Advanced Engineering Informatics 46 (2020) 101102

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Enriching standards-based digital thread by fusing as-designed and as-inspected data using knowledge graphs

Soonjo Kwon[a,b], Laetitia V. Monnier[a,c], Raphael Barbau[a,d], William Z. Bernstein[e,*]

[a] Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
[b] Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
[c] Université de Bourgogne, Franche-Comté, Dijon, France
[d] Engisis LLC, Bethesda, MD 20817, USA
[e] Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA

ARTICLE INFO

Keywords:
Digital thread
Knowledge graph
Ontology
Industrial data standard
Product lifecycle
Data fusion

ABSTRACT

Realizing the digital thread is essential for linking and orchestrating data across the product lifecycle in smart manufacturing. Linking heterogeneous lifecycle data is critical to maintain associativity and traceability in a digital thread. Recently, researchers have successfully leveraged ontology models with knowledge graphs in engineering domains for threading different lifecycle data. One of the most successful of such efforts is OntoSTEP which enables the formal capture of information embedded in the STandard for Exchange of Product model data (STEP) data representation, or ISO 10303. Meanwhile, an emerging inspection standard, called the Quality Information Framework (QIF), has garnered significant attention as it can bring quality information into the digital thread. Implementing more automated methods for product quality assurance is challenging due to the lack of unified information models from design to inspection. To this end, we propose an approach to fuse as-designed data represented in STEP and as-inspected data represented in QIF in a standards-based digital thread based on ontology with knowledge graphs. Specifically, we present an automated pipeline for generating knowledge graphs representing STEP and QIF data, a mapping implementation to integrate STEP and QIF knowledge graphs, and rules and queries to demonstrate the integration's potential for better decision making with respect to product quality assurance.

1. Introduction

Smart manufacturing technologies fuse advanced manufacturing capabilities with digital technologies to improve agility, productivity, efficiency, and sustainability of production systems [1]. Central to the smart manufacturing concept, often referred to as the next industrial revolution, are standards activities. Standards, including formal data representations, instructive guidelines, and reference architectures, position the research and industry communities on common footing. Though the terms, "smart manufacturing" and "Industry 4.0" are relatively new, the standards activities relevant to the realization of their potential are not.

For example, the STandard for the Exchange of Product model data (STEP), or the International Organization of Standardization (ISO) 10303 series [2], has been the focal point of an active working group for three decades [3]. Engineering firms have realized the benefits of a standard exchange format for design information with STEP in the context of engineering design practice. However, not until recently have the full benefits of such a digital representation been uncovered. Formally representing design data, including Geometric Dimensioning and Tolerancing (GD&T) information, helps connect the *digital thread*, promoting a common context across the product development process, from its initial conceptual design, across its production, and through its sustainment in use.

One of the mechanisms for linking such concepts is through ontology with Knowledge Graphs (KGs). Many recent studies have used the term *knowledge graph* to emphasize the relationships between knowledge entities represented as a graph in an *ontology*. Though there may be multiple interpretations between the two terms, the term *ontology* is usually preferred for taxonomy (or schema, T Box) which includes types, properties and relationships between entities, whereas the term *knowledge graph* is used for information (or instances, A Box) in the form of triplets (subject, predicate, object) with values which can derive new knowledge using a reasoner [4,5]. With the more flexible data

* Corresponding author.
E-mail address: wzb@nist.gov (W.Z. Bernstein).

https://doi.org/10.1016/j.aei.2020.101102
Received 24 January 2020; Received in revised form 24 March 2020; Accepted 17 April 2020
1474-0346/ Published by Elsevier Ltd.

querying and fusing mechanisms that ontology representations provide, researchers and practitioners have successfully leveraged the engineering-based ontology with KGs for threading the digitization of the product lifecycle, including automatically generating tolerancing recommendations [6], building data dictionaries for engineering materials [7], and realizing more intelligent disassembly patterns [8], to name a few. Encapsulating digital thread concepts, OntoSTEP [9,10] formally captures and represents elements and attributes embedded in the STEP data representation. The goal of OntoSTEP is to facilitate database design/construction, information retrieval, and product-based reasoning based on ontology with KGs.

Partly inspired by these achievements, we provide such functionality with an emerging inspection (and quality assurance) standard, called the Quality Informatino Framework (QIF) developed through the Digital Metrology Standards Consortium (DMSC). QIF provides a collection of data dictionaries to formally characterize inspection plans, rules, and results in a consistent manner across engineering domains and disciplines [11]. The dissemination of the QIF standard is spreading quickly throughout the manufacturing sector, presenting an opportunity to provide additional computer-enabled tools for realizing its full potential. A recent mapping specification between STEP and QIF [12] also signals the probable adoption of QIF representations across standards-driven Model-Based Enterprise (MBE) practices.

In this work, we present (1) an automated pipeline for generating KGs representing STEP- and QIF-driven data, (2) a mapping implementation between STEP and QIF KGs, and (3) sample rules and queries to demonstrate such integration's potential to facilitate rapid and more seamless reasoning for design decisions and product quality assurance. We anticipate that this demonstration will lead to more diverse use of the inspection-driven KGs together with design KGs.

2. Background and related work

Here, we comment on emerging standards, concepts, methods related to product quality and to the larger vision of smart manufacturing. We anticipate other researchers to take advantage of our presented pipeline to influence their own inspection-relevant pipeline with more intelligence.

2.1. Model-based definition

Model-Based Definition (MBD) is a strategy of defining a three-dimensional (3D) digital model, formally characterizing all of the associated requirements of a product. MBE uses MBD, rather than paper-based drawings and documentation, as the main data source for all engineering activities throughout the product lifecycle [13,14]. Commercial computer-aided design (CAD) systems have adopted MBD strategy in recent years, and they also support standard formats like STEP Application Protocol (AP)242 which can store MBD without any loss of information.

Camba et al. [15] proposed "extended annotation" to widen the scope of the annotation capabilities available in the current MBE approach by allowing users to explicitly communicate geometric design intent. Huang et al. [13] pointed out the lack of feature information in the current MBD approach and proposed a multi-level "structuralized MBD model" to capture machining semantics information in a hierarchical way. Hallmann et al. [16] presented a method to link 3D tessellated geometries with product and manufacturing information (PMI) assigned on the exact CAD geometry (i.e., boundary representation, B-rep) of STEP.

Some researchers surveyed the impact of applying MBD to industry-driven practices. Comparing model-based processes to paper-based methods, Hedberg et al. [14] reported measured results to identify the benefits and drawbacks for implementing MBD in industry. Ruemler et al. [17] surveyed a similar pool of users to better understand the needs for the common information model across the product lifecycle. The results highlight evidence that industrial practitioners are potentially embracing MBD. However, more workflow-specific information is still required to establish a common information model.

There are still barriers in terms of the tools, standards, and processes to fully support model-based data interoperability from design through manufacturing to inspection across the supply chain. Trainer et al. [18] claimed that an open standards method for MBD interoperability will bring maximum value to industry. Here, one of the major challenges is to define a unified information model that can be utilized throughout the product lifecycle.

2.2. Digital thread

The digital thread can be defined as the ensemble of data that enables the combination of MBD, manufacturing, and inspection. Digital thread plays a key role in linking disparate systems by unifying and orchestrating data across the product lifecycle [14,19]. The capability to integrate data models from disparate sources is necessary for understanding, analyzing, and controlling a product's performance [20]. The concept of the digital thread is garnering more attention since some argue it is the backbone of digital twin applications [21].

Recently, researchers have focused heavily on linking different lifecycle data in a digital thread. Helu et al. [22] presented a reference architecture designed to enable the fusion of manufacturing and other product lifecycle data in the digital thread. Helu et al. [23] presented a way to link as-planned and as-executed product data in the standards-based digital thread. To support lifecycle decision-making, Bernstein et al. [24] developed a prototype system correlating as-designed CAD, as-planned toolpath, and as-executed toolpath by visually overlaying the toolpath onto CAD geometry. Monnier et al. [25] proposed a mapping method focusing on relating controller data and numerical control (NC) code in standard data representations to facilitate the flow of information between as-planned and as-executed data and to identify discrepancies during the execution process. Trainer et al. [12] presented a STEP-QIF PMI mapping table to bridge the gap between CAD and CMM communities for future interoperability tools.

There have been several studies on system- or process-level CAD/computer-aided manufacturing (CAM)/computer-aided inspection (CAI) integration. Cho and Seo [26] presented an improved inspection planning strategy for sculptured surfaces in an on-machine measurement system by integrating CAD/CAM data into the CAI process. Another attempt to integrating CAD, CAM, and CAI was made by extracting feature information from STEP files and using the information as reference data for better quality control [27]. However, only a handful of studies have reported on fusing different standard formats of as-designed and as-inspected data in the context of a standards-based digital thread.

