 [6, 7] (alternatively, ISO/IEC 62264), or ISA-95 for short, to represent the manufacturing domain. The ISA-95 common object model [7] describe entities at the shop floor level, where IT and OT interact, whereas the ISA-95 process framework describes exactly this interaction.

Conversely, while ISA-95 describes the physical domain, it does not describe the business or IT domains very well, nor was it intended to model EAs in the first place. Thus, to be capable of modelling the EA of a smart manufacturer, ArchiMate 3.0's meta-model needs to be able to express all architectural concepts from ISA-95. To that end, this paper tries to answer the following questions:

RQ1. To what extent can ArchiMate 3.0 express the EA of any smart manufacturer per ISA-95?

RQ2. If ArchiMate 3.0 cannot fully express the EA of any smart manufacturer per ISA-95, what changes to the meta-model of ArchiMate 3.0 are necessary to make this possible?

Thus, the contribution of this research concerns an analysis of whether the meta-model of ArchiMate 3.0 is expressive enough to model an EA in the manufacturing domain. Secondly, we propose a set of modelling patterns describing how ISA-95 concepts can be expressed in ArchiMate. These patterns can be simple direct mappings, or may involve a grouping of Archimate concepts. Finally, to enhance ArchiMate's expressiveness and enable the modelling of certain smart manufacturing concepts some change suggestions are made.

The remainder of the paper is organised as follows. In Sect. 2 we explain the methodology we followed to define a mapping from ISA-95 to ArchiMate, and to analyse the expressiveness of ArchiMate. Section 3 describes the results of the analysis,

and contains the main contribution of the paper. Section 4 gives an account of how we validated our findings. We conclude the paper with a discussion of the related work in Sect. 5 and with conclusions and some pointers to future work in Sect. 6.

2 Methodology

To define a mapping from ISA-95 to ArchiMate and answer research questions, we followed a four-step approach. Firstly, we derived a subset of architectural concepts from the concepts defined by ISA-95. ISA-95 was written with IT/OT integration in mind. To apply its concepts to architecture modelling, an assessment is necessary to find out which concepts qualify as architectural. For this assessment, the same criteria that were used to define the current set of concepts in ArchiMate are applied to each concept in ISA-95. These criteria are explained in Sect. 3.1.

Secondly, we make a comprehensive mapping of the architectural ISA-95 concepts onto ArchiMate 3.0. Criteria used for the mapping are the similarity of concept definitions, as well as similarity of direct relations to other concepts (depth = 1).

Thirdly, the ArchiMate's expressiveness concerning the smart manufacturing domain is investigated by identifying semantic deficiencies in terms of the types defined by Wand and Weber [14] (see Sect. 3.3). We assume that the ISA-95 common object model is a complete representation of entities at the shop floor level. Given our goal of representing this same domain in ArchiMate 3.0, the ISA-95 common object model should fully map onto ArchiMate 3.0. Whether ISA-95 can fully express ArchiMate is not of interest. Thus, we only consider deficiencies of type construct overload, where several ISA-95 constructs map to one ArchiMate construct, and type construct deficit, where an ISA-95 construct does not map to any ArchiMate construct.

The deficiencies identified are subsequently analysed and, if necessary, addressed. In the case of construct overload, an assessment is made concerning critical expressiveness loss as result of the higher abstraction level. In the case of construct deficit, it must be determined whether the intended meaning of the ISA-95 concept can be expressed using a combination, or 'pattern', of constructs currently present in ArchiMate 3.0's meta-model. If the current meta-model is found insufficiently expressive, we suggest a pattern that includes new constructs (i.e., new relations or concepts).

Finally, the identified patterns are validated as part of a case study at SteelCorp. The validation aims to prove the usefulness of the patterns in modelling the EA of a manufacturer, as well to demonstrate the usefulness of such a model through two common manufacturing use cases: an impact of change analysis and an operational excellence analysis.

3 Analysis

The results of several parts of the analysis have been summarized in a spreadsheet (from here on referred to as 'the spreadsheet') which is made available online via <http://bit.ly/2amGJqi>.

3.1 Excluding Non-architectural Concepts from ISA-95

To determine the architectural concepts in the ISA-95 common object model, it is necessary to perform a ‘normalization’ of the ISA-95 concepts to a level of abstraction that coincides with that of ArchiMate concepts. The criteria for normalization are the same as those originally used to determine the ArchiMate concepts. ArchiMate uses for this a layered structure [8]. Starting at the lowest specialization level, concepts are defined in a highly abstract manner as simply entities and relationships between them. At the next level, concepts are specialized as either passive structure concepts, behaviour concepts or active structure concepts, corresponding to the basic structure of the ArchiMate language (dynamic system level). Concepts are then further specialized as EA concepts, which are the ones used to build architecture models. ArchiMate defines implementations of concepts in architecture models as its lowest level of abstraction.

At each specialization step, the utility of the specialization must be argued based on the modelling goals that the modeller has in mind. Following this structure, any ISA-95 concept that is architectural will need a specialization relation to one of the concept types at ArchiMate’s dynamic system level. The concepts at the dynamic system level are listed in Table 1.

Table 1. Dynamic system level concept types [8].

Concept type	Description
Active structure concept	An entity that is capable of performing behaviour
Behaviour concept	A unit of activity performed by one or more active structure elements
Passive structure concept	An object on which behaviour is performed

By eliminating all ISA-95 concepts that do not have a specialization relationship to one of these concepts, we end up with a normalized set of architectural concepts. The normalization analysis reveals that 66% of ISA-95 concepts are architectural. The remaining 33% are non-architectural. For example, ‘person’ qualifies as architectural concept since a person can perform behaviour. Properties describing that person are non-architectural concepts.

To review specifically which concepts classify as architectural, please refer to the spreadsheet.

