Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 27 (2015) 124 – 130

www.elsevier.com/locate/procedia

13th CIRP conference on Computer Aided Tolerancing

Ontology-based semantic interpretation of cylindricity specification in the next-generation GPS

Yuchu Qin[a], Wenlong Lu[a,]*, Qunfen Qi[b], Xiaojun Liu[a], Liping Zhou[a], Tukun Li[b]

[a]*The State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering,Huazhong University of Science and Technology, Wuhan 430074, PR China*

[b]*EPSRC Centre for Innovative Manufacturing in Advanced Metrology, University of Huddersfield, Huddersfield, HD1 3DH, UK*

* Corresponding author. Tel.: +86-027-87557835; fax: +86-027-87543670.*E-mail address:* hustwenlong@mail.hust.edu.cn

Abstract

Cylindricity specification is one of the most important geometrical specifications in geometrical product development. This specification can be referenced from the rules and examples in tolerance standards and technical handbooks in practice. These rules and examples are described in the form of natural language, which may cause ambiguities since different designers may have different understandings on a rule or an example. To address the ambiguous problem, a categorical data model of cylindricity specification in the next-generation Geometrical Product Specifications (GPS) was proposed at the University of Huddersfield. The modeling language used in the categorical data model is category language. Even though category language can develop a syntactically correct data model, it is difficult to interpret the semantics of the cylindricity specification explicitly. This paper proposes an ontology-based approach to interpret the semantics of cylindricity specification on the basis of the categorical data model. A scheme for translating the category language to the OWL 2 Web Ontology Language (OWL 2) is presented in this approach. Through such a scheme, the categorical data model is translated into a semantically enriched model, i.e. an OWL 2 ontology for cylindricity specification. This ontology can interpret the semantics of cylindricity specification explicitly. As the benefits of such semantic interpretation, consistency checking, inference procedures and semantic queries can be performed on the OWL 2 ontology. The proposed approach could be easily extended to support the semantic interpretations of other kinds of geometrical specifications.

© 2015 The Authors. Published by Elsevier B.V This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the organizing committee of 13th CIRP conference on Computer Aided Tolerancing

*Keywords:*Semantic interpretation; Cylindricity specification; Category language; OWL 2; Ontology

1. Introduction

The trend in global manufacturing urgently requires a rigorous and systematic common language to characterize the geometrical characteristics in geometric product development. An international technical language, called Geometrical Product Specifications (GPS)[1], has been created to satisfy this requirement. GPS includes various kinds of geometrical specifications. Cylindricity specification is one of the most important geometrical specifications. In practice, cylindricity specification can be referenced from the rules and examples in tolerance standards [2] and technical handbooks [3]. These rules and examples are described in the form of natural language, which may lead to ambiguities because different designers may have different understandings on a rule or an example.

To address such ambiguous problem, the cylindricity specification should be formalized. A typical formalized method was proposed by Lu et al. [4] at the University of Huddersfield. The method used category language [5] to establish a data model of cylindricity specification in the next-generation GPS. Then an information system for complex cylindricity specification data manipulation, named VirtualGPS [6], was developed on the basis of the categorical data model. VirtualGPS system enables designers to query specific rules to design cylindricity specification. The ambiguous problem caused by describing cylindricity specification in natural language has been well solved. However, VirtualGPS has great difficulty in interpreting the semantics of cylindricity specification explicitly because category language can only develop a syntactically correct data model instead of developing a semantically correct one.

2212-8271 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the organizing committee of 13th CIRP conference on Computer Aided Tolerancing
doi:10.1016/j.procir.2015.04.054

To interpret the semantics of cylindricity specification explicitly, an ontology-based approach on the basis of the categorical data model is proposed in this paper. Ontology [7], an explicit specification of a conceptualization, is well-known for having rigorous logic-based and computer-interpretable semantics. Although the application of ontology has its root in the field of the Semantic Web, it has been extended to many other fields. In the field of product development, ontology has been used to enrich product data semantics [8,9], model and reason out assembly tolerance types [10] and improve the interoperability of industrial information systems [11]. The role of ontologies with well-defined semantics is highlighted in [12]. In the proposed ontology-based approach, the categorical data model of cylindricity is translated into an OWL 2 ontology by designing a scheme to translate category language to OWL 2 [13], an ontology representation language developed by the World Wide Web Consortium (W3C). Then the semantics of cylindricity specification can be explicitly interpreted and the consistency checking, inference procedures and semantic queries can be performed.