Commercial solutions that organize and integrate data generated across the product lifecycle exist. However, these systems are typically expensive and tend to not target a diverse group of users, especially small-to-medium sized enterprises. Open reference architectures are needed to support standards-based digital thread to exchange information between design, fabrication, and inspection that can be freely used by all participants [1]. To support this vision, the National Institute of Standards and Technology (NIST) Smart Manufacturing Systems (SMS) Test Bed[1] provides an example cyber-physical infrastructure to collect, curate, and access manufacturing data. One key feature of the SMS Test Bed is that its architecture supports multiple domain standards. Thus, we leverage the NIST SMS Test Bed [24]. Fig. 1 shows the data types collected by the NIST SMS Test Bed and the use of the existing data standards as a basis for their representation.

Note that the SMS Test Bed itself does not present specific guidelines for mapping or linking across these different formats, rather the SMS Test Bed provides reference data for testing mapping techniques and methods, some of which are described above. Next, we conclude Section 2 with a discussion on how the use of ontology with KGs can aid

[1] Can be accessed via https://smstestbed.nist.gov.

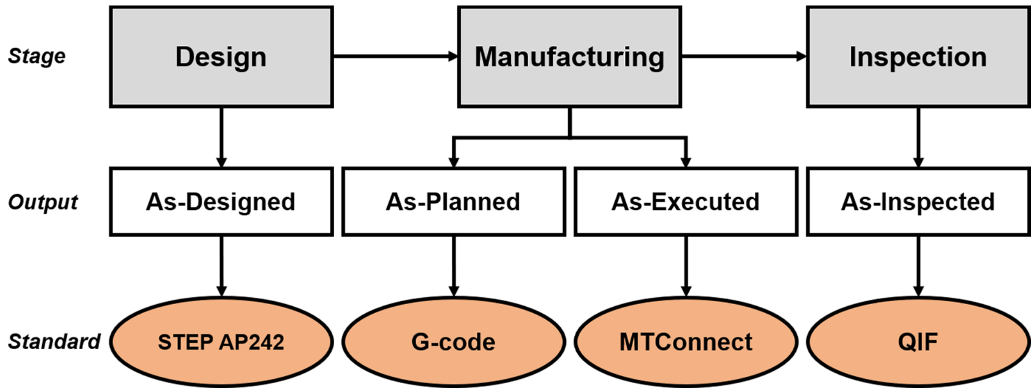

Fig. 1. Types of data collected from the NIST SMS Test Bed.

in engineering data integration.

2.3. Data integration based on ontology with knowledge graphs

Defining a structured process for integrating disparate data formats requires high-level abstractions of their underlying entities. If done properly, interoperability of the underlying data representations can theoretically be realized.

To achieve such data integration, ontologies with KGs have been implemented due to their richness in semantic definitions to extract and infer knowledge in a human readable way [28]. Similar to many other data-enriched disciplines, domains related to product and manufacturing systems, e.g. systems engineering, product design, and production, invest in the use and development of ontologies and KGs for product data integration.

For example, Sudarsan et al. [29] recommended the use of ontologies to extend a proposed product lifecycle management model. Matsokis and Kiritsis [30] translated the unified modeling language (UML) based semantic object model (SOM) into an ontology, demonstrating its use through a case study in automotive industry. Panetto et al. [31] developed OntoPDM aiming to use STEP and International Electrotechnical Commission (IEC) 62264 standards in order to define a product ontology that would allow interoperability. Lu et al. [32] built an ontology for variational geometric constraint specifications to enrich the semantic in order to allow product data to be exchanged during the product development.

To leverage the use of ontologies, Lee et al. [33] proposed to create 4 layers of ontologies, each stage having its own lower level ontology which will respect the topmost ontologies. The goal of using an ontology hierarchy with KGs is to further strengthen data integration by keeping a concordance between the information shared across multiple systems. Barbau et al. [9] developed a tool OntoSTEP that translates STEP schema and instances into an ontology and a KG respectively in an easy and automated way. Krima et al. [10] provides full specification of OntoSTEP. Sarigecili et al. [34] defined EXPRESS data models for tolerance analysis and then translated them into ontologies using OntoSTEP to infer new knowledge from the existing KGs in a more understandable way. Pursuing the integration of data using ontology with KGs is still a work in progress that has a lot of opportunities. However, efforts have been focused in the early product lifecycle stages which is a barrier when trying to realize the digital thread.

3. Design and inspection data fusion in standards-based digital thread based on knowledge graphs

To maximize the usability of a standards-based digital thread in the context of smart manufacturing, linking heterogeneous data from different lifecycle stages is critical to maintain associativity and traceability [35]. Moreover, realizing model-based inspection (MBI) requires an integrated data model to capture all information generated through design, manufacturing, and inspection [36]. Thus, it is necessary to integrate different lifecycle data representations and link them to enable model-based design, manufacturing, and inspection in the existing standards-based digital thread.

In this work, we use ontology with KGs to integrate lifecycle data representations. Many studies have attempted to convert existing information models and data (such as those that are eXtensible Markup Language (XML)-based and EXPRESS-based) into ontologies and KGs since ontology models provide additional affordances, including the ability to share common knowledge among humans and/or machines, provide formal analysis, and reuse domain knowledge [30]. Fig. 2 depicts the proposed process to build an integrated knowledge base by translating lifecycle data in the standards-based digital thread concept. First, each data format is separately translated into a KG. Then, each resulting KG is linked together to make a fully-connected integrated knowledge base.

In this study, we fuse as-designed data in the form of STEP AP242 and as-inspected data represented by QIF. Note that integrating manufacturing data is out of scope for this paper. Fig. 3 roughly shows the intersection between information models of STEP AP242 and QIF. As two standards intersect in a model-based definition, it is possible to enrich each data model by linking the overlapping information and providing exclusive information in each format to one another, such as product data information in STEP and inspection results in QIF. Note that each standard has different perspectives in its feature definitions. As our primary focus is to fuse as-designed and as-inspected data, the following discussion will outline methods that we adopted to translate STEP and QIF into KGs.

The main contribution of this study is to propose a framework that integrates and links heterogeneous standard data formats by using ontology with KGs in the context of standards-based digital thread. The tools adopted in this study, such as translators, can be replaced by other tools that serve the same purpose. One of the key advantages of the proposed framework is that it uses semantic web technologies for representing different lifecycle data in a homogeneous manner, which also enables users to query existing data or infer new knowledge.

3.1. Translating STEP into a knowledge graph

Data models and schemas defined in the STEP framework (as ISO 10303) are represented in the EXPRESS language. STEP Part 11 defines the specification of EXPRESS. The basic constituents of EXPRESS are entities, which define concepts. Entities contain attributes which are either other entities or certain types of values. STEP instances that conform to a specific EXPRESS schema are represented in a specific language whose syntax is defined in STEP Part 21 (P21). EXPRESS and P21 have not been widely adopted for information modeling except for STEP and the Industry Foundation Classes (IFC) standard (ISO 16739)

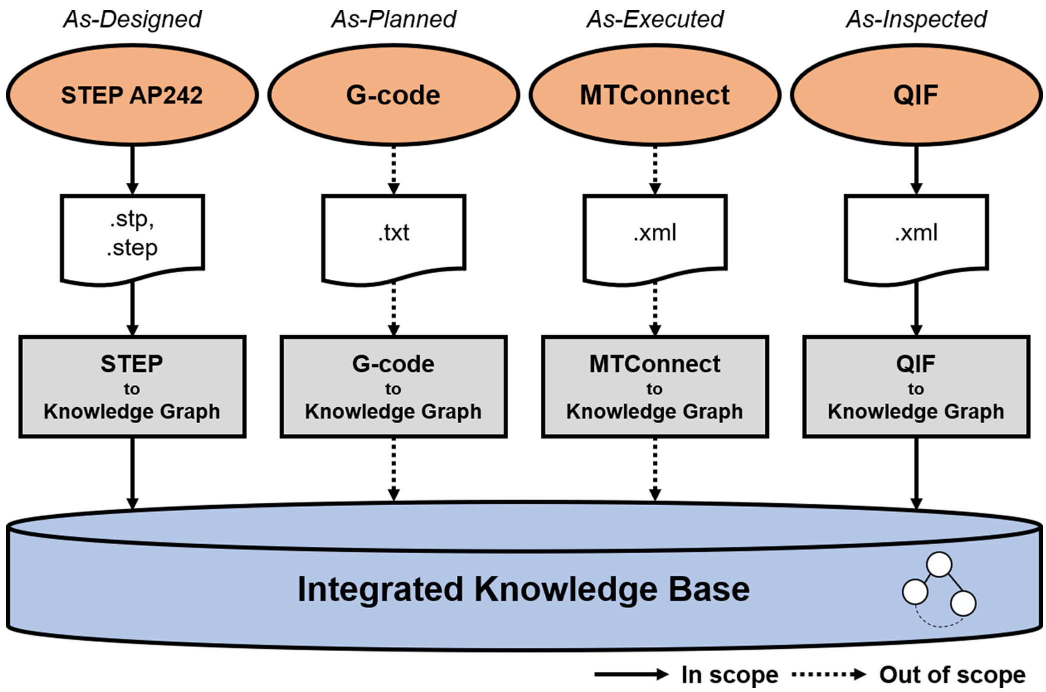

Fig. 2. Building an integrated knowledge base by translating lifecycle data in the digital thread into knowledge graphs.