3.2 Mapping ISA-95 to ArchiMate 3.0

To define a mapping from ISA-95 concepts to ArchiMate we follow a two-step approach: Firstly, for each architectural ISA-95 concept, a comparison is made between its definition and the definition of every ArchiMate concept. Secondly, if there is a fit with one or more definitions, a further comparison is made. In this comparison, each direct relation (depth = 1) of the ISA-95 concept is compared to each of the concepts directly surrounding the ArchiMate concept. This includes both the definition of the surrounding object and the definition of the connecting relationship. If these relations are also in alignment, an ISA-95 concept maps to ArchiMate.

Of the architectural concepts in ISA-95, 12% of ISA-95 concepts map to ArchiMate completely. 75% does have a fit based on definition, but has one or more relations that cannot be mapped. Finally, 13% has no matching definition to start.

For a reference on which specific ISA-95 concept maps to which specific ArchiMate 3.0 concept, please refer to the spreadsheet.

N-to-M mappings. In some cases, it turns out that that several concepts from ISA-95 map to several other concepts from ArchiMate. These mappings are ambiguous, causing uncertainty with regards to which concept to use. According to the mapping, several concepts would be correct. These n-to-m mappings need to be addressed before moving forward. Particularly, this concerns the following two mapping scenarios.

Process Segment

*Process Segment, Process Segment Dependency, Operations Segment,
Operations Segment Dependency*

Map to

Business Process, Business Function, Business Interaction, Business Event

There appears to be an n-to-m mapping in this scenario. However, strictly comparing the definitions of the ISA-95 concepts, as well as the relations they share to surrounding concepts (depth = 1), the ISA-95 concepts turn out to be synonymous. This resolves the n-to-m mapping to *concept redundancy*, which will be addressed in Sect. 3.3. This case shall be further referred to as *Process Segment*.

Equipment

Equipment Class, Equipment

Map to

Business Role, Location, Equipment, Facility

In this second scenario, Equipment and Equipment Class are not synonymous per the ISA-95 meta-model. However, given that ArchiMate does not distinguish between classes and instances, Equipment Class and Equipment can safely be abstracted to mean the same thing. This, again, resolves the n-to-m mapping to *concept redundancy*, which will be further discussed in Sect. 3.3. This case shall be further referred to as *Equipment*.

3.3 Classifying Deficiencies in ArchiMate 3.0

Based on the previously established mapping of ISA-95 onto ArchiMate, several deficiencies in ArchiMate 3.0 can be identified. Classifying each deficiency will help find a suitable solution at a later stage. Four types of deficiency exist [14]. Table 2 describes each type.

We assume that the ISA-95 common object model is a complete representation of the entities on the shop floor. Thus, if ArchiMate is capable of modelling the EA of a smart manufacturer, its meta-model should be capable of expressing ISA-95. Based on this analysis, several cases of construct overload, as well as construct deficit, are uncovered. The following sections discuss the occurrences of each type.

Table 2. Types of deficiencies [14].

Type	Description
Construct overload	Several ontological constructs map to one grammatical construct
Construct redundancy	Several grammatical constructs map to one ontological construct
Construct excess	A grammatical construct might not map to any ontological construct
Construct deficit	An ontological construct might not map to any grammatical construct

Cases of construct overload. Construct overload (i.e., more ISA-95 concepts map onto one ArchiMate 3.0 concept) occurs in the case of the following ArchiMate concepts:

Business Object is used to represent information objects that are used on the shop floor and may serve as a placeholder for more complex entities like a schedule or a bill of materials. Specifically, Table 3 describes the objects that map to Business Object.

Where a business object is used, the model will depend on relationships to other entities to provide the expressiveness needed to model the meaning that the user intends. If this level of expressiveness cannot be achieved, this causes a construct deficit.

Personnel Class and Equipment map to **Business Role**. This happens specifically in the case where Equipment refers to an automated production unit. This abstraction loses the direct distinction between a manual and an automated role. However, depending on whether a given role depends on an actor or not, this distinction can still be derived.

Material Class, Material Definition, Material Lot and Material Sublot map to Material in ArchiMate. Because of this, the distinction between a class of material and a specific type of material used as part of a process is lost. Furthermore, the difference between a class of material and an identifiable (group of) its instances is also lost.

Table 3. Construct overload to business object.

Qualification test specification	Operations material bill
Equipment capability test specification	Personnel specification
Physical asset capability test specification	Equipment specification
Material test specification	Physical asset specification
Material assembly	Material specification
Material definition assembly	Material specification assembly
Material class assembly	Operations schedule
Personnel segment specification	Segment requirement
Equipment segment specification	Personnel requirement
Material segment specification	Equipment requirement
Material segment specification assembly	Physical asset requirement
Physical asset specification	Material requirement

Cases of construct deficit. Several deficits have been identified as part of the mapping analysis. When a deficit occurs, the ISA-95 concept cannot be expressed in ArchiMate. Each deficit is explained in the paragraphs below.

Various concepts in ISA-95 are related to a *test specification* that is used to test certain properties of said concepts. A Test Specification maps to a Business Object. The ArchiMate meta-model only allows for an association relationship between Active Structure concepts and a Business Object. The dependency in ISA-95 is, however, stronger (<is tested by>).

An *assembly* is a collection or set of related elements. In ISA-95, they are represented as classes related to aggregation relationships between elements. In ArchiMate, every element can also have an aggregation relation with an element of the same type. There is, however, no class that represents information about this relation.

A *process segment* (maps to business process) in ISA-95 is a collection of several concepts, including specific *parameters* that do not fall into the category of personnel, equipment, physical asset or material. The ‘other’ parameters are known as process segment parameters. ArchiMate allows only well-defined concepts to be related to a business process.

