Determining the Quality of Product Data Integration

Julian Tiedeken[1], Thomas Bauer[2], Joachim Herbst[3], Manfred Reichert[1]

[1] Institute of Databases and Information Systems, Ulm University, Germany
{julian.tiedeken, manfred.reichert}@uni-ulm.de
[2] Department Information Management, University of Applied Sciences, Neu-Ulm, Germany thomas.bauer@hs-neu-ulm.de
[3] ITM Group Research & Product Development MBC, Daimler AG, Böblingen, Germany
joachim.j.herbst@daimler.com

Abstract. To meet customer demands, companies must manage numerous variants and versions of their products. Since product-related data (e.g., requirements' specifications, geometric models, and source code, or test cases) are usually scattered over a large number of heterogeneous, autonomous information systems, their integration becomes crucial when developing complex products on one hand and aiming at reduced development costs on the other. In general, product data are created in different stages of the product development process. Furthermore, they should be integrated in a complete and consistent way at certain milestones during process development (e.g., prototype construction). Usually, this data integration process is accomplished manually, which is both costly and error prone. Instead semi-automated product data integration is required meeting the data quality requirements of the various stages during product development. In turn, this necessitates a close monitoring of the progress of the data integration process based on proper metrics. Contemporary approaches solely focus on metrics assessing schema integration, while not measuring the quality and progress of data integration. This paper elicits fundamental requirements relevant in this context. Based on them, we develop appropriate metrics for measuring product data quality and apply them in a case study we conducted at an automotive original equipment manufacturer.

Keywords: Product Data Integration, Integration Quality, Integration Process

1 Introduction

During the last 15 years, the amount of product data has more than doubled, while at the same time the duration of product development lifecycles has decreased by 25 percent [13]. In addition to product data management systems [26], which track and manage changes related to product data, proprietary information systems are used by the various business divisions involved in product development to manage specific product data (e.g., test cases). In particular, many information systems were introduced to quickly adopt to new business challenges such as emerging technologies, standards, or legal regulations. Consequently, product data are scattered over a multitude of information systems managing data of different quality. Usually, distributed product data cover different perspectives on the product (e.g., requirements' specifications, geometric models, source code, and test cases), which are recorded for different purposes at different points in time. Finally, different techniques for handling variants and versions of product data are prevalent [8].

At certain points during product development, product data shall be integrated in a complete and consistent way. Note that even minor errors like, for example, a wiring harness of an automotive prototype, might lead to high costs, as construction costs of prototypes are very high. Due to heterogeneous product structures (e.g., list, hierarchy, or array representation of product parts) [4] as well as varying data quality, however, the integration of product data constitutes a challenging task. Especially, the identification of different artifacts related to the same real-world object is a cumbersome task that cannot be fully automated. As manual interaction is required, which is costly as well as time consuming, targeting at a full integration of all available product data from each application for all points in time is unfeasible.

In practice, an on-demand integration of subsets of product data is required, i.e., only those artifacts necessary to realize a particular business use case shall be integrated. As example consider the creation of a consistency check between requirements specifications on one hand and geometric models of product parts on the other.

To ensure for a high quality of incrementally integrated product data at a certain point in time, the progress of the integration process should be monitored based on proper quality metrics. Examples include metrics measuring the completeness of the correspondences between records stemming from different information systems. Note that data related to the same semantic concept may be documented in multiple information systems. Existing approaches provide integration quality metrics at the schema level, e.g., to assess the completeness of mappings between semantic concepts from different information systems. However, appropriate integration quality metrics at the data (i.e., instance) level are missing. Several challenges need to be tackled when aiming to measure the quality of product data integration. For example, companies may have numerous information systems maintaining thousands of product data artifacts. Furthermore, product data and corresponding attributes are recorded at different points in time. Finally, various stakeholders with different requirements may be involved in the integration process.

This paper addresses the following research questions: (1) How can the quality of product data integration be measured? (2) What are appropriate quality metrics for this purpose? Accordingly, the contribution of the paper are as follows: First, we present results from an in-depth requirements analysis for measuring the quality of integrated product data. Second, different quality metrics for measuring the integration of product data are elaborated. Third, we apply the metrics to a real-world case.

Section 2 presents our framework for product data integration required for understanding this work. In Section 3, we elicit requirements for measuring the quality of product data integration. To meet these requirements, Section 4 presents different metrics for assessing the progress and quality of incrementally integrated product data. A proof-of-concept prototype is presented in Section 5. In turn, in Section 6 we apply the metrics to a real-world case. Related work is discussed in Section 7. Section 8 summarizes the paper and gives an outlook.

2 A Framework for Product Data Integration

In order to integrate product data from heterogeneous, autonomous information systems several challenges need to be tackled [8]. Among others, one must cope with varying data quality, missing global identifiers, and differences regarding the management of product data variability and versions. To better understand these challenges, we analyzed a variety of information systems used in the context of product engineering by a German automotive original equipment manufacturer (OEM). Based on these practical insights as well as an in-depth literature study, we derived a framework for integrating heterogeneous product data. In detail, this framework relies on four *data integration layers*, i.e., product data collection layer, object layer, variant layer, and version layer [8].

2.1 Local and Global Product Ontology

In practice, different perspectives on product data are captured in application-specific data models. In turn, the various applications rely on different data management technologies, including relational databases, XML documents, and files. To cope with this heterogeneity, product data should be abstracted in a platform-independent way. For this purpose, in our framework product data is represented as reusable, interoperable, and platform-independent ontologies. Hence we apply the terms *schema concept* and *individual* from ontology engineering to express data model concepts as well as their extents.