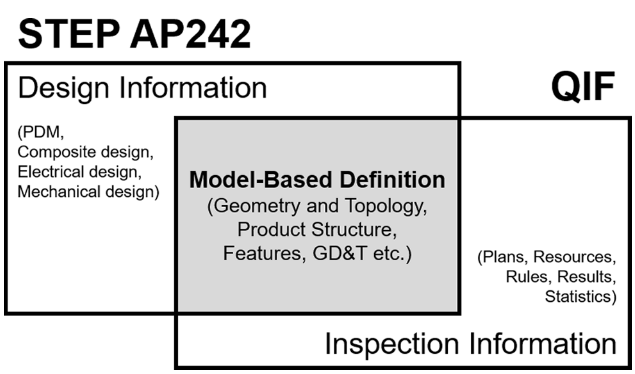

Fig. 3. Simplified view of the intersection between STEP AP242 and QIF.

used in the building industry. Therefore, it is challenging to integrate STEP data with other types of data.

Despite the rich expressiveness of EXPRESS compared to other modeling languages, researchers [9,37] have pointed out that EXPRESS lacks the ability to express formal semantics. They insisted that converting EXPRESS into a computational, logic-based language, such as the Web Ontology Language (OWL) and the Resource Description Framework (RDF), will bring advantages in complex knowledge representation and the reuse of predefined knowledge definitions. To this end, there have been several studies on EXPRESS to OWL conversion to maintain the richness of the data models and improve their interoperability.

Here, we leverage OntoSTEP [9] to translate as-designed data in STEP AP242 into ontology with KGs using OWL. OntoSTEP was implemented as a plug-in of Protégé and supports automated conversion from EXPRESS and P21 to OWL. Note that some researchers developed another EXPRESS to OWL translator [38] and IFC to RDF translator [39] which is freely available on the web. That being said, OntoSTEP[2] is the only free translator that supports the conversion of STEP instance files written in P21 into KGs written in OWL or RDF.

[2] OntoSTEP is available in Protégé 4.1.

Fig. 4 provides an example of STEP to OWL translation using OntoSTEP. An entity *product* is defined in EXPRESS and is instantiated in P21, as shown in Fig. 4(a) and (c), respectively. Fig. 4(b) shows a part of the translation of the entity *product* into OWL. Each attribute is translated into a class and a corresponding object property is created to connect the attribute with the parent class. Fig. 4(d) shows the translation of the P21 instance provided in Fig. 4(c) into an OWL-expressed ontology and knowledge entities. There are four individuals asserted, one class (*i*1) and three data properties (*i*1*_id*, *i*1*_name*, *i*1*_desc*), and three object properties are asserted to connect each data property individual with the class individual. Fig. 4(e) shows an ontology with knowledge entities represented as a KG. OntoSTEP defines only four data property types: *to_boolean*, *to_decimal*, *to_integer*, *to_string*. In this example, all attributes represented in EXPRESS (*id*, *name*, *description*) are inherently of string type. As a result, all data property individuals are asserted as *to_string* in OWL.

3.2. Translating QIF into a knowledge graph

QIF provides a unified XML framework that contains the information models expressed as XML SChema Definitions (XSDs). DMSC chose XML for encoding QIF because it is publicly open, free, and being widely used in many standards. Moreover, there are tools available for manipulating and incorporating XML into software [40]. XML is suited for storing and sharing structured data between different processes. However, RDF and OWL add a semantically rich layer that provides a way to incorporate reasoning and logic-based methods. Therefore, the translation of XML documents into OWL-based or RDF-based KGs will help formalize rich semantics and allow querying, linking, and enriching existing knowledge acquired from their original primary sources [41].

The translation of XML representations to OWL can be classified into two approaches [41]: *validation* and *instance* approaches. Validation approaches (e.g., [42–44]) use XML schema and XML documents conforming to that schema, whereas the instance approaches (e.g., [45–47]) use only XML documents without any schema. The validation approaches generate more structured and semantically rich ontologies since schemas usually include more constructs than instances. This,

(a)

```
ENTITY product;
       id : identifier;
       name : label;
       description : OPTIONAL text;
END_ENTITY;
```

(b)

```
SubClassOf(:product ObjectAllValuesFrom(:product_has_id :identifier))
SubClassOf(:product ObjectAllValuesFrom(:product_has_name :label))
SubClassOf(:product ObjectAllValuesFrom(:product_has_description ObjectUnionOf(:text owl:Nothing)))
SubClassOf(:product ObjectMaxCardinality(1 :product_has_id))
SubClassOf(:product ObjectMaxCardinality(1 :product_has_name))
SubClassOf(:product ObjectMaxCardinality(1 :product_has_description))
```

(c)

```
#1=PRODUCT('id_1','product_1','This is a product.');
```

(d)

```
ClassAssertion(:product <http://nist.gov/ontology#i1>)
ObjectPropertyAssertion(:product_has_id <http://nist.gov/ontology#i1> <http://nist.gov/ontology#i1_id>)
ObjectPropertyAssertion(:product_has_name <http://nist.gov/ontology#i1> <http://nist.gov/ontology#i1_name>)
ObjectPropertyAssertion(:product_has_description <http://nist.gov/ontology#i1> <http://nist.gov/ontology#i1_desc>)
DataPropertyAssertion(:to_string <http://nist.gov/ontology#i1_id> "id_1"^^xsd:string)
DataPropertyAssertion(:to_string <http://nist.gov/ontology#i1_name> "product_1"^^xsd:string)
DataPropertyAssertion(:to_string <http://nist.gov/ontology#i1_desc> "This is a product."^^xsd:string)
```

(e)

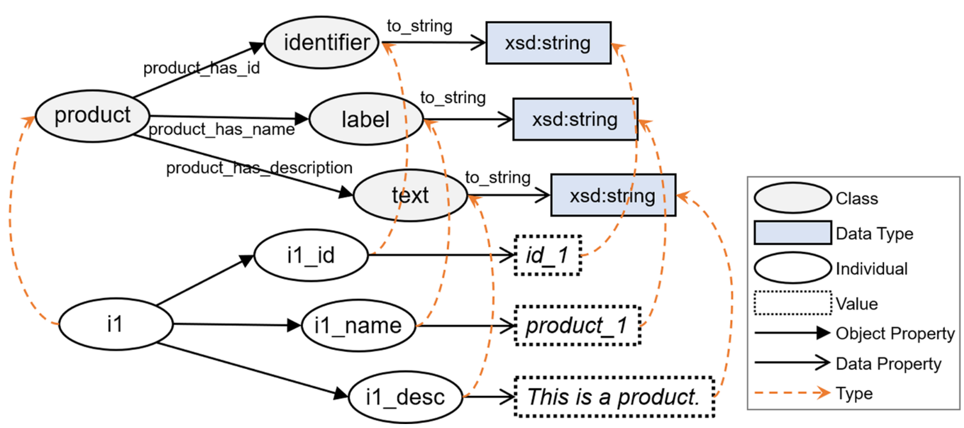

Fig. 4. Example of a STEP entity and its translation to OWL.

however, obfuscates the implementation of the translation. On the other hand, the instance approaches only use elements and attributes in XML instances to generate a KG including classes, object properties, and data properties without advanced operators and restrictions. Therefore, the resulting KG is not the best suited for reasoning purposes despite implementation being simple [41].

Despite all the efforts to transform XML into OWL, we have not found any *free and open* solution for translating the QIF schema as well as QIF instances into OWL ontology with KGs. Moreover, our primary goal is to link and enrich instance files of various types acquired from the NIST SMS Test Bed. Therefore, we use an instance-based translator, XML Tab [47], as a means of translating QIF instances into OWL-defined KGs. XML Tab[3] is a plug-in for Protégé that facilitates the import of XML documents by generating classes, properties, and instances in a knowledge base.

Fig. 5 shows an example of an XML instance to OWL translation using XML Tab. Unlike STEP instances, XML instances explicitly specify element and attribute names (as shown in Fig. 5(a)), affording ontology class generation. XML Tab transforms elements without values except the root (e.g., “product” in Fig. 5(a)) into classes, and the elements with values (e.g., “id”, “name”, and “description” in Fig. 5(a)) into data properties as shown in Fig. 5(b). Fig. 5(c) shows an ontology with knowledge entities represented as a KG. The complexity of the KG is much simpler than that of STEP KG in Fig. 4(e) because the result ontology does not have object properties. Note that all values are of string type since it is impossible to know the data types solely from XML documents without a schema definition. Therefore, some numerical data types should be manually edited after the translation as needed.

3.3. Fusing design and inspection knowledge graphs

MBI necessitates the integration of design through manufacturing to inspection. To achieve such integration, a unified information model is required to encapsulate all information generated through inspection processes into MBD models [36]. DMSC has developed QIF as a unified framework supporting the digital thread. QIF provides information models which enable the effective exchange of metrology data and improve the ability of quality assurance throughout the product lifecycle [48].