While an ISA-95 Material can be directly mapped to an ArchiMate material, a problem occurs when attempting to map a *Material Lot*. A requirement for a Material Lot is that it should be possible to determine its current state based on the lot ID. This requires traceability to an information object, i.e., a Business Object. While it is possible to relate a Material Lot to a Business Object through an association, the relationship between a physical and an information object is deemed more meaningful.

The *operations definition* describes the relation between a production, maintenance, inventory or quality operation, the way in which it is implemented and the resources that are needed. A framework for these kinds of manufacturing operations is defined by the first part of ISA-95 [6]. ArchiMate only loosely defines business processes, independent of their context.

ISA-95 defines a *schedule* concept. It is implemented as a set of *operations* requests, which directly relate to an operations definition. There is no similar concept in ArchiMate.

ISA-95 makes a distinction between the definition of a process, the planned process and the actual process. Once executed, *Operations Responses* are returned for every Operations Request (which make up the schedule). In ArchiMate, an Operations Response can be represented as either a Business Object or Data Object, depending on whether this information is collected digitally or not. The actual production information is, however, much too volatile to model as part of the architecture.

3.4 Addressing the Deficiencies Found

Now that several deficiencies have been identified, solutions can be defined that allow ArchiMate to express all the architectural concepts in ISA-95, thus making the language suited to model the shop floor and, by extension, the EA of a smart manufacturer.

The solutions to the deficiencies identified will be discussed below as modelling patterns. A pattern is a set of constructs from ArchiMate that expresses a certain aspect

of ISA-95. Preferably, only existing constructs will be included in these patterns. If a new construct must be introduced, it will conform to the requirements for constructs in ArchiMate [11]. The following paragraphs discuss the solutions per deficiency.

Test Specifications. Various concepts in ISA-95 are related to a test specification that is used to test certain properties of said concepts. Often, these concepts are mapped to active structure concepts in ArchiMate. For example, a Person (maps to Actor) relates to a Qualification Test Specification (maps to Business Object). A Business Object is, however, a passive structure concept. The ArchiMate meta-model only allows for an association relationship between Active Structure concepts and a Business Object. As discussed in Sect. 3.3, we must rely on the context of the ArchiMate model to define the meaning of a Business Object. For a Test Specification, which has a very specific purpose in ISA-95, we deem an association relationship insufficient, since this association without context can be interpreted in many ways.

A stronger relation [2] between an Active Structure concept and a Business Object can only be established via a Behavior concept, specifically the assigned to relations (for Active Structure concepts) and accesses relations (for Business Objects) to Business Service, Business Event and Business Process.

Since relations from the physical layer are only allowed to Business Process, Business Function and Business Interaction (and not Business Service or Business Event), this leaves Business Process as the only option. Given this limitation, we define the Test Specification Pattern as shown in Fig. 1.

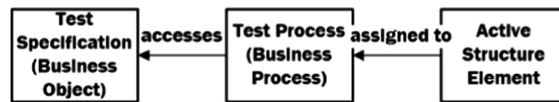


Fig. 1. Test specification pattern for ArchiMate 3.0.

Assemblies. An assembly is a collection or set of related elements. In ISA-95, they are represented as classes related to aggregation relationships between elements. In ArchiMate, every element can also have an aggregation relation with an element of the same type. There is no class that represents information about this relation. For example, to express the size of an assembly in ArchiMate, it would be necessary to model an entity for each element in the collection. This makes sense in a scenario where each instance of a class is meaningfully different. For example, since every person has different qualifications, it is meaningful to model people separately as part of a team. However, in the case where the elements of a collection are not meaningfully different, e.g. a set of materials used for the production of a batch (bill of materials), it makes more sense to model each material as a class rather than as separate instances. When modelling only a class, however, the quantity of the material used for the production of e.g. a batch is still meaningful information. Both alternatives below present a solution that makes use of a parameter to a relationship to express meaning. Such a pattern can also be used to express the Operations Material Bill Item concept per ISA-95.

Alternative 1. To model such information relevant to an assembly, parameters for the relation between the class (e.g. a material) and the assembly (e.g., a bill of materials) is proposed. While ISA-95 defines assemblies broadly, in the specification they only occur in relation to materials. A placeholder mapping for assembly would be a business object. Currently, there exists an indirect relation between Business Object and Material through Business Process. The information relevant to an assembly could be attached to the relation between Material and Business Process as a (set of) parameter(s).

Alternative 2. This implementation eliminates the need for a separate Business Object by modelling the bill of materials implicitly through the set of relations between said Business Process and the Materials used. Figure 2 illustrates the proposed pattern.

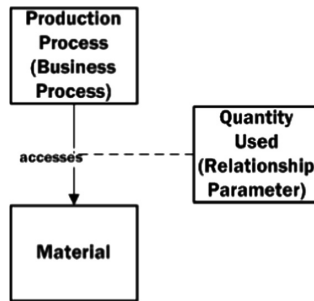


Fig. 2. Implicit bill of materials pattern for ArchiMate.

However, the solution presented in alternative 1 does not allow for a bill of materials to be modelled explicitly. A bill of materials is quite common in manufacturing, so the capability to include this concept explicitly may be desirable. To do so, a direct relation between Business Object and Material is necessary. An association relationship is currently available between Material and Business Object. However, as explained in Sect. 3.3, we feel that the use of an association relationship in this case is not sufficiently expressive.

Instead, an aggregation relationship is proposed. An aggregation relationship indicates that a concept (the bill of materials) groups a number of other concepts (materials). While Materials are meaningful independent of one another, the bill of materials groups them for the purposes of use in a production process.

The proposed parameters would be attached to this relationship. This solution is, however, not perfect either. The relation between Business Object and Material makes the relation between Business Process and Material redundant, since the Bill of Materials will always be related to a production process (Business Process).