A common architecture for integrating heterogeneous systems is to define a global schema into which schema concepts and corresponding data of information systems (denoted as local systems in the following) are integrated [2]. In the same manner, product data from a local information system may be abstracted into a *local product ontology* (*LPO*). The latter is then integrated into the *global product ontology* (*GPO*), which constitutes a holistic view on different product data aspects. Remember that a local information system solely maintains those parts (i.e., aspects) of the product data being relevant for specific stakeholders to accomplish their tasks (e.g., requirements engineer, CAD engineer, or test manager). In realistic scenarios, a multitude of local product ontologies needs to be integrated into the global product ontology.

Fig. 1 depicts an example of a local product ontology from the automotive domain.[4] In particular, a local information system maintains geometric models of electronic control units (ECUs). Furthermore, it uses specific techniques to store ECU variants and versions. These techniques are captured in the information systems' conceptual data models. Furthermore, semantic concepts of the latter are mapped onto hierarchically structured schema concepts of a local product ontology: A *product* consists of different *ECUs*. For each ECU, different *variants* (ECU Variant) and *versions* (ECU Version) are maintained. For example, a *car* requires an ECU controlling its *engine*. Different engine types, in turn, require appropriate variants of this ECU (*Diesel, Gas*). Finally, different versions of these ECU variants must be maintained (*1.0, 1.1, 2.0, 2.1*) as well.

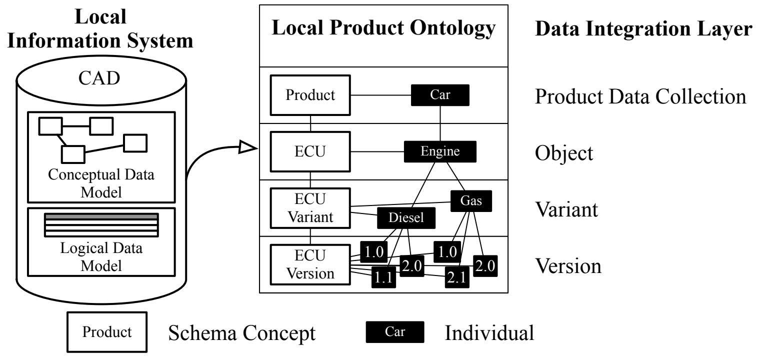

Fig. 1: Example local product ontology

Like local product ontologies, schema concepts and individuals of the global product ontology are organized in four data integration layers as well. Hence, it becomes possible to integrate product data for each layer separately. As a major challenge to be tackled when integrating individuals (i.e., data entities) from a local product ontology into the global product ontology, individuals related to the same real-world object need to be identified.

2.2 Schema Concept Rules and Actions

As there are usually only few schema concepts in a local product ontology that need to be integrated into the global product ontology, this part of the integration process can be performed manually by integration experts. In turn, there may exist thousands of individuals for each schema concept, especially regarding the version layer. Consequently, the integration of individuals should be automated as far as possible in order to reduce human efforts.

[4] To improve readability, attributes of the schema concepts as well as corresponding attribute values of the individuals are hidden.

As discussed, product data is maintained in heterogeneous and autonomous information systems, which were designed with a specific use case in mind (e.g., requirements engineering, computer-aided design). Typically, corresponding information systems obey specific conventions for labeling individuals. Existing approaches for mapping local with global concepts are usually based on string metrics (e.g., Levenshtein distance, Hamming distance) or phonetic algorithms (e.g, Metaphone [3]). If two information systems do not rely on the same naming convention, however, respective algorithms fail in finding correspondences between individuals. Hence, other techniques are required to integrate these individuals as well.

Record linkage techniques aim to find records stemming from different data sets that refer to the same entity [10]. Some of these techniques are based on rules related to attributes of records to define whether or not two records refer to the same entity. Similarly, for each schema concept of a local product ontology, *schema concept attribute rules* (*SCARs*) are defined to determine related individuals between local and global product ontology. In particular, these rules define relationships between *attributes* of local product ontology schema concepts and the ones of global product ontology schema concepts. Furthermore, for each SCAR, a *matching function* is defined that is applied during the integration process to identify correspondences between attribute values of individuals from a local and the global product ontology.

If the naming conventions for schema concepts from a local and the global product ontology are similar or equal, matching functions based on string metrics may be defined between attributes of these schema concepts. Typically, product data stemming from different information systems share common attributes (e.g., cross-references), which may be exploited as well. Furthermore, naming conventions may share patterns. Hence, SCARs that evaluate regular expressions for attribute values of individuals from both ontologies may be defined. Finally, if no matching functions can be defined based on string metrics or regular expressions, mapping tables must be maintained. In particular, the latter are lists of corresponding source and target attribute values.

In general, the integration of local product ontologies into the global product ontology is performed biliterally; i.e., there is a one-to-one mapping between schema concepts of local and global product ontologies on each integration layer. As example consider Figure 2. For each local product ontology, Fig. 2 depicts a schema concept and a corresponding individual. For the sake of simplicity, we restrict ourselves to schema concepts and individuals at the object layer.[5] The local product ontology on the left maintains geometric models of ECUs, whereas the local product ontology on the right captures corresponding requirement documentations of these ECUs. Both schema concepts consist of four attributes (*Label, Geom, Name, Doc*). In particular, for each local product ontology schema concept, a SCAR describes the relationship to a corresponding schema concept in the global product ontology.

[5] SCARs for the other data integration layers are defined accordingly.