The initial version of QIF focused on only encoding measurement *results*. Through its newest releases, the scope has been extended to fully support model-based design and inspection, partly through the concepts of *rules* and *plans*. STEP AP242 is already widely adopted for MBD in commercial CAD systems. Moreover, the QIF architects accept the idea of utilizing different formats for MBD in their framework. Therefore, we propose a method to merge two heterogeneous formats, STEP and QIF, into an integrated knowledge base to enrich standards-based digital thread that will, in turn, support model-based integrated inspection.

Fig. 6 shows our concept of fusing STEP and QIF KGs as an integrated knowledge base. Since the two ontologies and KGs are

[3] XML Tab is available in Protégé 3.3.1.

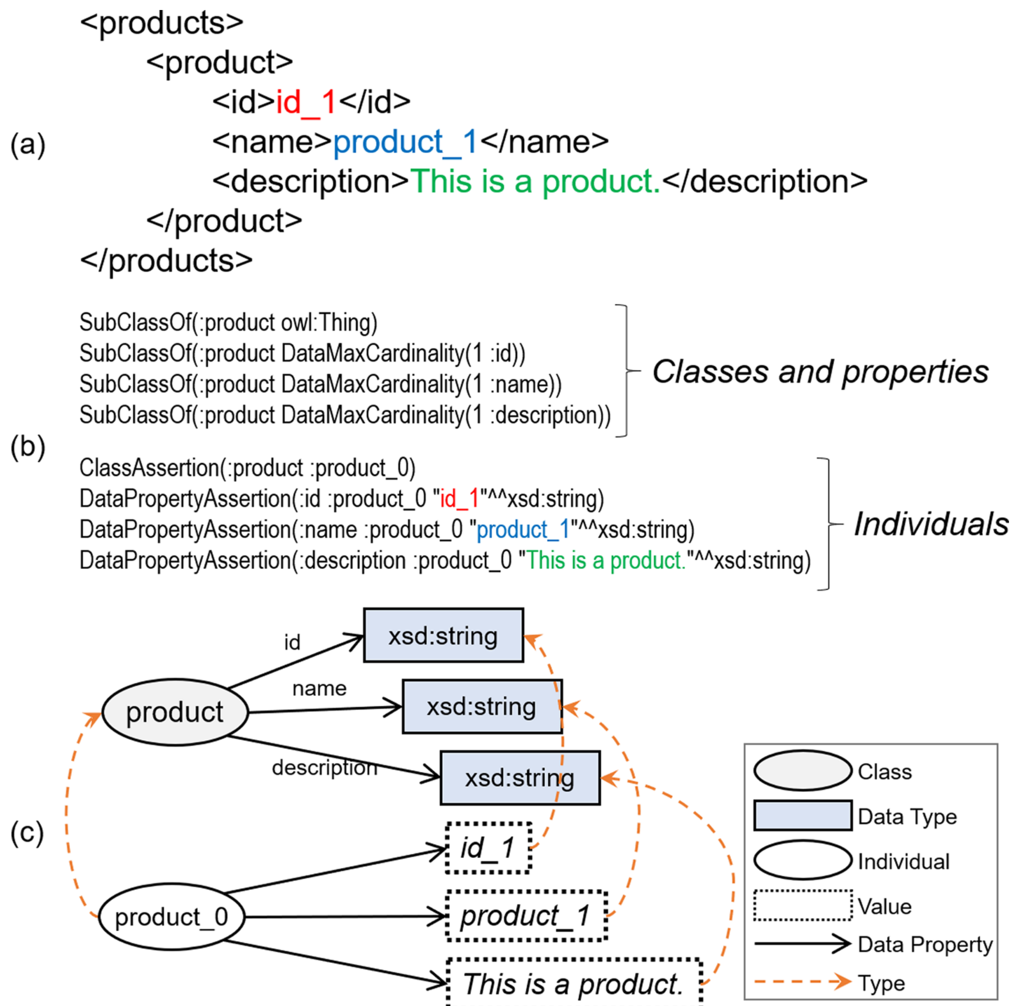

Fig. 5. Example of a XML instance and its translation to OWL.

generated separately and have different structures, a mapping specification is required to link them together. Hence, we leverage the PMI mapping specification between STEP and QIF [12]. Table 1 represents an abridged version of the mapping specification, focusing on geometric tolerances. In regards to GD&T, the data model of QIF is more detailed than that of STEP. Smeantic Web Rule Language (SWRL) rules need to be devised to connect different classes with the same semantics based on the mapping relation. After linking them, users can leverage the enhanced integrated knowledge base by querying information using Semantic Query-Enhanced Web Rule Language (SQWRL) queries.

Fig. 7 shows the data models in STEP and QIF for flatness tolerance. In the STEP data model, every entity is directly related to each other whereas elements in the QIF data model are indirectly referenced through id values. Based on the analysis of the dataset from the NIST SMS Test Bed, we found that the only way to link these two data models is to compare their *names*. Ideally, if a persistent identifier[4] was assigned to each GD&T item and used throughout the product lifecycle, then persistent identifiers could be compared instead of names. We analyzed the QIF instances in the technical data package (TDP) from the NIST SMS Test Bed and found that among four aspects (i.e., *definition*, *nominal*, *item*, and *actual*) only "item" (*FlatnessCharacteristicItem* in this case) has a *name* element instantiated. That means that the attribute *name* of the entity *geometric_tolerance* in the STEP data model needs to be compared and linked to the *name* element of *FlatnessCharacteristicItem* in the QIF model. In this example, data enrichment can then be accomplished from the viewpoint of the STEP data model by allowing access to *FlatnessCharacteristicActual*, which contains inspection results.

The integrated knowledge base can be used for various purposes. The data consistency can be checked by comparing GD&T definitions in STEP with QIF. Here, definitions in STEP are considered as the "ground truth". In the design stage, designers can refer to inspection plans generated by inspectors if we assume that any new information in the form of KG will be pushed to the knowledge base instantaneously. For example, dimensions read from a coordinate-measuring machine can be taken into account when determining the dimension of a part, or vice versa. Above all, anyone who is allowed to gain access to the knowledge base can leverage the associated knowledge accumulated over time.

4. Case study

To demonstrate the utility of the integrated knowledge base, we leveraged a dataset describing a three-component box assembly. Fig. 8 shows the MBDs and manufactured parts of three components of the assembly: cover, plate, and box. This dataset is available as a TDP as part of the NIST SMS Test Bed open to the public [49]. We acquired STEP AP242 files by using SolidWorks 2019 MBD functions since the dataset only provides the product definition in SolidWorks 2016. For as-

[4] QIF provides a mechanism to assign persistent identifiers to objects, whereas in STEP, there is no recommended practice to assign persistent identifiers to entities yet. However, there is an ongoing discussion on the CAx implementor forum (https://www.cax-if.org/) on this matter.

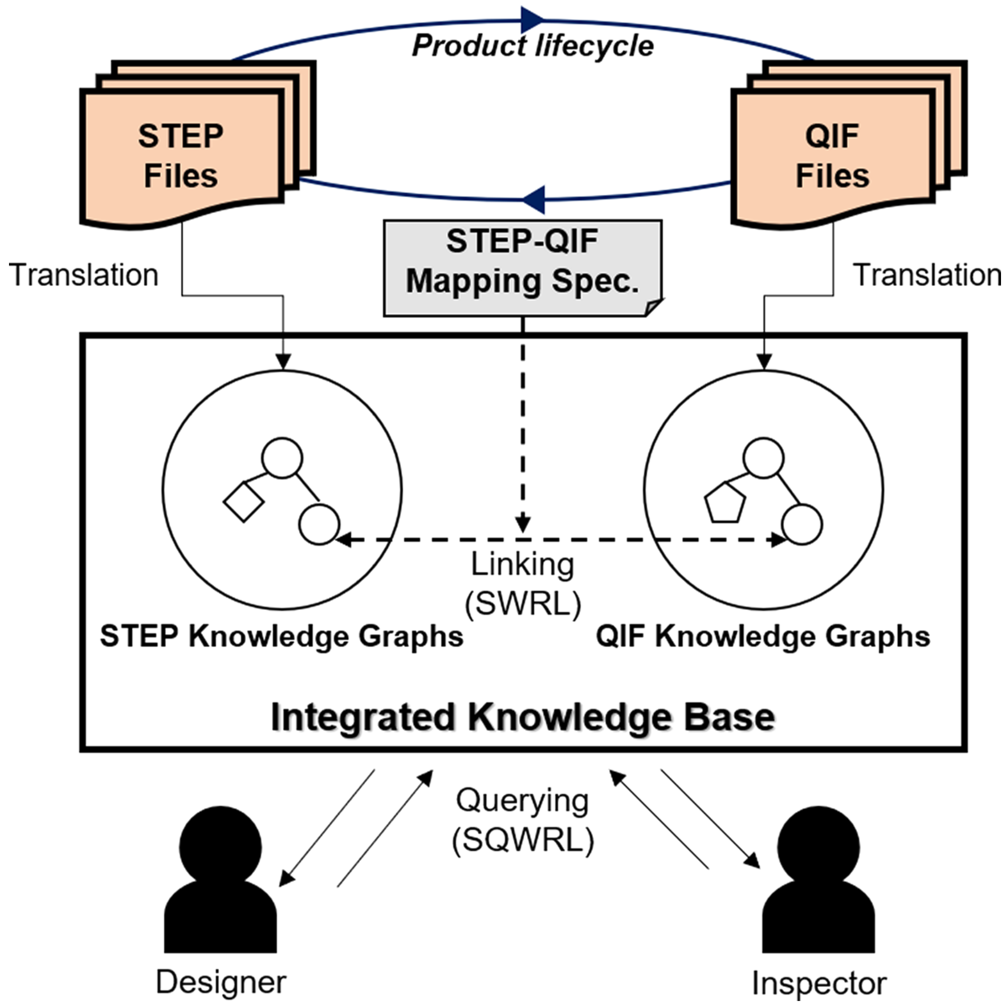

Fig. 6. Fusing STEP and QIF in the integrated knowledge base.