The organization of the paper is as follows. Section 2 provides the details of the semantic interpretation approach. Section 3 explains the implementations of the approach and gives some examples to illustrate the benefits of the implemented approach. Section 4 carries out some discussions and Section 5 draws some conclusions.

2. Semantic interpretation approach

This section describes a mechanism to interpret the semantics of cylindricity specification explicitly. The schematic representation of this mechanism is shown in Fig. 1. The first step is to design a scheme to translate category language into OWL 2. The second step is to translate the categorical data model of cylindricity specification to an OWL 2 ontology with the use of the designed scheme. Then the semantic interpretation of cylindricity can be implemented. Such interpretation is reflected in three benefits: (1) Consistency checking by the reasoning mechanism of OWL 2 Description Logic (DL) [14]. (2) Inference procedures by the reasoning mechanisms of OWL 2 DL and Semantic Web Rule Language (SWRL) [15]. (3) Semantic queries by DL query mechanisms. The details of these three steps are discussed in the following three sub-sections, respectively.

2.1. Translation from category language to OWL 2

The translation of the basic concepts in category language is illustrated through the example shown in Fig. 2 [4]. In this example, a category named Feature is described. A Feature has an initial object Fe#, a Feat_type, a Ref_diameter, a Length_G and a DOF.

The concept of category in category language is similar to the concept of class in object-oriented modeling language since category can be seen as the abstraction of real-world individuals and can be organized in hierarchies (subcategory). For example, the category Feature in Fig. 2 can be seen as the abstraction of all real-world features. An object in category language specifies a relation between a category and a value. For instance, the object Feat_type specifies the feature type (spherical, cylindrical, planar, helical, revolute, prismatic or complex) of the category Feature. In addition to category, subcategory and object, morphism is also an important concept in category. A morphism is defined as an inheritance relation from one object to this object or the other object [5]. As shown in Fig. 2, four morphisms from the initial object Fe#to the objects Feat_type, Ref_diameter, Length_G and DOF describes the inheritance relations between these five objects. Based on the above analysis, the concepts category, subcategory, object and morphism in category language can be naturally translated to class, subclass of, data property and sub data property of in OWL 2, respectively.

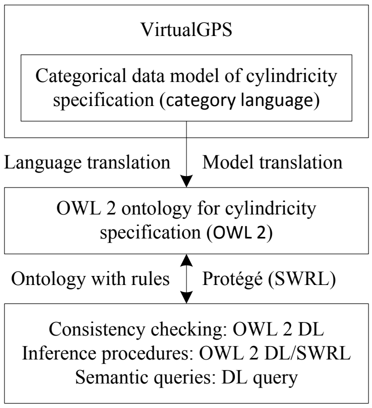

Fig. 1.The schematic representation of the semantic interpretation approach. SWRL: Semantic Web Rule Language. DL: Description Logic.

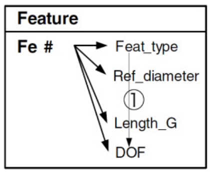

Fig. 2.The categorical data model for the partition of a cylindrical feature [4]. Fe# is the initial object in the category Feature. Feat_type denotes the feature type. Ref_diameter denotes the diameter of each circumferential section. Length_G denotes the length of the generatrix. DOF denotes the degree of freedom. ① denotes the degree of freedom can be determined by the type of geometrical feature.

An OWL 2 ontology may include the assertions related to classes (TBox), properties (RBox) and individuals (ABox). In the designed translation scheme, a categorical data model is translated into an OWL 2 ontology that includes the definitions of the classes, properties and individuals. Table 1 summarizes the OWL 2 translation of the basic concepts in category language.

After translating the basic concepts in category language, the translation of some manipulations can be considered. Pull back and functor are the two manipulations used in the categorical data model of cylindricity specification in [4]. The schematic representation of pull back is illustrated in Fig. 3 [4]. It can be seen in the figure that if a, b, c and d are four arbitrary objects in a category C, f is a morphism from b to a, g is a morphism from c to a, h is a morphism from d to b, k is a morphism from d to c and there exists a unique morphism z that is from d to x (x is an object in C), then there exist two morphisms, in which one is from x to b (morphism p) and the other is from x to c (morphism q). Such object x and morphisms p and q are called as the pull back of f and g.