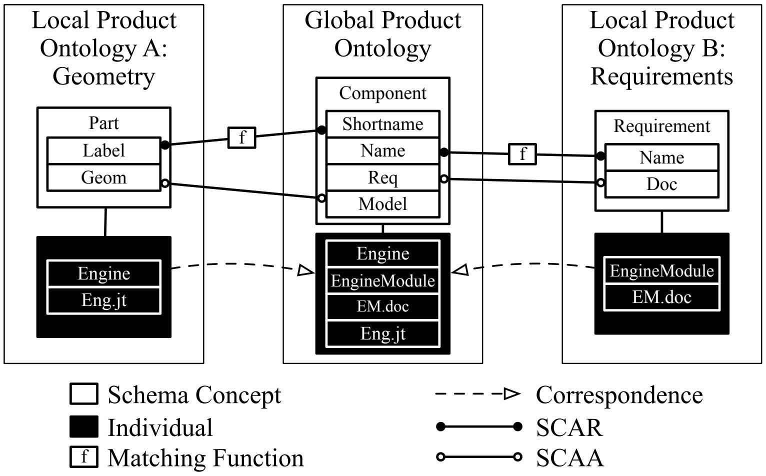

Fig. 2: SCARs and SCAAs between local and global product ontology

Since the global product ontology provides a holistic view on all product data perspectives (e.g., requirements, geometric models, or source code), mappings between attributes of local product ontology schema concepts to corresponding ones in the global product ontology must be defined. While a SCAR corresponds to a rule that relates individuals based on their schema concepts, a *schema concept attribute action* (*SCAA*) defines those attributes that should be integrated into the global product ontology after identifying correspondences between individuals.

For each local product ontology schema concept in Fig. 2, an SCAA is defined. In particular, attribute values (*Geom*) from individuals of *Part* (LPO A) will be copied into corresponding values (*Model*) from individuals of *Component* (GPO). In the same way, attribute values (*Doc*) from individuals of *Requirements* (LPO B) will be copied into corresponding ones (*Req*) of *Component.*

2.3 Integration Set

As a prerequisite for defining metrics measuring the quality of product data integration, several terms need to be defined. The quality metrics are based on the notion of *integration set*, which consists of a local product ontology that shall be integrated into a global product ontology. Correspondences between individuals from both ontologies are identified through the aforementioned schema concept attribute rules. In the following, the integration set as well as additional functions necessary for measuring the quality of product data integration are defined.

Definition 1 (Integration Set). An integration set $IS := (LPO, GPO, M, COR)$ is a quadruple with the following properties:

- $LPO := (SC_{LPO}, IND_{LPO}, Attr_{LPO}, AttrVal_{LPO}, HierarchySC_{LPO}, HierarchyIND_{LPO}, LayerSC_{LPO}, SCAttr_{LPO}, INDAttrVal_{LPO}, Member)$ defines a local product ontology, where
  - SC_{LPO} is a set of schema concepts,
  - IND_{LPO} is a set of individuals,
  - $Attr_{LPO}$ is a set of attributes,
  - $AttrVal_{LPO}$ is a set of attribute values,
  - $HierarchySC_{LPO} \subset SC_{LPO} \times SC_{LPO}$ is a set of directed edges representing the hierarchy of the schema concepts in SC_{LPO},
  - $HierarchyIND_{LPO} \subset IND_{LPO} \times IND_{LPO}$ is a set of directed edges representing the hierarchy of the individuals in IND_{GPO},
  - $LayerSC_{LPO}$: $SC_{LPO} \rightarrow \{PDC, OBJ, VAR, VER\}$ assigning a layer to each schema concept,
  - $SCAttr_{LPO} \subset SC_{LPO} \times Attr_{LPO}$ is a set of directed edges between schema concepts and attributes,
  - $INDAttrVal_{LPO} \subset IND_{LPO} \times Attr_{LPO} \times \mathcal{D}$ is a set of directed edges between attributes of individuals to attribute values,
  - $Member_{LPO} \subset SC_{LPO} \times IND_{LPO}$ is a set of directed edges between schema concepts and individuals
- $GPO := (SC_{GPO}, IND_{GPO}, HierarchySC_{GPO}, HierarchyIND_{GPO}, LayerSC_{GPO}, Attr_{GPO}, AttrVal_{GPO}, Member)$ defines a global product ontology[6]
- $M := (SCAR, SCAA)$ defines the mapping between local and global product ontology where
  - $SCAR \subset SC_{LPO} \times SC_{GPO}$ is the a of schema concept attribute rules,
  - $SCAA \subset SC_{LPO} \times SC_{GPO}$ is the a of schema concept attribute actions
- $COR \subset IND_{LPO} \times IND_{GPO}$ is a set of correspondences between individuals from a local and the global product ontology
- $GetSC_{LPO}(l) := \{sc \in SC_{LPO} \mid LayerSC_{LPO}(sc) = l \wedge l \in \{PDC, OBJ, VAR, VER\}\}$ returns the schema concepts corresponding to a particular layer,
- $GetIND_{LPO}(sc) := \{ind \in IND_{LPO} \mid \exists\ m = (sc, ind) \in Member_{LPO}\}$ corresponds to the set of individuals associated with a given schema concept,
- $L2GSC(sc_{LPO}) := \{sc_{GPO} \in SC_{GPO} \mid \exists\ scar = (sc_{LPO}, sc_{GPO}) \in SCAR\}$ corresponds to the schema concept of a global product ontology onto which a schema concept of a local product ontology is mapped, and
- $GetSCAR(sc_{LPO}, sc_{GPO}) := \{scar \in SCAR \mid scar = (sc_{LPO}, sc_{GPO})\}$ corresponds to the set of schema concept attribute rules between a schema concept from local and global product ontologies.