Table 1
STEP-QIF mapping specification for geometric tolerances.

PMI Type	STEP AP242	QIF
Tolerance	geometric_tolerance	CharacteristicDefinitionBaseType
Angularity	angularity_tolerance	AngularityCharacteristicDefinitionType
Circular runout	circular_runout_tolerance	CircularRunoutCharacteristicDefinitionType
Circularity/roundness	roundness_tolerance	CircularityCharacteristicDefinitionType
Coaxiality	coaxiality_tolerance	ConcentricityCharacteristicDefinitionType
Concentricity	concentricity_tolerance	ConcentricityCharacteristicDefinitionType
Cylindricity	cylindricity_tolerance	CylindricityCharacteristicDefinitionType
Flatness	flatness_tolerance	FlatnessCharacteristicDefinitionType
Parallelism	parallelism_tolerance	ParallelismCharacteristicDefinitionType
Perpendicularity	perpendicularity_tolerance	PerpendicularityCharacteristicDefinitionType
Position	position_tolerance	PositionCharacteristicDefinitionType
Profile of a line	line_profile_tolerance	LineProfileCharacteristicDefinitionType
Profile of a point	n/a	PointProfileCharacteristicDefinitionType
Profile of a surface	surface_profile_tolerance	SurfaceProfileCharacteristicDefinitionType
Straightness	straightness_tolerance	StraightnessCharacteristicDefinitionType
Symmetry	symmetry_tolerance	SymmetryCharacteristicDefinitionType
Texture	n/a	SurfaceTextureCharacteristicDefinitionType
Thread	n/a	ThreadCharacteristicDefinitionType
Total runout	total_runout_tolerance	TotalRunoutCharacteristicDefinitionType

inspected data, QIF version 2.1[5] was used to retrieve incoming inspection reporting data. Each data instance contains 20 inspection results of the same part.

Each component has one STEP AP242 file (as-designed) and one QIF file (as-inspected). As mentioned earlier, two files must contain overlapping MBD information to accomplish the mapping (or merging). We checked that as-designed files contain 3D shape and GD&T information and the as-inspected files contain features and GD&T information. Thus, we were able to use the mapping in Table 1 to link as-designed and as-

[5] The latest version of QIF is 3.0.

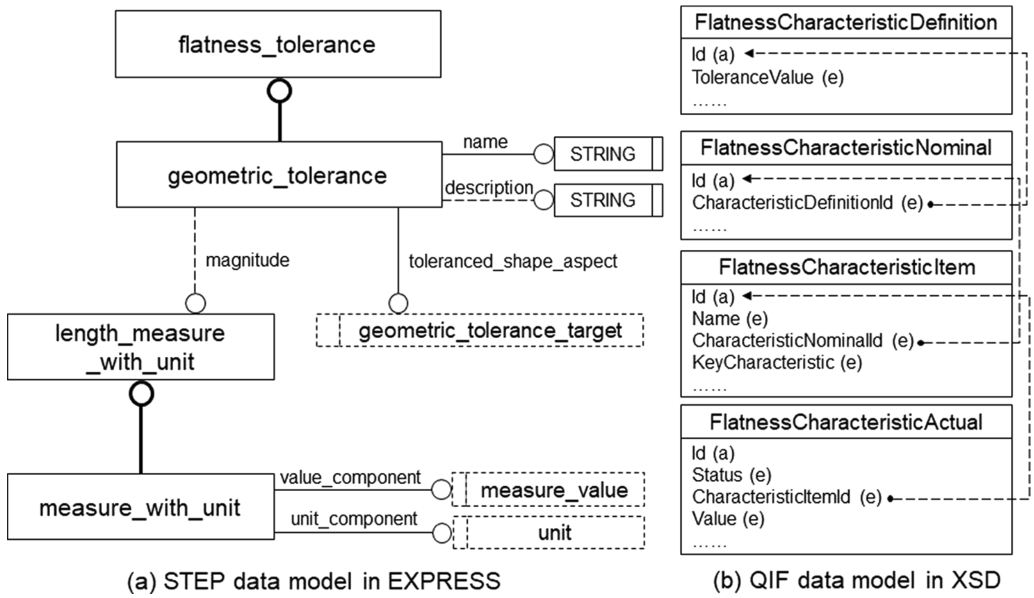

Fig. 7. Flatness tolerance defined in STEP and QIF data models.

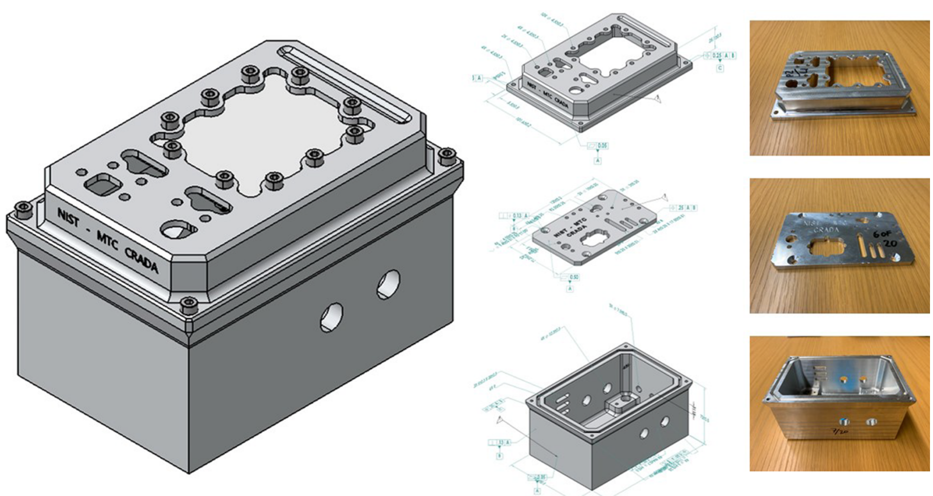

Fig. 8. Three-component box assembly.

inspected data since GD&T information overlapped. We published a new dataset[6], including STEP AP242 files, QIF files, STEP KGs in RDF, and QIF KGs in RDF used in this study.

We used Protégé 5.5.0 to merge STEP and QIF ontologies with KGs and manipulate the integrated knowledge base. Fig. 9 shows that the two ontologies are in a single hierarchy by having *owl*: *Thing* as a common superclass. We used Pellet and Drools, which are already deployed in Protégé, for reasoning and querying purpose, respectively. We used SWRL to define rules and SQWRL to define queries.

To construct general rules and queries, it is desirable for the ontology of the knowledge base to have a meaningful class hierarchy. Therefore, we made temporary hierarchies for GD&T classes of both STEP and QIF KGs. We referred to each schema definition to build each hierarchy. For example, a characteristic type in QIF is defined by four aspects: *definition*, *nominal*, *item*, and *actual*. Thus, *CharacteristicDefinition*, *CharacteristicNominal*, *CharacteristicItem*, *CharacteristicActual* were added for grouping all types of characteristics in the QIF KG.

Listing 1 shows a SWRL rule to link geometric tolerances defined in STEP and QIF, respectively. The prefixes *step*: and *qif*: indicate to which representations each data item belongs. The rule matches names first, finds *CharacteristicDefinition* by comparing IDs, and finally links individual data in *geometric_tolerance* and *CharacteristicDefinition* by the *owl*: *sameAs* function, based on a mapping specification [12]. After applying this rule, users can see an integrated individual in a single view in Protégé as shown in Fig. 10(a). Fig. 10(b) represents the integrated view as a connected KG. We used GraphDB of Ontotext[7] to generate visual KGs. As a result, it becomes possible to gain access to inspection results (*CharacteristicActual*) in the QIF KG starting from the STEP KG.

The integrated knowledge base can also be used to check data consistency. It is safe to assume that any two corresponding individuals in STEP and QIF must have the same tolerance value. The SWRL rule presented in Listing 2 is used to store STEP individual whose tolerance value is not equivalent to that of the corresponding QIF individual into *qif*: *AbnormalCharacteristic*. Note that we included

[6] The dataset will be made available online.

[7] https://www.ontotext.com/products/graphdb/.