Figure 3 shows a pattern for the modelling of an explicit bill of materials. There are two major differences between this pattern and the pattern for an implicit bill of materials. Firstly, this pattern includes a Business Object that denotes the bill of materials. This Business Object is related to the Business Process via an accesses relation. This relation currently exists in ArchiMate. The bill of materials lists one or

more Materials via an aggregation relation. This aggregation relation is newly introduced for this purpose. Secondly, the information describing the assembly is related to the aggregation relation between Material and Business Object, as denoted by the dotted line.

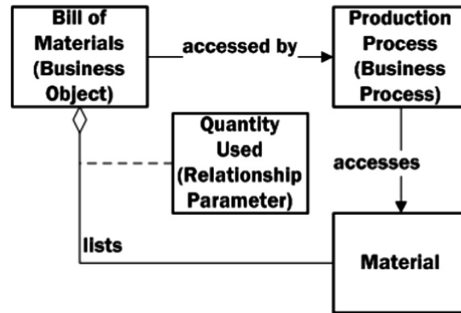


Fig. 3. Explicit bill of materials pattern for ArchiMate.

Process Segment Parameter. A process segment (maps to business process) in ISA-95 is a collection of several concepts, including specific parameters that do not fall into the category of personnel, equipment, physical asset or material. The ‘other’ parameters are known as process segment parameters. For a production process, an example might include the known lead time of a process step (e.g. the steel coil needs to be in the oven for 10 min). For a quality process, a parameter might be the size of the sample to be pulled (e.g. 1 coil).

ArchiMate allows only well-defined concepts to be related to a business process. The only concepts that fit with the description of Process Segment Parameter (i.e. related to business process and not a person, equipment or material) are Business Service and Business Event (behavior), or Business Object (passive structure). A timer like in the oven example would typically be modeled as an event, but a parameter like sample size cannot be expressed formally in ArchiMate. If needed, such information can be included as part of the sub-process name (e.g. take a quality sample, size 1). Modelling this information as such works as a way to capture it informally, e.g. for presentation purposes. However, for analysis purposes, a more formal approach will be required, since information stored in a concept name cannot be queried easily.

The proposed solution is to introduce parameters related to a business process. This is similar to the solution introduced to model assemblies, with the difference being that the parameters are related to a concept rather than a relation. Examples of parameters are average duration, sample size or temperature. This parameter pattern can also be used to model other manufacturing object parameters, per the ISA-95 object properties. The parameter pattern for concepts is illustrated in Fig. 4.

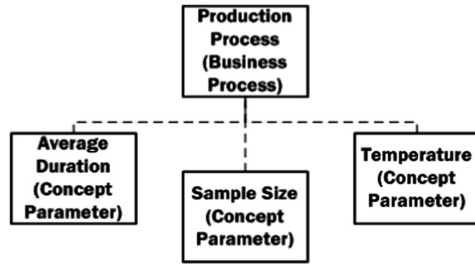


Fig. 4. Process parameter pattern for ArchiMate 3.0.

Material Lot. While an ISA-95 Material can be directly mapped to an ArchiMate Material, a problem occurs when attempting to map a Material Lot. The current state of a Material Lot should be accessible via its ID. This requires traceability to an information object, i.e. a Business Object. It is possible to associate a Material with a Business Process and a Business Object with a Business Process. It is even possible to draw an association between Material and Business Object. In the case of a Material Lot, however, the relationship between Physical Object and Information Object is more meaningful than an association. The relationship should describe how the informational object reflects the state of the physical object it represents.

To add this expressiveness, a realization relationship is proposed. A realization relationship links a logical entity with a more concrete entity that realizes it. Thus, a realization relation could describe how a physical object is represented by an information object. Furthermore, a Data Object may realize a Business Object. This Data Object can, by means of an indirect relation, be considered as the digital representation of said Material stored in some information system. This extrapolation would not be valid if a weaker relation should be used between the physical object and the Business Object. Finally, by linking the data model of said Data Object to the architecture, it becomes possible to perform analyses of a material’s production lifecycle.

The same logic also applies to other physical elements. For example, a piece of equipment may be used as part of a business process, causing it to change state (e.g. from ‘idle’ to ‘in use’). Per ISA-95, entities associated with processes include materials, as well as physical assets, equipment and people. Because of this relation in ISA-95, the same realization relation that is proposed for ArchiMate between Material and Business Object is also proposed as a relation between Business Object and Business Role, Business Actor, Equipment and Facility (see Table 4).

Table 4. Proposed relations.

ArchiMateConcept	ISA-95 concept	Relation	Concept
Material	Material lot	Realizes	Business object
Business role	Personnel class	Realizes	Business object
Business actor	Person	Realizes	Business object
Equipment	Equipment class	Realizes	Business object
Facility	Physical asset	Realizes	Business object

Finally, the Business Process concept is included to show that the newly added realization relation is only intended for those concepts that have an accesses or assigned to relation with Business Process.

Figure 5 illustrates the proposed extension. The newly added realization relation is marked with a red circle. For the sake of legibility, the ‘Physical Elements’ block serves as a placeholder for the ArchiMate concepts listed in Table 4. The figure also shows how an indirect realization relation between Data Object and a Physical element can be derived using the realization relation between Data Object and Business Object.

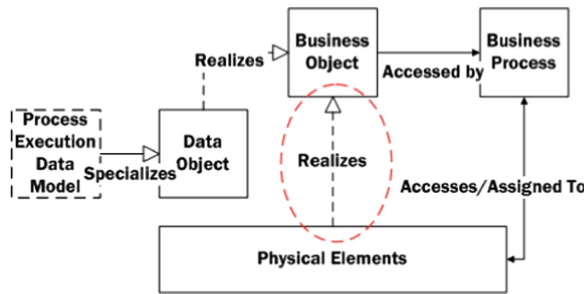


Fig. 5. Informational representation of a material. (Color figure online)

Operations Definition. The operations definition describes the relation between a production, maintenance, inventory or quality operation, the way in which it is implemented and the resources that are needed to carry out the process. A framework for these kinds of manufacturing operations is defined by the first part of ISA-95. ArchiMate only loosely defines business processes, independent of their context. However, the ISA-95 process framework [6] can be implemented in ArchiMate. It can then provide structure through composition relations from framework processes to processes that are company-specific.