2.4 Initial Integration

The integration of individuals from local product ontology schema concepts with those of the global product ontology should be automated where possible. As

[6] Definitions of the components are similar to the ones of the LPO and are omitted for the sake of space.

local and global product ontology are structured in the same way, SCARs can be evaluated on each data integration layer. In general, the execution of the integration process is performed top-down, i.e., SCARs between the schema concepts of a local and the global product ontology located at the product data collection layer are evaluated first. Then, SCARs related to the object layer are evaluated, and so forth.

Algorithm 1 depicts this initial integration process. As input, an integration set IS is chosen (cf. Definition 1). It consists of a local and global product ontology for which a complete schema concept mapping exists, i.e., for each schema concept of the local ontology there exists at least one schema concept attribute rule as well as an action. Algorithm 1 returns a set of correspondences between individuals of schema concepts from the two ontologies.

```
   Input: Integration set $IS$
   Result: Set of correspondences $COR$
 1 foreach $layer \in \{PDC, OBJ, VAR, VER\}$ do
 2     foreach $sc_{LPO} \in GetSC_{LPO}(layer)$ do
 3         $L^{IND} = GetIND_{LPO}(sc_{LPO})$;
 4         $sc_{GPO} = L2GSC(sc_{LPO})$;
 5         $G^{IND} = GetIND_{GPO}(sc_{GPO})$;
 6         $SCAR^{L,G} = GetSCAR(sc_{LPO}, sc_{GPO})$;
 7         foreach $l \in L^{IND}$ do
 8             foreach $g \in G^{IND}$ do
 9                 $COR_{l,g} = evaluateSCAR(l, g, SCAR^{L,G})$;
10                 if $COR_{l,g} \neq \emptyset$ then
11                     $executeSCAA(COR_{l,g})$;
12                     $COR = COR \bigcup COR_{l,g}$;
13                 end
14             end
15         end
16     end
17 end
```

Algorithm 1: Algorithm for the initial integration of an integration set

In particular, the integration of the local product ontology with the global one is performed for each data integration layer separately (Line 1), starting with the product data collection layer. For each schema concept of the local product ontology of the given layer (Line 2), its associated individuals are identified (Line 3). Then, the corresponding schema concept of the global product ontology as well as its individuals are obtained (Lines 4 and 5). Next, the schema concept attribute rules between both schema concepts are determined (Line 6). The cartesian product of all individuals of a schema concept from the local product ontology and all individuals of the corresponding one from the global product ontology is created (Lines 7 and 8). For each member of this product, the previously defined SCARs are then evaluated (Line 9).

If a correspondence between individuals l and g has been identified (Line 10), the corresponding schema attribute actions are executed (Line 11) and the correspondence is added to the result set (Line 12).

The previous steps will be repeated for each product data integration layer. After completing the integration process, individuals of a local product ontology are linked to individuals from the global product ontology. Typically, a local product ontology comprises only a subset of the product data. Therefore, multiple local product ontologies need to be integrated to obtain a holistic view.

Various stakeholders and users, who perform different tasks, are involved in this integration process. For instance, *domain experts* are responsible for defining mappings between the information systems' conceptual data models and the corresponding local product ontologies. Furthermore, they specify SCARs and SCAAs between schema concepts of local and global product ontology. This task is supported by *integration experts* that have an in-depth knowledge of interdependencies between schema concepts of the different information systems. Due to the complex nature of product data, the integration cannot be fully automated; partially, manual interaction is required, which causes high efforts.

In general, the integration algorithm may not find all corresponding individuals between local and global product ontology. Hence, *data quality experts* maintain correspondences to assure a complete and consistent integration. Finally, *end users* utilize the integrated product data from the global product ontology to realize different business use cases (e.g., management report).

3 Requirements

To support the needs of the different stakeholders involved in the integration process, metrics that allow assessing data integration quality become necessary. This section elicits the requirements for respective metrics. For this purpose, we analysed development processes for electrical and electronic components (e.g., electronic control units, sensor, and actuators) at a german automotive OEM. This includes expert interviews as well as an in-depth survey of numerous information systems maintaining product data.

Requirement 1 (Aspects). Several stakeholders are involved in the integration process (e.g., domain experts, integration experts, data quality experts, and end users) raising different requirements for measuring the quality of integrated product data. For example, domain and integration experts focus on integrating schema concepts from a local and the global product ontology and, hence, need quality metrics taking SCARs and SCAAs between schema concepts into account. In turn, end users are solely interested in individuals of the global product ontology. Thus, it should be possible to measure the quality of product data integration for different aspects of an integration set (i.e., schema concepts, individuals, or attribute values).

Requirement 2 (Perspective). Furthermore, stakeholders have different perspectives on the integration process. Domain experts and data quality experts are responsible for maintaining local product ontology aspects (i.e., schema concepts, individuals, and attributes) in relation to the global product ontology. In turn, end users consider global product ontology individuals related to a set of local product ontology. Consequently, it should be possible to measure the integration quality of integrated product data from different perspectives (local to global and vice versa).

Requirement 3 (Scope). Product data evolve over time and, hence, have different lifecycle states (e.g., *specified, designed, implemented, integrated, and released*). Since product data is managed by heterogeneous, autonomous information systems that use different techniques for dealing with product data variants and versions, entirely integrating product data at all points in time is too costly. Therefore, it should be possible to define the scope of integration quality metrics.