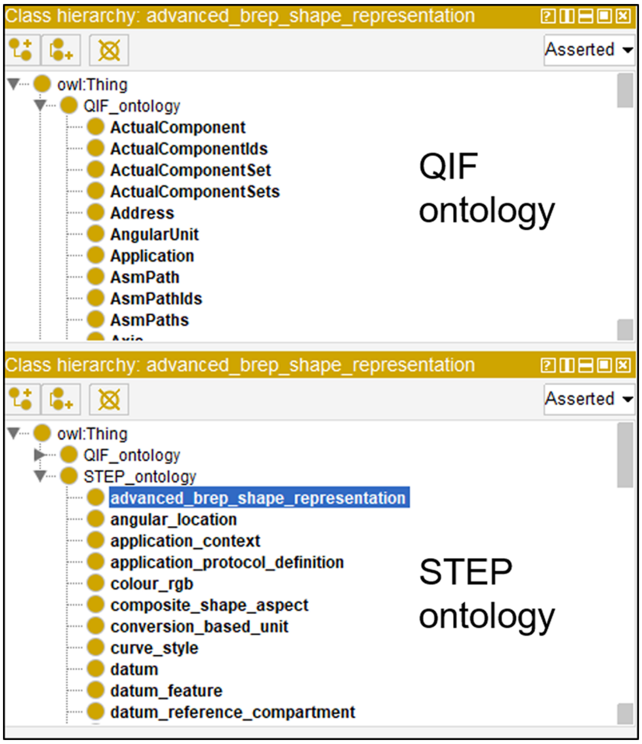

Fig. 9. Integrated STEP and QIF ontologies in Protégé.

```
step:geometric_tolerance(?sTol)
^ step:geometric_tolerance_has_name(?sTol,?sName)
^ step:to_string(?sName,?sNameVal)
^ qif:CharacteristicItem(?qItm)
^ qif:Name(?qItm,?qNameVal)
^ swrlb:stringEqualIgnoreCase(?sNameVal,?qNameVal)
^ qif:CharacteristicNominalId(?qItm,?qNomId1)
^ qif:CharacteristicNominal(?qNom)
^ qif:_id(?qNom,?qNomId2)
^ swrlb:equal(?qNomId1,?qNomId2)
^ qif:CharacteristicDefinitionId(?qNom,?qDefId1)
^ qif:CharacteristicDefinition(?qDef)
^ qif:_id(?qDef,?qDefId2)
^ swrlb:equal(?qDefId1,?qDefId2)
-> owl:sameAs(?sTol,?qDef)
```

Listing 1. SWRL rule to map geometric tolerances.

qif: *AbnormalCharacteristic* as a hypothetical class to check "abnormal" GD&T individuals through the rule shown in Listing 2. This will ensure a more straightforward mechanism for users to check discrepancies by just checking inferred individuals of the class. Note that the QIF data property *qif*: *ToleranceValue* can take the STEP individual *?sTol* as input because QIF data properties now also belong to STEP individuals thanks to the rule in Listing 1, which makes the query expression concise. Meanwhile, in the QIF representation, some characteristic types store the nominal value in *ToleranceValue* of *CharacteristicDefinition* whereas some store as a *TargetValue* of the *CharacteristicNominal*. Thus, another query must be devised since the current query conforms to the former.

From a design perspective, the integrated knowledge base is beneficial for designers to check if all the GD&T defined in the as-designed data were actually taken into account in the subsequent inspection processes. Listing 3 shows a query to identify GD&T items coexisting in STEP and QIF by comparing their names. *draughting_callout* individuals in STEP KG are used to store all the GD&T items, including datums and annotations. By applying this query, we were able to find a discrepancy in the GD&T items between STEP and QIF. We also found that more than two items were missing in the QIF KG in the case of plate component.

While linking and comparing STEP and QIF KGs is our main area of interest, we can simply check consistency either of the STEP or the QIF KG. Listing 4 presents a query that finds QIF individuals with negative nominal values, which would not make physical sense. Through this query, we discovered an individual of negative perpendicularity from the QIF KG for both the cover and plate components of the assembly, as shown in Fig. 11. QIF also provides a way to validate the quality of information for the QIF documents through eXtensible Stylesheet Language Transformations (XSLTs) language checks. Note that the quality checking mechanism provided by QIF does not resolve negative values itself.

As QIF stores inspection results, looking into the failed inspection results will facilitate a better understanding of the results. The query presented in Listing 5 is precisely for this purpose, and will return *CharacteristicActual* individuals with the status of *FAIL*, together with its nominal definition (*CharacteristicNominal*). Fig. 12 shows the result of the query. For example, if a hole failed in inspection (information), then we can infer by rules and queries that no pin will fit into the hole (knowledge), and finally we can determine to remove more material (wisdom). The beauty of using an OWL ontology with KGs is that it can be used to incorporate not only information but also taxonomy, rules/queries, knowledge, and wisdom in a homogeneous manner.

5. Discussion

In the case study, we introduced rules and queries that utilize the integrated knowledge base. The first rule in Listing 1 mapped STEP and QIF based on the mapping specification. Here, we presented an upper level and general rule for simplicity. However, linking each type of GD&T items separately might be required in order to best leverage the mapping relation. In this regard, the STEP data model may need to be improved to be fully compatible with the QIF data model from the GD&T point of view.

A rule in Listing 2 and a query in Listing 3 aimed to find out discrepancies between STEP and QIF, and actual discrepancies were found in the TDP. These are particularly important because they coincide with the core goal of this study, i.e., to obtain useful information from the integrated knowledge base. Two queries in Listings 4 and 5 purported to check data inconsistency and inspection results in QIF. One might claim that these types of checking could be done without the proposed framework. However, that would require additional efforts to develop a validation mechanism or software whereas a logic-based language, i.e., OWL, can easily accomplish such tasks. In our case study, each rule and query took less than a second to perform. It might take longer for complex rules and queries in a larger knowledge base.

Currently, each data translation (STEP to OWL, QIF to OWL) is a fully automated process. On the other hand, merging the two KGs was done manually in this study, and it was not particularly challenging to do so. Moreover, some data types in QIF KGs also needed to be manually modified because of the loss of semantics due to the nature of the instance-based XML to OWL converter, not because of the difficulty in automation. Therefore, we are confident that the whole pipeline can be easily automated.

The quality and accuracy of the generated KGs are heavily dependent on the capability of individual translators. Due to the intrinsic differences between languages, it is challenging to keep the original content intact after translation. Therefore, the main purpose of translators is usually set to minimize the information loss. Regarding OntoSTEP, it still has room for improvement in terms of converting rules and functions defined in EXPRESS. Regarding XML to OWL translators, validation approaches have advantages in maintaining the semantics over instance approaches. To compare the quality of generated OWL files, the statistics including the number of classes, properties, and axioms are often used. In general, the larger the number of entities, the better the quality of the translator.