Table 1.Translation of the basic concepts from category language to OWL 2.

Category language	OWL 2
Categorical data model	OWL 2 ontology
Category	Class
Subcategory	Subclass of
Object	Data property
Morphism	Sub data property of

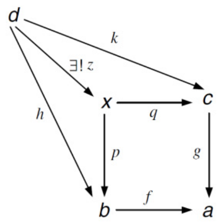

Fig. 3.The schematic representation of the pull back manipulation [4].

Intuitively, pull back manipulation describes the relations among objects. Because objects have been translated to data property, these relations can be seen as the relations among data properties. To translate the pull back manipulation is in fact to represent the relations between two properties. Unfortunately, OWL 2 does not provide a mechanism to realize such representation. As a result, OWL 2 is combined with SWRL. Let five OWL 2 class A, B, C, D and X denote the five objects a, b, c, d and x in Fig. 3 (these objects are in fact mapping to data properties in OWL 2). Then the morphisms f, g, h, k, p, q and z in Fig. 3 can be seen as some binary relations among A, B, C, D and X. By this way, the pull back manipulation in Fig. 3 can be translated to an OWL 2/SWRL rule shown in Table 2.

Table 2.Translation of the pull back manipulation in category language.

Antecedent	Consequent
$A(?a)$, $B(?b)$, $C(?c)$, $D(?d)$, $X(?x)$, $f(?b, ?a)$, $g(?c, ?a)$,	$p(?x, ?b)$,
$h(?d, ?b)$, $k(?d, ?c)$, $z(?d, ?x)$	$q(?x, ?c)$

Now the translation of functor manipulation is considered. In category language, functor manipulation can be simply seen as a mapping from one category to the other [5]. Since category has been translated to class, functor manipulation is naturally translated to object property, in which the class translated by the source category is defined as the domain of this object property and the class mapped by the target category is defined as the range of the object property.

2.2. OWL 2 ontology for cylindricity specification

The construction of an OWL 2 ontology for cylindricity specification is done by translating the categorical data model of cylindricity specification in [4] using the designed scheme for the translation from category language to OWL 2. The first step in this construction is to identify the primitive symbols that often include a set of class names N_C, a set of property names N_P and a set of individual names N_I. Because category in category language has been translated to class in OWL 2, the class names in the OWL 2 ontology are the category names in the categorical data model in [4]. So the set N_C can be achieved as follow:

N_C={Callout, Extraction, Restriction, Sampling, Instrument, Evaluation, Parameter, Association, Filtration, Feature, Spherical, Cylindrical, Planar, Helical, Revolute, Prismatic, Complex } (1)

Similarly, the property names in the OWL 2 ontology are the object names in the model in [4] because object in category language has been translated to data property. The set NP is easily achieved as follow:

N_P={hasC#, hasSymbol, hasSpec_value, hasRest, hasFilt_name_R, hasFilt_name_G, hasCutoff_wavelength, hasUpper_wavelength, hasUpper_frequency, hasLower_frequency, hasAsso, hasPara, hasSampling_strategy, hasE#, hasSampling, hasInstrument, hasR#, hasRest_name, hasS#, hasSamp_space_R, hasSamp_space_G, hasSamp_point_R, hasSamp_point_G, hasNum_cutoff_R, hasNum_cutoff_G, hasSamp_strategy, hasSamp_length_R, hasSamp_length_G, hasI#, hasInstru_name, hasInstru_type, hasZ_resolution, hasSpatial_range, hasTip_radius, hasEv#, hasMeas_value, hasMeas_uncertainty, hasP#, hasPara_name, hasPara_value, hasEvaluation_length_R, hasEvaluation_length_G, hasA#, hasAsso_name, hasC, hasO, hasFi#, hasFilt_name, hasFilt_type, hasUplimit_frequency, hasLowlimit_frequency, hasUpper_wavelength_fi, hasLower_wavelength_fi, hasFe#, hasFeat_type, hasRef_diameter, hasLength_G, hasDOF } (2)

The values of the objects in the categorical data model of cylindricity specification in [4] are all translated to the individuals in the OWL 2 ontology. Thus the set N_I is a set of values names. Further, individuals may always be different for different engineering examples. The individuals for a specific engineering example will be defined in Section 3.