Requirement 4 (Monitoring). Quality gates are milstones in product development processes at which predefined requirements must be fulfilled. In the same way, it should be possible to specify reference values along the lifecycle of product data for which predefined values for different integration quality metrics must be reached. If the actual values of the quality metrics deviate from a pre-defined reference value, countermeasures may be performed to still achieve a complete and consistency integration set at a specific point in time.

4 Measuring the Quality of Product Data Integration

Quality metrics enable the different stakeholders of the product data integration process to monitor their tasks. Considering the requirements elicited in Section 3, we present metrics measuring the quality of product data integration grouped along different viewpoints (local-to-global vs. global-to-local) (cf. Req. 2) to support the various stakeholders in performing their integration tasks.

Section 4.1 presents metrics measuring the integration quality of a single local product ontology. Section 4.2 then deals with metrics that measure the integration quality of the global product ontology with respect to a given set of local product ontologies. For both viewpoints, quality metrics for different integration aspects are presented (cf. Req. 1).

4.1 Local-to-Global Mapping

Local Schema Concept Completeness. Domain and integration experts are responsible for maintaining mappings between local and global product ontology schema concepts (cf. Sect. 2). As the initial integration process may only be executed if for each schema concept there exists at least one schema concept attribute rule and action, their completeness must be determined. Hence, for a given integration set *IS* (cf. Definition 1), *local schema concept completeness* (*LocSCComp*) relates the set of schema concepts with at least one schema concept attribute rule and action (MPD^{SC}) to all schema concepts of a local product ontology (SC_{LPO}). If both sets are the same, the integration set is denoted as *local schema complete* (i.e., $LocSCComp = 1$):

$$LocSCComp(IS) = \frac{|MPD_{LPO}^{SC}|}{|SC_{LPO}|} \in [0,1], \text{ where}$$
$$MPD_{LPO}^{SC} = \{sc_{LPO} \in SC_{LPO} \mid \exists\ sc_{GPO} \in SC_{GPO} \wedge$$
$$\exists\ scar = (sc_{LPO}, sc_{GPO}) \in SCAR \wedge \exists\ scaa = (sc_{LPO}, sc_{GPO}) \in SCAA\}$$

Local Individual Completeness. As the integration of product data cannot be fully automated, data quality experts must maintain correspondences between local and global product ontology individuals identified during the initial integration process. To measure the progress of the integration process, a specific data quality metric becomes necessary taking the correspondences between individuals from the local and global product ontologies into account.

Thus, for a given integration set (IS), *local individual completeness* (*LocIndComp*) is defined as the ratio of individuals of a local product ontology with at least one correspondence to an individual of the global product ontology (COR_{LPO}^{IND}) related to all individuals of the local product ontology (IND_{LPO}). If both sets are equal, the integration set is denoted *local individual complete* (i.e., $LocIndComp = 1$):

$$LocIndComp(IS) = \frac{|COR_{LPO}^{IND}|}{|IND_{LPO}|} \in [0,1], \text{ where}$$
$$COR_{LPO}^{IND} = \{a_{LPO} \in IND_{LPO} \mid \exists\ c = (a_{LPO}, b_{GPO}) \in COR\}$$

4.2 Global-to-Local Mapping

The metrics presented so far focus on the integration quality of a local product ontology with respect to the given global product ontology. End users, in turn, are solely interested in individuals of the global product ontology being completely and consistently integrated, i.e., all necessary attribute values from global product ontology individuals are recorded to enable particular use cases.

Global Individual Completeness. The main goal of product data integration is to enable sophisticated business use cases. The latter include, for example, the creation of physical mock-ups. Since production costs of such mock-ups are very high, errors (e.g., missing attributes, inconsistent attributes) during product data integration should be avoided. Hence, each individual of the global product ontology needs to be linked to at least one individual of a local product ontology.

Formally, *global individual completeness* (*GlobalIndComp*) of multiple integration sets $IS_1, \ldots, IS_n$ corresponds to global product ontology individuals linked to at least one local product ontology individual (MPD_{GPO}^{IND}) related to all global product ontology individuals (IND_{GPO}). If both sets are equal, the integration sets is denoted *global individual complete*:

$$\mathit{GlobalIndComp}(IS_1,\ldots,IS_n) = \frac{|MPD^{IND}_{GPO}|}{|IND_{GPO}|} \in [0,1], \text{ where}$$
$$MPD^{IND}_{GPO} = \{ind_{GPO} \in IND_{GPO} \mid$$
$$\exists\, c = (ind_{LPO}, ind_{GPO}) \in COR_1 \cup \ldots \cup COR_n\}$$

Global Individual Attribute Completeness. Product data evolves over time and, therefore, their attributes are captured at different points in time. Though, there may be correspondences between individuals from the local product ontology and the global one. Since attribute values of individuals from local product ontologies may have not been set yet, corresponding attribute values of individuals from the global product ontology cannot be set as well. As these attribute values might be necessary to realize a particular business use case, the completeness of individual attribute values from the global product ontology must be determined. In particular, the attribute value of an individual from the global product ontology is complete, if it is not empty.