Figure 6 shows a pattern for how to apply the ISA-95 process framework to company-specific business processes. Such processes are modeled as sub-processes (hence the composition relation) of processes from the ISA-95 process framework. Since both ISA-95 processes and their sub-processes have flow relations between them, sub-processes cannot compose more than one ISA-95 process. If a process in a currently existing model cannot fulfil this requirement (e.g. Batch Annealing in Fig. 6), that process needs to be decomposed such that each sub-process only has one relation to an ISA-95 process.

ISA-95 also explicitly defines a Bill of Materials in relation to an Operations Definition. A business object best fits the definition, but a business object cannot have a relation to a material (except through a business process). ArchiMate does implicitly define a bill of materials through the accesses relation between business process and material. The pattern introduced for Assemblies solves this issue.

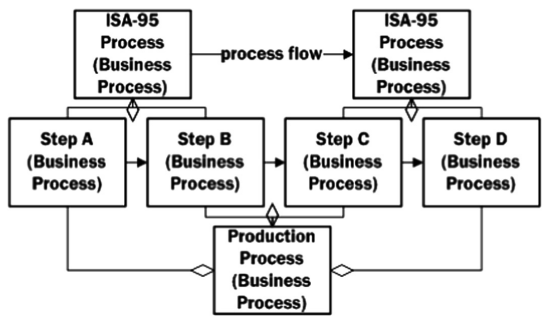


Fig. 6. Operations definition pattern (incl. example).

Operations Schedule. ISA-95 defines a schedule concept. It is implemented as a set of operations requests, which directly relate to an operations definition. There is no particular ordering (time sequence) to the set. There is no similar concept in ArchiMate. While the schedule itself could be modeled as a business object, another issue arises with regards to the relation between a business object and a business process. A business process is typically modeled as a class in ArchiMate, while the schedule must relate to instances to be meaningful. It would either be necessary to model each instance of the process separately, or to model no relations to business processes at all, effectively making the schedule a placeholder object. The first is preferable from an analysis standpoint, while the second is preferable from a complexity standpoint. A compromise between these two options is to, rather than model each instance as part of the architecture, include a reference to the data model used to store each instance (Fig. 7). This data model can then serve as the basis for a query. The way in which this query is structured shall depend on the viewpoint for which the information is required. For example, a query based on product ID may reveal which execution path was followed for the production of that unit.

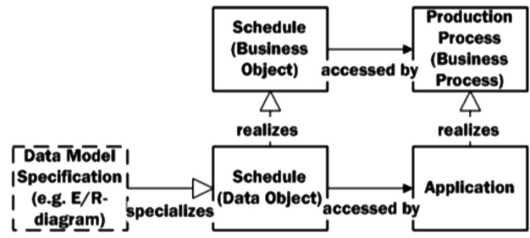


Fig. 7. Operations schedule pattern.

Operations Performance. ISA-95 makes a distinction between the definition of a process (operations definition), the planned process (operations schedule) and the actual process (operations performance). Once executed, Operations Responses are returned for every Operations Request (which make up the schedule). An Operations

Response is made up of ‘actuals’, which represent the people, equipment, materials and physical assets that were utilized. In ArchiMate, an Operations Responses can be represented as either a Business Object or Data Object, depending on whether this information is collected digitally or not. The actual production information, such as the actual execution of the process, any errors that may have occurred, is however much too volatile to model as part of the architecture. Instead, it is recommended to relate an Operations Response object to a specification of the data model, describing how the data can be obtained externally (e.g. an E/R-diagram). Based on this specification and the relation to a data object accessed by some application, it will be possible to generate a query for analysis purposes. The proposed pattern is shown in Fig. 8.

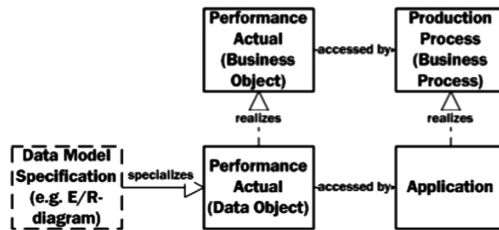


Fig. 8. Performance actual pattern.

4 Validation

A case study has been done to validate whether ArchiMate (plus ISA-95 based modelling patterns) effectively introduces EA modelling capability to manufacturers. The case concerned a large steel manufacturer (named SteelCorp for the sake of anonymity) that intends to make a change in one of its production processes. We show how the models created do not only help manufacturers in their transformation efforts, as well as how the models created can be used as a basis for further analysis applicable to a smart manufacturing landscape.

The process modelled as part of this case study concerns a batch annealing process for steel coils at one of SteelCorps factories. During batch annealing, a group of three coils is placed into an oven. Heat is applied over a period of time to change certain qualities of the steel. SteelCorp is looking to optimize this process and to harmonize its surrounding application landscape.

Proposed optimizations include the integration of information used in several activities preparatory to production into the production planning system (PPS). The PPS is used to manage the utilization of the ovens. Secondly, to increase oven utilization, SteelCorp plans to generate optimized batches of coils from the PPS, rather than having employees combine each batch manually. Thirdly, to minimize waiting times once a batch leaves the oven, SteelCorp wants to know how long it takes for a coil to cool down in inventory. For this reason, they will install thermometers that monitor each coil periodically. Finally, actual oven temperature curves will be recorded and stored in the data warehouse with the intent of using this data to optimize energy efficiency.