Starting with the defined sets N_C, N_P and N_I, the construction process of the OWL 2 ontology can be carried out through the following steps:

(1) *Define and construct a TBox for the OWL 2 ontology*. A TBox or a terminology is defined as a finite set of terminological axioms that are in the forms of $CN \sqsubseteq CE$ and $CN \equiv CE$ ($CN \in N_C$, CE is a class expression) [14]. According to the actual meaning of each class in N_C, a TBox for the OWL 2 ontology is constructed as follow:

$\mathcal{T}_{CS}$={Callout⊑owl:Thing, Extraction⊑owl:Thing,
Restriction⊑owl:Thing, Sampling⊑owl:Thing,
Instrument⊑owl:Thing, Evaluation⊑owl:Thing,
Parameter⊑owl:Thing, Association⊑owl:Thing,
Filtration⊑owl:Thing, Feature⊑owl:Thing,
{Spherical, Cylindrical, Planar, Helical, Revolute,
Prismatic, Complex}∋*f*⊑ Feature} (3)

(2) *Define and construct an RBox for the OWL 2 ontology*. An RBox or a role hierarchy is defined as a finite set of role inclusion axioms that are in the form of $P_1 \sqsubseteq P_2 (P_1, P_2 \in N_P)$ [14]. Since morphism in category language has been translated to sub data property of in OWL 2, an RBox for the OWL 2 ontology can be constructed as follow:

$\mathcal{R}_{CS}$={{hasC#, hasE#, hasR#, hasS#, hasI#, hasEv#,
hasP#, hasA#, hasFi#, hasFe#}∋*top*⊑
owl:topDataProperty,
{hasSymbol, hasSpec_value, hasRest,
hasFilt_name_R, hasFilt_name_G,
hasCutoff_wavelength,
hasUpper_wavelength, hasUpper_frequency,
hasLower_frequency, hasAsso, hasPara,
hasSampling_strategy}∋*c*⊑hasC#,
{hasSampling, hasInstrument}∋*e*⊑hasE#,
{hasRest_name}∋*r*⊑hasR#,
{hasSamp_space_R, hasSamp_space_G,
hasSamp_point_R, hasSamp_point_G,
hasNum_cutoff_R, hasNum_cutoff_G,
hasSamp_strategy, hasSamp_length_R,
hasSamp_length_G}∋*s*⊑hasS#,
{hasInstru_name, hasInstru_type,
hasZ_resolution, hasSpatial_range,
hasTip_radius}∋*i*⊑hasI#,
{hasMeas_value,
hasMeas_uncertainty}∋*ev*⊑hasEv#,
{hasPara_name, hasPara_value,
hasEvaluation_length_R,
hasEvaluation_length_G}∋*p*⊑hasP#,
{hasAsso_name, hasC, hasO}∋*a*⊑hasA#,
{hasFilt_name, hasFilt_type,
hasUplimit_frequency,
hasLowlimit_frequency,
hasUpper_wavelength_fi,
hasLower_wavelength_fi}∋*fi*⊑hasFi#,
{hasFeat_type, hasRef_diameter,
hasLength_G, hasDOF}∋*fe*⊑hasFe# } (4)

(3) *Define the facet of each property in the constructed RBox*. The facets of the properties in $\mathcal{R}_{CS}$ can be easily defined through analyzing the objects in the categorical data model in [4]. For example, the domain of the data property hasSpec_value is class Callout and the range of this property can be defined as the data type float.The domain and range of other properties in $\mathcal{R}_{CS}$ can be defined in a similar way.

(4) *Define and construct an ABox for the OWL 2 ontology*. An ABox is defined as a finite set of assertions that are in the forms of $CE(x)$, $P(x, y)$ and $\neg P(x, y)$ ($CE$is a class expression, $P \in N_P$, x, $y \in N_I$) [14]. According to this definition and the categorical data model in [4], an ABox named $\mathcal{A}_{CS}$ can be constructed through instantiating the classes in N_C and the data properties in N_P.

Through the above four steps, the OWL 2 ontology for cylindricity specifications is constructed and can be defined as a finite set $O_{CS} = \{ \mathcal{T}_{CS}, \mathcal{R}_{CS}, \mathcal{A}_{CS} \}$.

2.3. Semantic interpretation of cylindricity specification

The logic basis of OWL 2 is DL SROIQ(D) [14], a knowledge representation and reasoning language for authoring OWL 2 DL ontologies.This language is capable of defining domain specific concepts (classes) and roles (properties) with a predefined and well understood formalism. Concepts (classes) are used to denote and describe the domain objects, while roles (properties) are used to denote and describe the relations between concepts. Concepts (classes) and roles (properties) are the main components of a (an) knowledge base (ontology).