Therefore, *global individual attribute completeness* (*GlobalIndAttrComp*) of multiple integration sets $IS_1,\ldots,IS_n$ corresponds to the ratio of those individuals of a global product ontology, where each attribute value is not empty ($COMPLATTR^{IND}_{GPO}$) related to all individuals of a global product ontology (IND_{GPO}). If both sets are equal, the integration set is denoted *global attribute complete*:

$$\mathit{GlobalIndAttrComp}(IS_1,\ldots,IS_n) = \frac{|COMPLATTR^{IND}_{GPO}|}{|IND_{GPO}|} \in [0,1], \text{ where}$$
$$COMPLATTR^{IND}_{GPO} = \{i \in IND_{GPO} \mid \exists\, sc \in SC_{GPO} \wedge \exists\, m \in Member(sc,i)$$
$$\wedge\, \forall a = (sc, attr) \in SCAttr_{GPO}\ \exists\, v = (i, attr, val) \in INDAttrVal_{GPO}\}$$

Global Individual Attribute Consistency. After applying the initial integration process (cf. Section 2.4) there may be correspondences between multiple individuals from local product ontologies to a single individual from the global product ontology. This will be the case if attributes describing the same real-world object are documented in multiple information systems. As changes of corresponding attribute values performed in one these information systems are not always propagated to the other ones maintaining the same attribute, integration conflicts might occur during the integration process. Hence, we need an appropriate quality metric to detect such conflicts.

In particular, *global individual attribute consistency* (*GlobalIndAttrCons*) of multiple integration sets $IS_1,\ldots,IS_n$ corresponds to the difference between all individuals of a global product ontology (IND_{GPO}) and the set of inconsistent individuals ($INCON^{IND}_{GPO}$). The latter consists of all individuals, for which there are at least two corresponding individuals ind_k and ind_l from two local product ontologies LPO_k and LPO_L having SCAAs on the same attribute $attr_{GPO}$ defined, while the attribute values for ind_k and ind_l are different for this attribute.

If $INCON_{GPO}^{IND}$ is an empty set, the integration set is denoted as *global attribute consistent*:

$$GlobalIndAttrCons(IS_1, \ldots, IS_n) = \frac{|IND_{GPO} \setminus INCON_{GPO}^{IND}|}{|IND_{GPO}|} \in [0,1], \text{ where}$$

$$\begin{aligned} INCON_{GPO}^{IND} = \{ & i \in IND_{GPO} \mid \exists\, scaa_k = (attr_{LPO}^k, attr_{GPO}) \in SCAA_k \\ & \wedge \exists\, scaa_l = (attr_{LPO}^l, attr_{GPO}) \in SCAA_l \\ & \wedge \exists\, a = (ind_k, i) \in COR_k \wedge \exists\, b = (ind_l, i) \in COR_l \\ & \wedge \exists\, v_1 = (ind_k, attr_{LPO}^k, av_1) \in INDAttrVal_{LPO} \\ & \wedge \exists\, v_2 = (ind_l, attr_{LPO}^l, av_2) \in INDAttrVal_{LPO} \wedge av_1 \neq av_2 \wedge 1 \leq k, l \leq n \} \end{aligned}$$

As discussed, integrating all available product data would be too costly. In practice, therefore, only subsets of product data are integrated, i.e., only those local product ontology individuals are integrated into the global product ontology necessary to enable relevant business use cases. As our metrics are based on set theory and any subset of an ontology is again an ontology, the metrics may be applied to arbitrary integration sets (cf. Requirement 3).

4.3 Reference Values

The quality metrics defined in Section 4.1 and 4.2 considered the current state of the local product ontologies and the global one. To also enable use cases that consider the quality of integrated product data at a specific point in time (denoted as t_{end}), the integration set should be *global individual complete*, *global attribute complete*, and *global attribute consistent.* As the integration process requires manual interaction (e.g., to maintain correspondences between local product ontologies and the global one), its progress needs to be monitored in order to guarantee these quality properties at t_{end} (cf. Req. 4). Hence, *reference values* are defined representing benchmark values for the different integration quality metrics to be met at certain points in time $t_1, \ldots, t_n$; $t_i < t_{end}$, $i = 1, \ldots, n$.

As example consider Figure 3, which depicts the evolution of the three different integration quality metrics over time. In particular, the solid line illustrates the *global individual completeness* values for integration sets IS_1 and IS_2, while the dashed line represents the *global individual attribute completeness* values for the same sets. Finally, the dotted line illustrates *global individual attribute consistency* values. Furthermore, reference values are defined at two points in time t_1 and t_2. In particular, squares represent reference values for the first, diamonds the ones for the second, and circles the ones for the third quality metric. Note that all curves have positive and negative gradients (e.g., deletion of correspondences). Except the *global individual attribute completeness* value at t_1, the other values fall below the predefined reference values. Hence, countermeasures need to be performed to be still able to achieve the required quality properties (global individual complete, global attribute complete, and global attribute consistent) for the integration sets at t_{end}.