4.1 Modelling

Creating a model of this process involved establishing the batch annealing process formally as part of the business domain, as well as modelling its relation to the physical, digital and IT infrastructure domains. Notable physical objects include the steel coils and the ovens. Information is associated with these objects at several stages in the process and this information moves through several systems throughout the production lifecycle.

An as-is process model was made. This model could successfully be used to demonstrate the challenges SteelCorp was facing and to motivate the proposed changes. The as-is model included first and foremost a process model describing each step in the batch annealing process. We modelled this primarily using business processes. Further, we modelled materials, equipment and facilities and linked them directly to the business processes. This helps to show the flow of goods and the use of physical assets across the shop floor, as well as how the process depends on these assets. We think this is a very common manufacturing use-case.

Additionally, we modelled the dataflow that occurs in the process. This helped us show the dependencies between the process and the applications running in the plant. This model allowed us to show how the data generated from the physical assets was being handled, and how it was being used in the process.

Next, a to-be model including the same elements as the as-is model was created to show how the proposed changes would contribute to the goals set forth by SteelCorp. The holistic overview provided by the model, including the process, application landscape, data flow and physical perspective, was of particular interest to them as their analyses so far had been mostly domain-specific.

4.2 Impact Analysis

We performed an impact analysis of the proposed changes. An impact analysis shows how the changing dependencies between processes, applications, data and physical assets affect the flow of the process, data and physical goods across the shop floor. We find it important to note that, in the process of adding processes, applications and physical assets to our model, a lot of (implicit) dependencies are created that, while adding a lot of useful details, could at times make the model hard to use and maintain.

4.3 Performance Analysis

We used our architecture model as the basis for some analyses using shop floor data to predict future performance of the process. Since we believe this is a use-case that is specifically applicable to an industry 4.0 scenario, we will describe it in more detail.

To make the best use of our architecture model in this case, we needed to add the structure of our data objects to the model. Since ArchiMate lends itself poorly to this (and is not intended for this use-case in the first place), we chose to model our data structures in UML instead and link them to our ArchiMate model.

As a metric for the performance of the batch annealing process, we chose operational excellence. An operational excellence score is based on three factors: quality, availability and utilization. Each of these three factors is scored based on their conformance to some norm. For example, the goal may be to achieve 1 defect in 100.000.000 products, which would correspond to a 10 (on a scale of 1–10). To determine the overall operational excellence score, the average of the three scores is taken.

We illustrate this use case with another example from the batch annealing process. Currently, when steel coils leave the oven, they are placed into inventory to cool down. Operators assume this process takes around a day for every coil. Thus, every coil stays in inventory for a day. This day cooling down time will serve as the norm for the availability score of this process step.

However, SteelCorp expects that the cooling down time of coils is actually variable, depending on the type of steel used, the thickness of the coil and the temperature at which the oven was running. Thus, by determining the cooling down time of a certain type of coil, it may be possible to improve operational excellence by reducing the time a coil spends in inventory. By moving a coil out more quickly, that coil can be delivered to the customer sooner, improving availability. Furthermore, inventory space is freed up for the next coil, improving utilization.

To determine the actual cooling down time of coils, the company wishes to change the current process in several ways. Firstly, an operator will be assigned to manually measure the temperatures of every coil as part of a pilot phase. During this phase, temperature data will be collected to establish a baseline cooling down time for a certain type of coil. At a later stage, this measurement process will be automated, allowing operators to determine real time when which each individual coil is cooled down.

The factor that determines whether a coil can be moved out of inventory is its current temperature. In the current process, there is no way to tell the temperature of a coil. In the to-be process, however, data will be collected which can be used to tell whether a coil is sufficiently cooled down. The simplified to-be architecture is shown in Fig. 9.

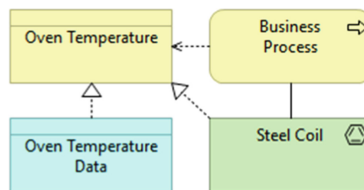


Fig. 9. Operational excellence example architecture.

The details of the to-be process and the new data services are shown in Fig. 10. The elements highlighted in read will be impacted by the change. There are a couple of major changes made to the application landscape. Most notably, the Excel spreadsheet

is replaced entirely by the PPS. The PPS now stores all production planning information. The data warehouse is now also directly linked to the production process, since oven temperature data is now stored for temperature curve analysis. An undetermined database solution will be used to store cooling down temperature data.

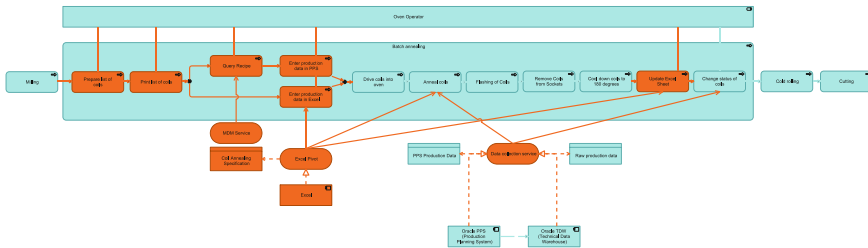


Fig. 10. Impact analysis on the to-be architecture.

Prominent entities in the cooling down process include the coil and a business object describing the temperature of the coil. The coil temperature is realized by the steel coil, as well as a data object that stores the information. The structure of the temperature data, while currently not yet determined, might be as shown in Fig. 11.

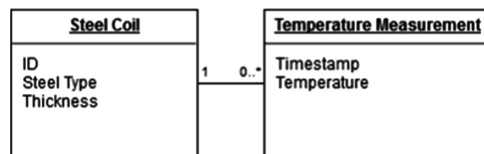


Fig. 11. Operational excellence example data structure.