DL SROIQ(D) can provide the maximum expressive capability and a highly efficient reasoning algorithm under the prerequisite of ensuring computational completeness and decidability. With the support of this reasoning algorithm, consistency checking of the ontology, inference procedures on the ontology and semantic queries in the ontology can be automatically done. These capabilities are also called as semantic interpretation capability. Category language does not have such capability. This is why the language has been translated to OWL 2 and the categorical data model is translated to the OWL 2 ontology.

To be more specific, the translation of the categorical data model in [4] to the OWL 2 ontology O_{CS} brings the following three benefits [8]:

(1) *Consistency checking of*O_{CS}. Consistency checking procedure can be applied at both class level and individual level. This procedure determines whether an instantiation of a class would create an inconsistency in the ontology at the class level and checks whether an individual of a class satisfies the definition of this class at the individual level.

Through using a DL reasoner (e.g. Pellet or HermiT), the consistency of O_{CS} can be automatically checked at both class level and individual level. An inference procedure on O_{CS} can be applied only if O_{CS} is checked to be consistent.

(2) *Inference procedures on*O_{CS}. An inference procedure takes the explicit knowledge in a context as input and uses certain problem solving strategies to achieve the implicit knowledge in this context. In short, an inference procedure is a process to reach new conclusions. It is performed by a DL reasoner on O_{CS}. After applying the inference procedures on O_{CS}, the new knowledge that is included in an enriched version of O_{CS} will become available. A semantic querying mechanism can be used to query this new knowledge.

(3) *Semantic queries in*O_{CS}. Semantic queries aim to retrieve some specific knowledge from a large amount of knowledge. The OWL 2 ontology O_{CS} is first checked for consistency, then inferred upon and finally queried. The widely used query method in Protégé Desktop 4.x is the DL query method. This method uses the knowledge reasoning mechanism to realize such semantic queries. Some examples will be given to illustrate the processes of semantic interpretation (i.e. consistency checking, inference procedures and semantic queries) in next section.

3. Implementations and examples

This section first presents and discusses aspects related to the implementation of the proposed semantic interpretation approach, then uses some examples to illustrate the processes of the semantic interpretation of cylindricity specification.

3.1. Implementations

The implementation process of the proposed semantic interpretation approach is facilitated through using Protégé [16], an ontology editor and knowledge acquisition system which offer an integration environment of creating, editing and saving ontologies in a visual way. Protégé also supports direct in-memory connections to DL reasoners such as Pellet and HermiT. Facilitating by Protégé, the implementation process is carried out according to the following steps:

(1)*Create classes and their hierarchies*. Classes in the OWL 2 ontology O_{CS} are created on the basis of the set N_C in Expression (1). The hierarchies of classes can be created according to the TBox $\mathcal{T}_{CS}$ in Expression (3).

(2) *Create properties and their hierarchies*. Properties in O_{CS} can be created according to the set N_P in Expression (2). The hierarchies of properties are created on the basis of the RBox $\mathcal{R}_{CS}$ in Expression (4).

(3) *Create OWL 2/SWRL rules*. OWL 2/SWRL rules can be created according to the translation results of all the pull back manipulations in the categorical data model of cylindricity specification in [4].

(4) *Instantiate classes and properties*. Classes and properties can be instantiated according to the ABox $\mathcal{A}_{CS}$.

3.2. Examples

An example of the cylindricity specification of the intermediate shaftof a gear reduceris given to verify the proposed semantic interpretation approach. As illustrated in Fig. 4, the tolerance types and values of the intermediate shaft have been determined by the ontology-based approach in [10]. The cylindricity specificationindicatedon the features$_1$(p_{10}) is taken as an example to illustrate the process of the semantic interpretation of cylindricity specification. Detailed semantic interpretation process is as follows:

(1) *Determine the cylindricity specification in the next generation GPS indicated on*s_1(p_{10}).UsingVirtualGPS [6], the cylindricity specification in the next-generation GPS corresponding to the cylindricity specificationindicated ons_1(p_{10}) is determined and shown in Table 3.