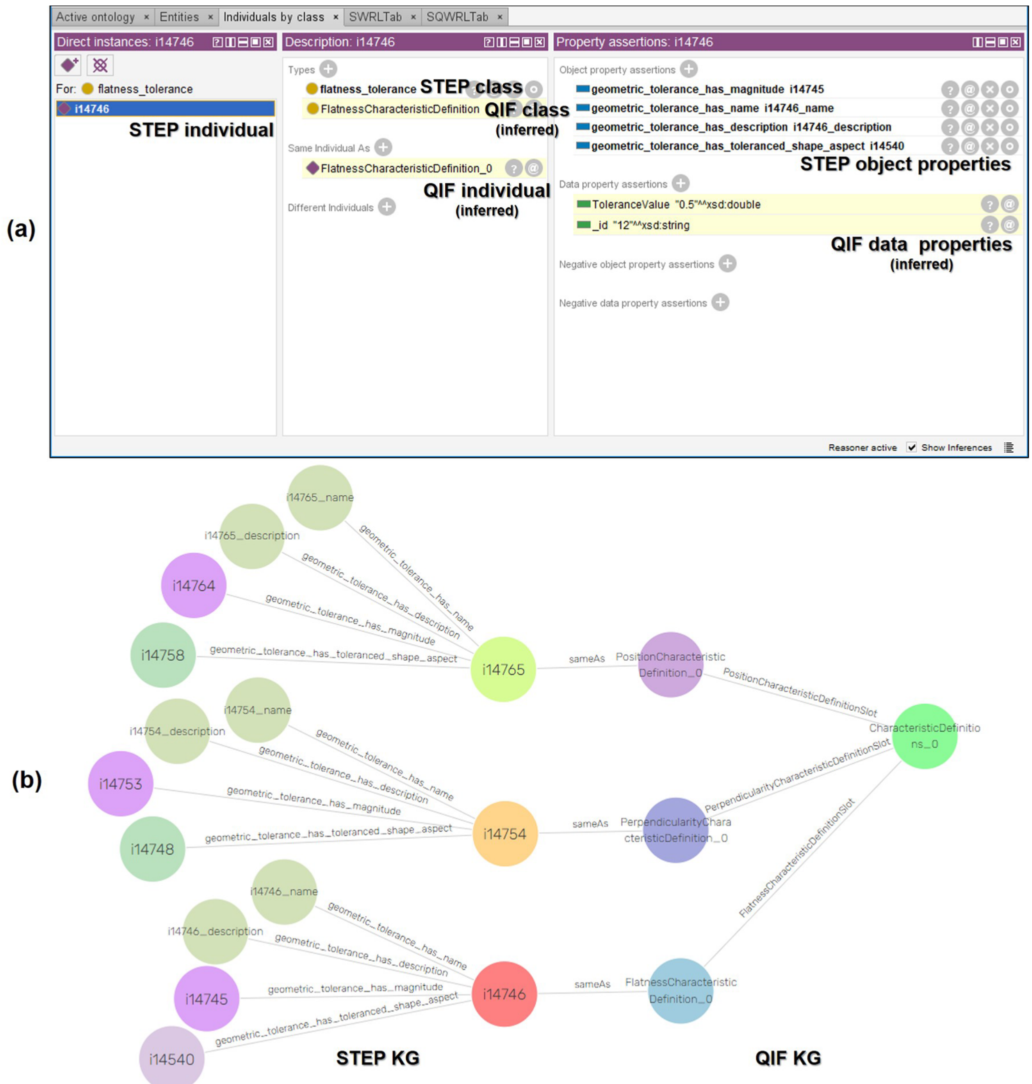

Fig. 10. Integrated STEP and QIF view (a) in Protégé and (b) as a knowledge graph (partial).

```
step:geometric_tolerance(?sTol)
^ step:geometric_tolerance_has_magnitude(?sTol,?sMag)
^ step:measure_with_unit_has_value_component(?sMag,?sCom)
^ step:to_decimal(?sCom,?sVal)
^ qif:ToleranceValue(?sTol,?qVal)
^ swrlb:notEqual(?sVal,?qVal)
-> qif:AbnormalCharacteristic(?sTol)
```

Listing 2. SWRL rule to find discrepancy in tolerance values.

```
step:draughting_callout(?sCallout)
^ step:representation_item_has_name(?sCallout,?sName)
^ step:to_string(?sName,?sNameVal)
^ qif:CharacteristicItem(?qItm)
^ qif:Name(?qItm,?qNameVal)
^ swrlb:stringEqualIgnoreCase(?sNameVal,?qNameVal)
-> sqwrl:select(?sCallout)
```

Listing 3. SQWRL query to list GD&T items in STEP and QIF.

```
qif:CharacteristicDefinition(?qDef)
^ qif:ToleranceValue(?qDef,?qVal)
^ swrlb:lessThan(?qVal,0.0)
-> sqwrl:select(?qDef,?qVal)
```

Listing 4. SQWRL query to list tolerances with negative values.

The main components that consist of the proposed framework such as standards, translators, and mapping specifications have been well established. Most of all, they are all open and free of charge, which dramatically improve the feasibility of wide implementation. Moreover, KGs are being widely adopted in graph databases such as Neo4j[8] and GraphDB. They can directly import OWL and RDF files and have become a key enabler to effectively integrate different lifecycle data while providing the improved capability of semantic web technologies. In the near future, we anticipate that the proposed framework will gain more attention together with graph databases.

6. Conclusion and future work

In this paper, we discussed fusing different data from across the product lifecycle to enrich the standards-based digital thread ultimately for making better decisions. Specifically, we focused on merging as-designed data in STEP AP242 and as-inspected data in QIF using ontology with KGs. Other researchers have tried to transform existing data models into ontologies and KGs since these representations exhibit advantages in building, analyzing, and sharing semantically rich knowledge in a way that both humans and machines can interpret.

[8] https://neo4j.com/.

qDef	qVal
qif:PerpendicularityCharacteristicDefinition_0	"-0.13"^^xsd:double

Fig. 11. Result of a query for negative tolerance values.

```
qif:CharacteristicActual(?qAct)
^ qif:StatusSlot(?qAct, ?qStat)
^ qif:Status(?qStat)
^ qif:CharacteristicStatusEnum(?qStat, ?qEnum)
^ swrlb:equal(?qEnum, "FAIL")
^ qif:CharacteristicItemId(?qAct, ?qItemId1)
^ qif:CharacteristicItem(?qItem)
^ qif:_id(?qItem, ?qItemId2)
^ swrlb:equal(?qItemId1, ?qItemId2)
^ qif:CharacteristicNominalId(?qItem, ?qNomId1)
^ qif:CharacteristicNominal(?qNom)
^ qif:_id(?qNom, ?qNomId2)
^ swrlb:equal(?qNomId1, ?qNomId2)
^ qif:TargetValue(?qNom, ?qNomVal)
^ qif:Value(?qAct, ?qActVal)
-> sqwrl:select(?qAct, ?qActVal, ?qNom, ?qNomVal)
```

Listing 5. SQWRL query to list failed inspection results.

Our main goal is to explore and prove out the concept of fusing as-designed and as-inspected for realizing benefits of the digital thread and smart manufacturing, in general. In doing so, we integrated STEP and QIF KGs based on STEP-QIF mapping specification rather than developing an individual translator. The ontology- and KG-based integrated knowledge base will have benefits in terms of data reuse as well as long-term archiving and retrieval. We also introduced rules and queries for finding discrepancies in information from different sources and querying information to support decision-making in the design and inspection stages. We verified that the queries can be effectively used by discovering actual discrepancies in the dataset. For demonstration, we leveraged a three-component assembly dataset, one of the TDPs published on the NIST SMS Test Bed.

Although our goal is to best leverage instance files in the dataset, the current integrated knowledge base lacks semantics because it was built upon KGs generated from instances without full taxonomy, especially in the case of QIF. From that perspective, the quality of the ontology with KGs completely depends on the capability of translators. Therefore, more translators need to be tested, especially for XSD/XML to OWL translation, to build an ontology which is semantically identical to the original XSD schema. Moreover, more discussion is warranted towards harmonizing and relating STEP and QIF [12,50].

More complex tasks in the assembly domain, such as tolerance stack-up analysis [34,51] and inferring tolerance or geometric constraint types [32,6,52], can also be supported by advanced rules and queries based upon KGs. Looking forward, 3D visualization might be an effective means for showcasing querying results. In the future, we plan to merge as-planned and as-executed data into the integrated knowledge base for enabling more functions and features for smart manufacturing capabilities. We believe that there are open research questions on leveraging ontology with KGs for the implementation of the digital thread and the more efficient realization of digital twin applications. We are actively tracking efforts related to standard ontology development for engineering systems, such as the Industrial Ontology Foundry [53]. Developing a consensus in industry for the use, development, compliance, and deployment of engineering ontologies is critical to realize the benefits of semantic web technologies, such as the concepts featured in this work.

qAct	qActVal	qNom	qNomVal
qif:DiameterCharacteristicActual_172	"4.425"^^xsd:double	qif:DiameterCharacteristicNominal_1	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_223	"4.344"^^xsd:double	qif:DiameterCharacteristicNominal_1	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_230	"4.458"^^xsd:double	qif:DiameterCharacteristicNominal_1	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_285	"4.316"^^xsd:double	qif:DiameterCharacteristicNominal_1	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_339	"4.457"^^xsd:double	qif:DiameterCharacteristicNominal_1	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_79	"4.424"^^xsd:double	qif:DiameterCharacteristicNominal_1	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_0	"4.286"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_130	"4.401"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_141	"4.387"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_159	"4.367"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_16	"4.319"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_164	"4.392"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_173	"4.316"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_193	"4.429"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_213	"4.273"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double
qif:DiameterCharacteristicActual_233	"4.283"^^xsd:double	qif:DiameterCharacteristicNominal_3	"4.0"^^xsd:double

Fig. 12. Result of a query for failed inspection results.

Disclaimer

No endorsement of any commercial product by NIST is intended. Commercial materials are identified in this report to facilitate better understanding. Such identification does not imply endorsement by NIST nor does it imply the materials identified are necessarily the best for the purpose.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank Allison Barnard Feeney, Sylvere Krima, Thomas Kramer, Hyunwoong Ko, Minchul Lee, and the anonymous peer-reviewers for their valuable comments and input.