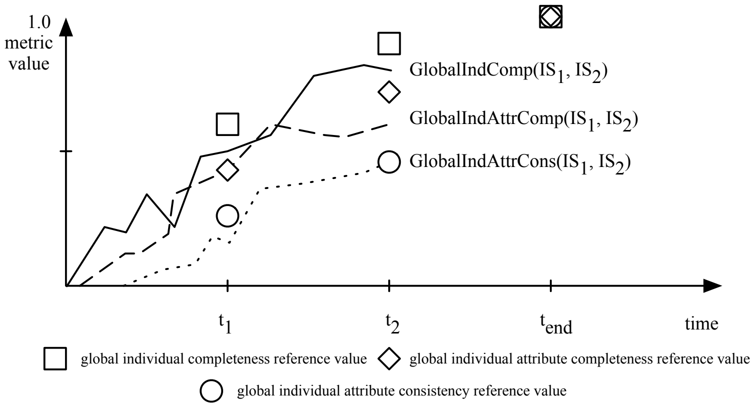

Fig. 3: Integration quality metrics over time in relation to reference values

5 Proof-of-Concept Prototype

The presented integration process as well as integration quality metrics have been implemented as a plugin for the Protégé ontology editor [15]. In particular, local and global product ontologies are represented as OWL2 ontologies [16]. To separate integration knowledge from schema concepts and individuals, SCARs, SCAAs, and resulting correspondences are maintained in a separate OWL ontology, denoted as *mapping ontology*. In particular, schema concepts are modelled as OWL2 classes, whereas SCARs and SCAAs are modelled as object types between OWL2 classes. We implemented the integration quality metrics based on semantic web rule language (SWRL) rules [17] to gather the different sets (e.g., COR_{LPO}^{IND}, $INCON_{GPO}^{IND}$). Applying these rules in a real world case study (cf. Sect. 6) revealed their limitation to ontologies with only of limited set of individuals. Consequently, we implemented the integration quality metrics with imperative functions based on the OWL API [18].

6 Case Study

The previously presented metrics have been applied in a real-world case study at a large German automotive OEM. During the development of a car, usually, mock-ups are produced to identify problems (e.g., packaging, functions) in an early development phase.

Modern cars consist of numerous electrical and electronic (E/E) components (ECUs, sensors, actuators) that enable safety systems (e.g., electronic stability control, collision avoidance system) as well as comfort systems (e.g., navigation devices, infotainment). While mechanical parts are described with geometric models, in turn, E/E components are described by multiple aspects. This includes, for instance, geometric models, funcional models, and software. Consequently, the integration of complex product data is crucial for creating prototypes. Note that a single error in one component might cause high financial losses since construction costs of prototypes are high.

The case study focuses on the practical use of the presented quality metrics on product data integration. More precisely we consider the integration of E/E components from three heterogeneous information systems: The first system maintains geometric models, while the second one stores hardware and software information; the third system captures the signals exchanged between E/E components. For each of the three information systems, local product ontologies were created (i.e., *LPO1, LPO2, LPO3*). The resulting schema concepts are depicted in Figure 4.

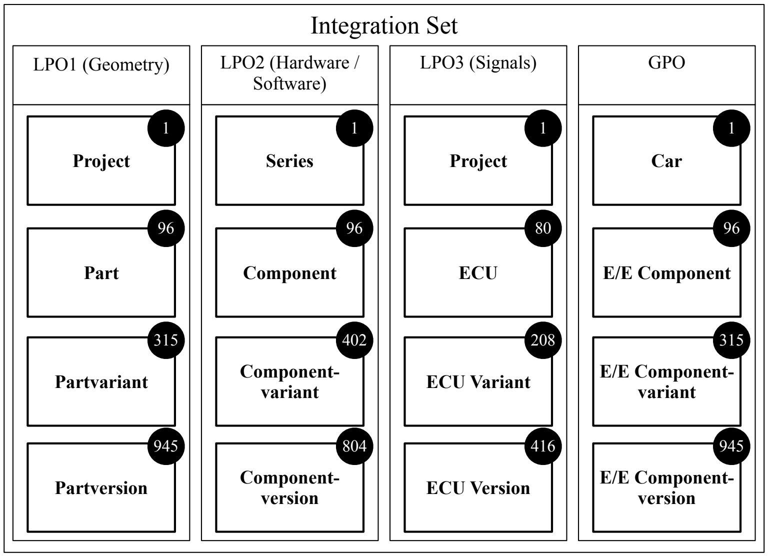

Fig. 4: E/E product data integration schema concepts and # of individuals [7]

Each local product ontology comprises of one schema concept at each data integration layer. Furthermore, each schema concept is populated with corresponding individuals. Since product data of one specific car have to be integrated, there is one individual for each schema concept at the product data collection layer. There are 96 different *parts* with 315 *partvariants*, and 945 *partversion* in the local product ontology for geometric models (*LPO1*). 96 *components* with 402 *component variants*, and 804 *component versions* (*LPO2*) as well as 80 *ECU*, 208 *ECU variants*, and 416 *ECU versions* (*LPO3*). Finally, schema concepts of the global product ontology *GPO* are defined (i.e., *CAR*, *E/E Component*, *E/E Componentvariant*, and *E/E Componentversion*). Furthermore, schema concept attribute rules and actions are defined between LPOs and the GPO. The global product ontology is populated with individuals from *LPO1*, which provides prod-

[7] The circles at each schema concept show the number of corresponding individuals.

uct data with the best accuracy. In detail, individuals at the object layer are labeled to comply with the pre-specified labeling schema.

Table 1 shows the values of the quality metrics after an initial integration (t_{init}). While 86 percent of the individuals of *LPO1* could be automatically mapped to corresponding ones of *GPO*, only 41 percent of the individuals of *LPO3* could be linked. The source information system of *LPO3* uses abbreviations to label individuals. Hence, no schema concept attribute rules evaluating the labels of individuals based on string metrics could be defined between *LPO3* and *GPO*. In turn, a mapping table (cf. Sect. 2.2) relating individual labels from *LPO3* to ones from *GPO* were established.

Quality Metric	t_{init}
GlobalIndComp	0.68
GlobalIndAttrComp	0.53
GlobalIndAttrCons	0.84
LocIndComp(LPO1)	0.86
LocIndComp(LPO2)	0.55
LocIndComp(LPO3)	0.41

Table 1: Integration quality after initial integration

Altogether, the following lessons learned resulted from the case study. The presented integration quality metrics could be successfully applied in practice. In particular, the latter allowed assessing the initial integration process and could be used to monitor the progress of product data integration over time. However, defining reference values for different integration quality metrics required experience in product data integration. Furthermore, the quality of the initial integration is related to the one of the product data sources to be integrated. In general, the quality of product data attributes in early stages of the product development lifecycle is rather low (e.g., missing attribute values, deprecated values). Consequently, the manual interaction efforts (e.g., maintaining correspondences between local and global product ontologies) will be higher compared to the integration of mature product data.