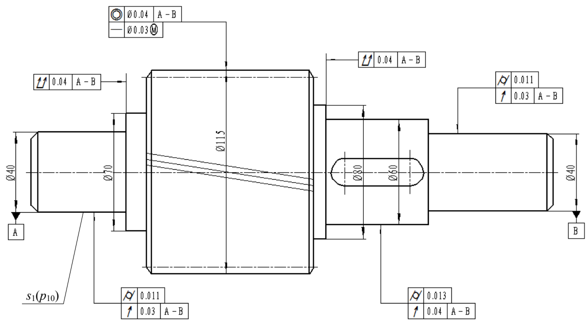

Fig. 4.The tolerance types and values of the intermediate shaft [10].

Table 3.The cylindricity specification in the next-generation GPS corresponding to the cylindricity specificationindicated ons_1(p_{10}). FPLG denotes linear profile Gaussian filter. UPR denotes undulations per revolution. *CYL*t denotes peak-to-valley cylindricity deviation. LSCY denotes the least-squares association method is used to obtain the reference cylinder. BC denotes bird cage sampling strategy. 10000f denotes infinity.

Specification type	Specification	Instance name
Feature type	Cylindrical	s1p10
Tolerance type	Cylindricity	cylindricity
Tolerance value	0.011 mm	0.011f
Geometric requirement	Null	null
Axial filter type	FPLG	fplg1
Cutoff wavelength	0.8 mm	0.8f
Upper wavelength	∞	10000.0f
Radial filter type	FPLG	fplg2
Cutoff frequency	150 UPR	150.0f
Lower cutoff frequency	1 UPR	1.0f
Evaluation parameter	*CYL*t	cylt
Association datum	LSCY	lscy
Sampling strategy	BC	bc

(2) *Instantiate the OWL 2 ontology*O_{CS}. The OWL 2 ontology O_{CS} is instantiated according to the determined cylindricity specification in the next generation GPS indicated ons_1(p_{10})(Table 3) and then theABox $\mathcal{A}_{CS}$can be augmented.

(3) *Consistency checking of*O_{CS}. At the class level, the DL reasoner uses the class definitions to determine whether a class is consistent or not. For example, class Cylindrical and class Planar are defined as two disjoint classes. Assume class Hole is defined as a subclass of Cylindrical. If the class Hole is asserted as a subclass of Planar, the reasoner will detect an inconsistency since Cylindrical and Planar are disjoint (Fig. 5). The consistency at the individual level can be checked by the DL reasoner in a similar way. Checking the consistency of O_{CS}, which is not available in the categorical data model, is a necessary condition to use an inference procedure.

(4) *Inference procedures on*O_{CS}. Once the DL reasoner has applied all the inference procedures on O_{CS}, the new knowledge can be made available. This dynamic modification also cannot be implemented in the categorical data model. As an example, the new knowledge inferred by the SWRL rule (Fig. 6) for representing the relations of the arrows 26, 34, 27, 28, 35, 29, 30, 36, 31, 32 and 33 in [4] is shown in Fig. 7.

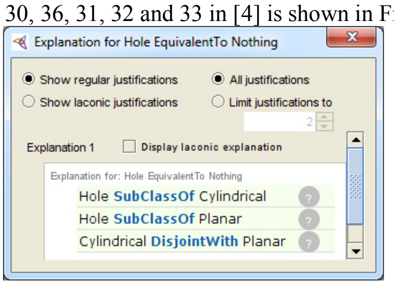

Fig. 5.An inconsistency at the class level.

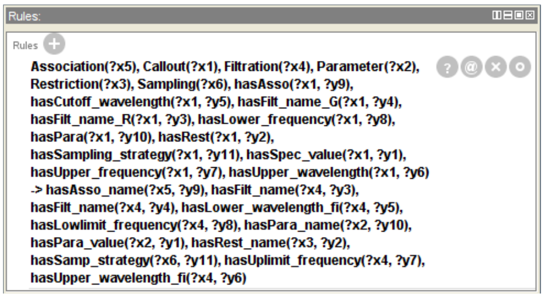

Fig. 6.An OWL 2/SWRL rule for representing the relations of the arrows26, 34, 27, 28, 35, 29, 30, 36, 31, 32 and 33 in [4].

Once the new knowledge has been inferred, one can use a querying mechanism to query this new knowledge in the semantically enriched version of the original O_{CS}.