References

[1] M. Helu, T. Hedberg Jr., Enabling smart manufacturing research and development using a product lifecycle test bed, Procedia Manuf. 1 (2015) 86–97.
[2] International Organization for Standardization, ISO 10303-1:1994 Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles, ISO, 1994.
[3] S.J. Kemmerer, STEP: the grand experience, NIST SP 939, 1999.
[4] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs, SEMANTiCS (Posters, Demos, SuCCESS) 48, 2016.
[5] H. Ko, P. Witherell, Y. Lu, S. Kim, D.W. Rosen, Data-to-Knowledge-to-Design Rule Transformation for Additive Manufacturing using Machine Learning and Knowledge Graph, Additive Manufacturing Under Revision, 2020.
[6] Y. Zhong, Y. Qin, M. Huang, W. Lu, W. Gao, Y. Du, Automatically generating assembly tolerance types with an ontology-based approach, Comput. Aided Des. 45 (11) (2013) 1253–1275.
[7] V. Premkumar, S. Krishnamurty, J.C. Wileden, I.R. Grosse, A semantic knowledge management system for laminated composites, Adv. Eng. Inform. 28 (1) (2014) 91–101.
[8] S. Chen, J. Yi, H. Jiang, X. Zhu, Ontology and CBR based automated decision-making method for the disassembly of mechanical products, Adv. Eng. Inform. 30 (3) (2016) 564–584.
[9] R. Barbau, S. Krima, S. Rachuri, A. Narayanan, X. Fiorentini, S. Foufou, R.D. Sriram, OntoSTEP: Enriching product model data using ontologies, Comput. Aided Des. 44 (6) (2012) 575–590.
[10] S. Krima, R. Barbau, X. Fiorentini, R. Sudarsan, R.D. Sriram, OntoSTEP: OWL-DL Ontology for STEP, NISTIR 7561, 2009.
[11] E. Morse, S. Heysiattalab, A. Barnard-Feeney, T. Hedberg Jr., Interoperability: linking design and tolerancing with metrology, Procedia CIRP 43 (2016) 13–16.
[12] A. Trainer, A.B. Feeney, T.D. Hedberg, Validation for Downstream Computer Aided Manufacturing and Coordinate Metrology Processes, NIST GCR 16-003, 2015.
[13] R. Huang, S. Zhang, X. Bai, C. Xu, Multi-level structuralized model-based definition model based on machining features for manufacturing reuse of mechanical parts, Int. J. Adv. Manuf. Technol. 75 (5–8) (2014) 1035–1048.
[14] T. Hedberg, J. Lubell, L. Fischer, L. Maggiano, A.B. Feeney, Testing the digital thread in support of model-based manufacturing and inspection, J. Comput. Inf. Sci. Eng. 16 (2) (2016) 021001.
[15] J. Camba, M. Contero, M. Johnson, P. Company, Extended 3D annotations as a new mechanism to explicitly communicate geometric design intent and increase CAD model reusability, Comput. Aided Des. 57 (2014) 61–73.
[16] M. Hallmann, S. Goetz, B. Schleich, Mapping of GD&T information and PMI between 3D product models in the STEP and STL format, Comput. Aided Des. 115 (2019) 293–306.
[17] S.P. Ruemler, K.E. Zimmerman, N.W. Hartman, T. Hedberg, A.B. Feeny, Promoting model-based definition to establish a complete product definition, J. Manuf. Sci. Eng. 139 (5) (2017) 051008.
[18] A. Trainer, T. Hedberg Jr., A.B. Feeney, K. Fischer, P. Rosche, Gaps analysis of integrating product design, manufacturing, and quality data in the supply chain using model-based definition, in: Proceedings of the ASME International Manufacturing Science and Engineering Conference, vol. 2, 2016.
[19] T. Hedberg, A.B. Feeney, M. Helu, J.A. Camelio, Toward a lifecycle information framework and technology in manufacturing, J. Comput. Inf. Sci. Eng. 17 (2) (2017) 021010.
[20] D.B. Kim, An approach for composing predictive models from disparate knowledge sources in smart manufacturing environments, J. Intell. Manuf. 30 (4) (2019) 1999–2012.
[21] PTC, Digital Twin: A Primer for Industrial Enterprises, https://www.ptc.com/en/resources/iiot/white-paper/digital-twin-industrial-enterprise, 2019.
[22] M. Helu, T. Hedberg Jr, A.B. Feeney, Reference architecture to integrate heterogeneous manufacturing systems for the digital thread, CIRP J. Manuf. Sci. Technol. 19 (2017) 191–195.
[23] M. Helu, A. Joseph, T. Hedberg Jr, A standards-based approach for linking as-planned to as-fabricated product data, CIRP Ann. 67 (1) (2018) 487–490.
[24] W.Z. Bernstein, T.D. Hedberg Jr, M. Helu, A.B. Feeney, Contextualising manufacturing data for lifecycle decision-making, Int. J. Product Lifecycle Manage. 10 (4) (2018) 326.
[25] L.V. Monnier, W.Z. Bernstein, S. Foufou, A proposed mapping method for aligning machine execution data to numerical control code, in, 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), 2019, pp. 66–72.
[26] M.-W. Cho, T.-I. Seo, Inspection planning strategy for the on-machine measurement process based on CAD/CAM/CAI integration, Int. J. Adv. Manuf. Technol. 19 (8) (2002) 607–617.
[27] S. Sivakumar, V. Dhanalakshmi, An approach towards the integration of CAD/CAM/CAI through STEP file using feature extraction for cylindrical parts, Int. J. Comput. Integr. Manuf. 26 (6) (2013) 561–570.
[28] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hübner, Ontology-based integration of information-a survey of existing approaches, in: OIS@ IJCAI, 2001.
[29] R. Sudarsan, S.J. Fenves, R.D. Sriram, F. Wang, A product information modeling framework for product lifecycle management, Comput. Aided Des. 37 (13) (2005) 1399–1411.
[30] A. Matsokis, D. Kiritsis, An ontology-based approach for product lifecycle management, Comput. Ind. 61 (8) (2010) 787–797.
[31] H. Panetto, M. Dassisti, A. Tursi, ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment, Adv. Eng. Inform. 26 (2) (2012) 334–348.
[32] W. Lu, Y. Qin, X. Liu, M. Huang, L. Zhou, X. Jiang, Enriching the semantics of variational geometric constraint data with ontology, Comput. Aided Des. 63 (2015) 72–85.
[33] J. Lee, H. Chae, C.-H. Kim, K. Kim, Design of product ontology architecture for collaborative enterprises, Expert Syst. Appl. 36 (2) (2009) 2300–2309.
[34] M.I. Sarigecili, U. Roy, S. Rachuri, Interpreting the semantics of GD&T specifications of a product for tolerance analysis, Comput. Aided Des. 47 (2014) 72–84.
[35] T.D. Hedberg, M. Bajaj, J.A. Camelio, Using graphs to link data across the product lifecycle for enabling smart manufacturing digital threads, J. Comput. Inf. Sci. Eng. 20 (1) (2020) 011011.
[36] R. Liu, G.-J. Duan, J. Liu, A framework for model-based integrated inspection, Int. J. Adv. Manuf. Technol. 103 (9–12) (2019) 3643–3665.
[37] J. Beetz, J. Van Leeuwen, B. De Vries, IfcOWL: A case of transforming EXPRESS schemas into ontologies, AI EDAM 23 (1) (2009) 89–101.
[38] P. Pauwels, W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, Automat. Construct. 63 (2016) 100–133.
[39] P. Pauwels, D. Van Deursen, IFC/RDF: adaptation, aggregation and enrichment, in: First International Workshop on Linked Data in Architecture and Construction, 2012, pp. 1–3.
[40] Y.F. Zhao, J.A. Horst, T.R. Kramer, W. Rippey, R.J. Brown, Quality information framework–integrating metrology processes, IFAC Proc. Vol. 45 (6) (2012) 1301–1308.
[41] M. Hacherouf, S.N. Bahloul, C. Cruz, Transforming XML documents to OWL ontologies: A survey, J. Informat. Sci. 41 (2) (2015) 242–259.
[42] D. Lacoste, K.P. Sawant, S. Roy, An efficient XML to OWL converter, in, Proceedings of the 4th India Software Engineering Conference, ACM, 2011, pp. 145–154.
[43] T.R. Kramer, B.H. Marks, C.I. Schlenoff, S.B. Balakirsky, Z. Kootbally, A. Pietromartire, Software tools for XML to OWL translation, NISTIR 8068, 2015.
[44] Ontmalizer, https://www.openhub.net/p/ontmalizer.
[45] M.J. O'Connor, A. Das, Acquiring OWL ontologies from XML documents, in: Proceedings of the Sixth International Conference on Knowledge Capture, ACM, 2011, pp. 17–24.
[46] T. Rodrigues, P. Rosa, J. Cardoso, Mapping XML to existing OWL ontologies, International Conference WWW/Internet, Citeseer, 2006, pp. 72–77.
[47] XML Tab, https://protegewiki.stanford.edu/wiki/XML_Tab.
[48] QIF Standard, https://qifstandards.org/.
[49] T.D. Hedberg Jr, M.E. Sharp, T.M. Maw, M.M. Rahman, S. Jadhav, J.J. Whicker, A. B. Feeney, M. Helu, Design, manufacturing, and inspection data for a three-component assembly, J. Res. Nat. Inst. Stand. Technol. 124 (2019).
[50] S. Heysiattalab, E. Morse, From STEP to QIF: Product, manufacturing information, in: 31th ASPE Annual Meeting, vol. 65, 2016, pp. 312–317.
[51] M.I. Sarigecili, U. Roy, S. Rachuri, Enriching STEP product model with geometric dimension and tolerance information for one-dimensional tolerance analysis, J. Comput. Inf. Sci. Eng. 17 (2) (2017) 021004.
[52] H. Vilmart, J.-C. Léon, F. Ulliana, From CAD assemblies toward knowledge-based assemblies using an intrinsic knowledge-based assembly model, Comput.-Aided Des. Appl. 15 (3) (2018) 300–317.
[53] E. Wallace, D. Kiritsis, B. Smith, C. Will, et al., The industrial ontologies foundry proof-of-concept project, in, IFIP International Conference on Advances in Production Management Systems, Springer, 2018, pp. 402–409.