Using the temperature data, a cooling down curve can be derived (see Fig. 12). This example uses mocked up historical data to illustrate how this might work. The example assumes a sample of the current temperature will be recorded every hour. Temperature may also be monitored in real-time for more accuracy.

The above temperature curve shows how a coil cools down over a period of time. The coil enters the warehouse at 12:35:27. At that time, the coil temperature is 1400 °C. Over time, the coil cools down at a linear rate of 55° per hour. The target temperature of 180 °C is reached at approximately 10:45:00. This means the coil has had a cooling down period of around 22 h and 10 min. This is one hour and fifty minutes less than the normal duration of 24 h.

Using the actual cooling down times of coils, certain patterns may be revealed regarding the cooling down time of a type of coil. This way, rather than reactively moving a coil out of the warehouse as soon as it is cool enough, planning can take into account the expected cooling down time of a coil.

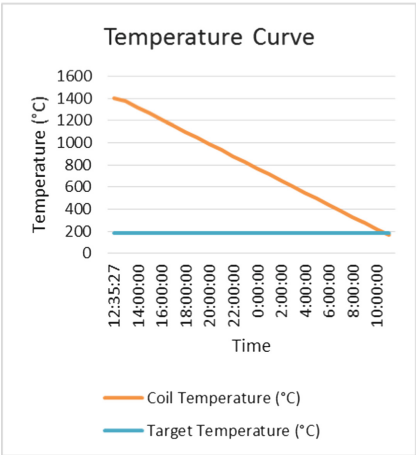


Fig. 12. Example cooling down curve of a steel coil.

Score calculation. Taking the above temperature curve as the expected temperature curve for this type of coil indicates that the current process is inefficient in terms of both availability and utilization by 7.6%. Assuming every other aspect of the process to be perfect, the operational excellence score calculation for the as-is situation is as shown in Table 5.

Table 5. Operational excellence score (no data use).

Factor	Score
Quality	10
Availability	9.24
Utilization	9.24
Operational excellence score	8.53

In the to-be situation, availability and utilization will be improved. The magnitude of improvement will depend on how the data is used. If the temperature data is used reactively, notifying operators that a coil is ready based on the hourly measurement, the actual gains will be somewhat variable based on what time a coil has entered the warehouse. At most, the time wasted will be an hour. Based on the most pessimistic scenario, this means an improvement to availability and utilization of 3.4%, leading to the scores shown in Table 6.

Table 6. Operational excellence score (reactive data use).

Factor	Score
Quality	10
Availability	9.58
Utilization	9.58
Operational excellence score	9.18

Finally, if the temperature data is used to its fullest, by using the data to compute average cooling down times based on a coil's cooling down temperature curve, the amount of time wasted can be minimized, leading to a perfect operational excellence score (of course, assuming every other task is performed fully efficiently as well), as shown in Table 7.

Table 7. Operational excellence score (temperature curve use).

Factor	Score
Quality	10
Availability	10
Utilization	10
Operational excellence score	10

4.4 Effectiveness of Patterns

The proposed modelling patterns proved useful in several instances. Patterns based on existing ArchiMate concepts were enough to model most of the case. However, some aspects of the case could not be modelled and required the use of modelling patterns that make use of new elements. For example, the current utilization of an oven (and the discrepancy between the perceived utilization of an oven and its actual utilization) could not be modelled. This required a realization relation to a business object per the pattern introduced in Sect. 3.4 (Table 4). Another example of this is the temperature data related to a steel coil that is monitored at regular intervals during the process.

5 Related Work

Urbaczewski and Mrdalj [12] reviewed the EA frameworks available at that time. They identified DoDAF as the only framework that allows for the modelling of physical elements. In another literature review Hermann et al. [5] identify CPS as major principle behind smart manufacturing/industry 4.0. Furthermore, Sacala and Moisescu [10] argue that modelling a CPS as part of an overall enterprise systems landscape requires a physical entity, an association with a business entity and an application with interfaces to both the business and the physical entity. Finally, The Open Group released in 2016 ArchiMate 3.0 [11], which introduced (among other things) several modelling concepts to describe physical elements and how these elements relate to applications and business entities. The current research draws upon all the above and relates ArchiMate to ISA-95, by exploring its current modelling capabilities for smart manufacturing.

6 Future Work

Since publishing the first version of this paper, we have had several more opportunities to further experience apply architecture models in the smart manufacturing/industry 4.0 domain. We have observed that, while many manufacturers use models to design their

products and formalize their processes, most are lacking a holistic architectural overview of their plants. SteelCorp remains quite representative of a typical manufacturing environment in this sense. At the same time, there is a great need for insights derived from data gathered by the manufacturers. Our current and future work remains focussed on using architecture models based on languages like ArchiMate and UML to provide not just the IT department, but also the other stakeholders in manufacturing with the insights they require to make the industry 4.0 transition.

In this light, we have observed several areas with potential for additional contributions. Firstly, the models that manufacturers are currently using are often not based on ArchiMate. In fact, they are often not based on any formal modelling language. In practice, this means that much manual labor is required to interpret existing documentation and transform it into a structured language like ArchiMate. This creates a high barrier to entry for manufacturers.

Further, since existing documentation will often be composed independent and regardless of other, similar documents, it can be difficult to integrate each of these sources into a coherent model. One problem is that authors will often use a level of abstraction in their descriptions that fits the use-case that they were addressing with their document. However, when attempting to create a cohesive models, this source material will often either leave unfilled gaps, or contain too many details as compared to the rest of the model. Furthermore, the use of terms and definitions is often inconsistent. To eliminate such inconsistencies, we need a way to discern different levels of abstraction in a model. When a model is on the same abstraction level, it may be possible to do pattern matching in the model and thereby eliminate many of the inconsistencies we are currently experiencing.