(5) *Semantic queries in*O_{CS}. It is now possible to take advantage of the semantically enriched O_{CS} to query the inferred knowledge. For example, to query the instances that have a parameter value 0.011f, one may need to input "hasPara_value value 0.011f", the result will be outputted as "p_s1p10" after executing this DL query.

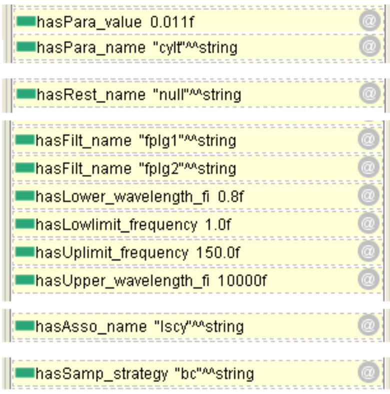

Fig. 7.The new knowledge inferred by the rule in Fig. 6.

4. Conclusions

This paper proposes an ontology-based approach to explicitly interpret the semantics of cylindricity specification. In this approach, a scheme for mapping category language to OWL 2 ontology representation language is designed and the categorical data model of cylindricity specification is translated to an OWL 2 ontology according to this mapping scheme. Since OWL 2 is based on description logic, the proposed semantic interpretation approach has rigorous logic-based and computer interpretable semantics to interpret the semantics of cylindricity specification explicitly. Benefiting from such semantic interpretation, this approach has the ability to check the consistency of the cylindricity specification data model, the ability to infer new knowledge from the checked model and the opportunity of performing semantic queries on the inferred model. The approach could be easily extended to support the semantic interpretations of other kinds of geometrical specifications.

Acknowledgements

The authors would like to acknowledge the financial support by the National Natural Science Foundation of China (No. 51475190 and No. 51275192) and National Basic Research Program of China (2014CB046705).

References

[1] Dantan JY, Ballu A, Mathieu L. Geometrical product specifications—model for product life cycle. Comput-Aided Des 2008; 40(4): 493-501.
[2] ISO 12180-1. Geometrical product specifications (GPS)—Cylindricity—Part 1: Vocabulary and parameters of cylindrical form. Geneva: International Organization for Standardization; 2011.
[3] VilleneuveF, MathieuL.Geometric Tolerancing of Products. Hoboken:John Wiley & Sons; 2010.
[4] Lu W, Jiang X, Liu X, et al.. Modeling the integration between specifications and verification for cylindricity based on category theory. Meas Sci Tech 2010; 21(11): 115107.
[5] Awodey S. Category theory. Oxford:Clarendon Press; 2006.

[6] Xu Y, Xu Z, Jiang X, et al.. Developing a knowledge-based system for complex geometrical product specification data manipulation. Knowl-Based Syst 2011; 24(1): 10-22.

[7] Gruber TR. A translation approach to portable ontology specifications. Knowl Acquisition 1993; 5(2): 199-220.

[8] Barbau R, Krima S, RachuriS, et al.. OntoSTEP: Enriching product model data using ontologies. Comput-Aided Des 2012; 44(6): 575-590.

[9] Sarigecili MI, Roy U, Rachuri S. Interpreting the semantics of GD&T specifications of a product for tolerance analysis. Computer-Aided Design 2014; 47(2): 72-84.

[10] Zhong Y, Qin Y, Huang M, et al.. Automatically generating assembly tolerance types with an ontology-based approach. Comput-Aided Des 2013; 45(11): 1253-1275.

[11] Fortineau V, Paviot T, Lamouri S. Improving the interoperability of industrial information systems with description logic-based models—the state of the art. Comput Ind 2013; 64(4): 363-375.

[12] Chandrasegaran SK, Ramani K, Sriram RD, et al.. The evolution, challenges, and future of knowledge representation in product design systems. Comput-Aided Des 2013; 45(2): 204-228.

[13] W3C. OWL 2 Web Ontology Language Document Overview (Second Edition), 11 Dec 2012. http://www.w3.org/TR/owl2-overview/.

[14] Horrocks I, Kutz O, Sattler U. The even more irresistible SROIQ. Palo Alto:AAAI Press; 2006. p. 68-78.

[15] Horrocks I, Patel-Schneider PF, Boley H, et al.. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 21 May 2004.http://www.w3.org/Submission/SWRL/.

[16] Stanford Center for Biomedical Informatics Research. Protégé Desktop 4.1, 2011. http://protege.stanford.edu/.