7 Related Work

We argued that a full integration of all product data available in any information system at any point in time is unfeasible. Similarly, [1] argues that data inconsistencies are common and hence should be tolerated. As opposed to our integration framework, the approach presented in [1] neither builds upon a global integration system nor a common data structure. Further, no quality metrics are provided.

In [9], different information quality metrics are introduced. First, *schema completeness* is defined as the ratio of its distinctive schema concepts to all schema concepts of a data integration system. Second, *schema data type consistency* is introduced as the total number of consistent attributes in relation to all attributes. Finally, the authors define *schema minimality* based on redundancy values of the entities and relations of a schema. These metrics are similar to our definitions, but neither take data completeness and consistency nor the definition of reference values into account.

In [12], the approach presented in [9] is extended with *schema structurality* and *schema proximity*. The definitions compare schema integration results. On one hand, the latter are automatically integrated by a tool, on the other they are integrated manually by experts. In detail, schema completeness is the proportion of the intersection of entities from the automatically generated schema related to all entities of the manually created schema. In our framework, the schema integration is performed manually by integration experts. Consequently, these measures cannot be applied. Furthermore, measuring the integration of data and reference values are not considered as well.

[11] proposes five quality criteria for data integration: *schema completeness*, *schema consistency*, *mapping consistency accuracy*, *minimality*, and *performance*. The approach introduces metrics measuring the integration between local schemas and a global schema. Therefore, it is related to our approach. Nevertheless, quality metrics concerning data integration are missing as well.

Altogether, we elaborated metrics measuring the quality of product data integration for different aspects (schema concepts, individuals, and attributes). While contemporary approaches solely focus on schema concepts, we provided further metrics taking the quality of data integration into account as well.

8 Summary and Outlook

In this paper, various quality metrics for measuring the integration of product data were elaborated. Based on an in-depth analysis of information systems maintaining product data at a German automotive OEM, we elicited the fundamental requirements for measuring product data integration. Furthermore, different metrics measuring the quality of product data integration were elaborated. To meet the requirements of the users involved in the integration process (e.g., domain experts, integration experts, data quality experts, and end users), appropriate quality metrics were introduced. In particular, we introduced metrics measuring product data integration of local product ontologies in relation to the global product ontology and vice versa. The presented metrics are generic in the sense that they can be applied to arbitrary subsets of local and global product ontologies. Finally, we suggested defining reference values and applied the metrics in a real-world case study. In future work, we will apply the presented metrics in further case studies from diverse domains and integrate them with product data management tools.

References

1. Easterbrook, S., Finkelstein, A., Kramer, J., Nuseibeh, B.: Coordinating Distributed ViewPoints: the anatomy of a consistency check. CERA 2(3), 209-222 (1994)
2. Wache, H. et al.: Ontology-based Integration of Information - A Survey of Existing Approaches. In: Proc. IJCAI-01 Workshop, pp. 108-117 (2001)
3. Philips, L.: Hanging on the Metaphone. Computer Language 7(12), 39-44 (1990)
4. Stark, J.: Product Lifecycle Management. Springer (2011)
5. Wiederhold, G., Qian, X.: Consistency Control of Replicated Data in Federated Databases. In: Workshop on the Management of Replicated Data, pp. 130-132 (1990)
6. Sheth, A.P., Rusinkiewicz, M.: Management of Interdependent Data: Specifying Dependency and Consistency Requirements. In: Workshop on the Management of Replicated Data, pp. 133-136 (1990)
7. Wiederhold, G., Qian, X.: Modeling Asynchrony in Distributed Databases. In: Proc. ICDE'87, pp. 246-250 (1987)
8. Tiedeken, J., Reichert, M., Herbst, J.: On the Integration of Electrical/Electronic Product Data in the Automotive Domain. Datenbank Spektrum 13(3). 189-199 (2013)
9. Batista, M.d.C.M., Salgado, A.C.: Information Quality Measurement in Data Integration Schemas. In: Proc. QDB'07, pp. 61-72 (2007)
10. Herzog, T. N., Scheuren, F.J., Winkler, W.E.: Data Quality and Record Linkage Techniques. Springer (2007)
11. Wang, J.: A Quality Framework for Data Integration. In: Proc. BNCOD'10, pp. 131-134 (2010)
12. Duchateau, F., Bellahsene, Z.: Measuring the Quality of an Integrated Schema. In: Proc. ER'10, pp. 261-273 (2010)
13. Roland Berger Strategy Consultants. Mastering Product Complexity, Düsseldorf, November 2012
14. Wang, R.Y., Strong, D.M.: Beyond Accuracy: What Data Quality Means to Data Consumers. J. of Management Information Systems 12(4), 5-33 (1996)
15. Gennari, J. H. et al.: The evolution of Protégé: an environment for knowledge-based systems development. Int. J. Hum.-Comput. Stud. 58(1), 89-123 (2003)
16. Motik, B. et al.: OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3C recommendation 27.65 (2009)
17. Horrocks, I. et al.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004
18. Horridge, M., Bechhofer, S.: The OWL API: A Java API for OWL Ontologies. Semantic Web 2(1), 11-21 (2011)