Finally, we continue to look into using the model to define interesting perspectives on the data that is gathered from the shop floor. Based on the dependencies in an architecture model, it may for example be possible to derive dimensions that are of interest to a particular stakeholder working on a certain part of the production process.

7 Conclusions

With the introduction of smart manufacturing (or Industry 4.0), IT and operations technology increasingly intertwine. For large manufacturers, this means increasing digitization of the shop floor and, consequently, the need to control the information flowing from the physical domain and to manage changes from a multidisciplinary (IT and OT) perspective. This is where EA helps, but existing EA frameworks and languages were not designed with this type of requirements in mind.

This research provides EA modelling support for smart manufacturing companies. Based on the ISA-95 standard for the integration of enterprise systems and control systems in the manufacturing industry [6, 7], this research has presented an analysis of ArchiMate 3.0 [11] in terms of its coverage of the manufacturing domain. The results of the analysis lead to the following answers to the research questions formulated in the introduction:

RQ1: Since ISA-95 was written on a different abstraction level than ArchiMate, not all of its concepts may be of architectural nature. To determine which concepts are architectural, the ISA-95 concepts were normalized using the criteria used to determine which concepts should be part of the ArchiMate language [8]. The normalization revealed that only 66% of ISA-95 concepts qualify as such. Given the set of architectural concepts identified, a mapping was made of each architectural ISA-95 concept to ArchiMate 3.0. To be able to express the EA of any smart manufacturer, ArchiMate should be able to express each architectural ISA-95 concept. The mapping analysis revealed that, while 12% of concepts can be mapped one-to-one, construct overload or deficit [14] occurs in 75% of cases. Solving these issues requires the use of modelling patterns based on either indirect relations or on new constructs.

RQ2: When a concept from the manufacturing domain cannot be mapped to ArchiMate, this will invariably cause issues when attempting to model the architecture of a manufacturing enterprise. Thus, this second question asks for a solution to the mapping difficulties uncovered as part of the mapping analysis.

For each identified issue, a pattern has been proposed that resolves the problem by using some combination of ArchiMate concepts to express the intended meaning of the ISA-95 concept, and/or by introducing some new constructs if ArchiMate's meta-model does not have sufficient expressive power. The following concepts are introduced:

1. *Concept Parameter and Relationship Parameter*. These concepts describe information about a concept (e.g. a steel coil) or relation (e.g., an item on a bill of materials) respectively.
2. An *aggregation relation between Material and Business Object* is proposed to enable the modelling of an explicit bill of materials.
3. A *realization relation between Business Object and Business Actor, Business Role, Material, Equipment and Facility* will allow for both the current physical and informational state of a physical object to be modeled.

The proposed modelling patterns enhance ArchiMate 3.0's coverage of ISA-95 architectural concepts from 12% to 92%, and were validated as part of a case study. They proved useful in modelling part of the production process at a steel manufacturer. The models could also effectively be used to perform two common analysis use-cases: impact analysis and operational excellence analysis.

Note that the proposed modelling patterns are applicable to ArchiMate only. Furthermore, the patterns should be further validated by testing them in more cases, also covering discrete and continuous processes, since SteelCorp is a batch process.

References

1. Boucharas, V., van Steenberghe, M., Jansen, S., Brinkkemper, S.: The contribution of enterprise architecture to the achievement of organizational goals: a review of the evidence. In: Proper, E., Lankhorst, M.M., Schönherr, M., Barjis, J., Overbeek, S. (eds.) TEAR 2010. LNBP, vol. 70, pp. 1–15. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16819-2_1

2. van Buuren, R., Jonkers, H., Iacob, M.-E., Strating, P.: Composition of relations in enterprise architecture models. In: Ehrig, H., Engels, G., Parisi-Presicce, F., Rozenberg, G. (eds.) ICGT 2004. LNCS, vol. 3256, pp. 39–53. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30203-2_5
3. Davis, J., Edgar, T., Graybill, R., Korambath, P., Schott, B., Swink, D., Wang, J., Wetzel, J.: Smart manufacturing. *Ann. Rev. Chem. and Biomol. Eng.* **6**, 141–160 (2015)
4. Henderson, J.C., Venkatraman, H.: Strategic alignment: Leveraging information technology for transforming organizations. *IBM Syst. J.* **31**(1), 472–484 (1993)
5. Hermann, M., Pentek, T., Otto, B.: Design principles for Industrie 4.0 scenarios: a literature review. *Technische Univ. Dortmund*, No. 01, pp. 4–16 (2015)
6. International Society of Automation: Enterprise-Control System Integration. ANSI/ISA standard 95.00.01-2010 (2010a)
7. International Society of Automation: Enterprise-Control System Integration. ANSI/ISA standard 95.00.02-2010 (2010b)
8. Lankhorst, M.M., Proper, H.A., Jonkers, H.: The anatomy of the archimate language. *Int. J. Inf. Syst. Model. Des. Arch.* **1**(1), 1–32 (2010)
9. Lee, E.A.: Cyber physical systems: design challenges. In: Proceedings of the 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), pp. 363–369 (2008)
10. Sacala, I.S., Moisescu, M.A.: The development of enterprise systems based on cyber-physical systems principles. *Rom. Stat. Rev.* **4**, 29–39 (2014)
11. The Open Group: ArchiMate 3.0 Specification (2016)
12. Urbaczewski, L., Mrdalj, S.: A comparison of enterprise architecture frameworks. *Issues Inf. Syst.* **7**(2), 18–23 (2006)
13. Wagner, H.-T., Weitzel, T.: Operational IT business alignment as the missing link from IT strategy to firm success. In: 12th Americas Conference on Information Systems (AMCIS 2006), AISeL, pp. 570–578 (2006)
14. Wand, Y., Weber, R.: Research commentary: information systems and conceptual modelling - a research agenda. *Inf. Syst. Res.* **13**(4), 363–376 (2002)