Int J Adv Manuf Technol (2018) 94:4187–4197
DOI 10.1007/s00170-017-1077-4

ORIGINAL ARTICLE

An ontology-based modelling and reasoning framework for assembly sequence planning

Lihong Qiao[1] · Yifan Qie[1] · Zuowei Zhu[1] · Yixin Zhu[1] · Uzair Khaleeq uz Zaman[2] · Nabil Anwer[3]

Received: 23 February 2017 / Accepted: 12 September 2017 / Published online: 30 September 2017

Abstract In the current era of increased customization, changing manufacturing systems and business globalization, effective use of product design information and knowledge generated from the product model can facilitate the decision-making of an assembly sequence by providing feasible product relationships and a viable semantic foundation. To enrich such semantics, a geometry-enhanced ontology modelling and reasoning framework is proposed in this paper to explicitly express relevant concepts for assembly sequence planning (ASP). A rule-based reasoning mechanism based on Ontology Web Language Description Logics (OWL-DL) and Semantic Web Rule Language (SWRL) is also suggested to clarify implicit relations by incorporating reasoning units (RUs) to process complex geometric information. This framework is then validated with a complex case study related to assembly sequence planning.

Keywords Assembly sequence planning · Geometry · Ontology · Reasoning

✉ Yixin Zhu
chunic@buaa.edu.cn

[1] School of Mechanical Engineering and Automation, Beihang University, 37 Xueyuan Road, Haidian District, Beijing 100191, China

[2] Laboratoire de Conception Fabrication Commande (LCFC), Ecole Nationale Supérieure d'Arts et Métiers (ENSAM), Metz Cedex 3, France

[3] LURPA, Ecole Normale Supérieure Paris-Saclay, University Paris-Saclay, 61, Avenue du Président Wilson, 94235 Cachan Cedex, France

1 Introduction

The assembly process is an important manufacturing activity, in which a product is created combining the processes of design, engineering, manufacturing and logistics in an efficient and cost-effective manner. According to Samy and ElMaraghy [1], assembly consumes up to 50% of total production time and accounts for more than 20% of total manufacturing cost.

Product modelling and assembly process planning have become active research topics since the early 1980s. The product model has emerged as a comprehensive concept for capturing geometric data and semantic information during the product lifecycle [2]. The assembly process is the result of assembly sequence planning (ASP), which considers available assembly resources to improve the design, simplify maintenance, and reduce the cost of production [3, 4]. Over the past decades, many researchers have developed various models and software solutions regarding assembly process. The development of computer-aided process planning (CAPP) techniques has facilitated ASP by deriving the best sequence of assembly operations given the geometrical representation of the assembly [5].

Although these models and techniques have been extensively investigated, inherent limitations in the existing work still exists when it comes to the knowledge sharing between the product model and assembly process planning. The effective reasoning of such knowledge also remains to be addressed. Product design relies on geometric information of the product, whereas assembly process planning is based on both the process information and the underlying geometric information in product models. Shape, position, contact, and mobility of parts are the main factors that must be considered in an assembly process. Moreover, ASP needs to incorporate such information to enrich the semantics in model representation and enhance the decision-making capabilities. Ontology modelling, one of the commonly used modelling methods in data management, can express unified, structural and semantic information and allows for reasoning capabilities due to the formal and logic-based specifications underlying in the information model. In other words, it makes the implicit information explicit [6].

In this context, a geometry-enhanced ontology modelling and reasoning framework for ASP is proposed in this paper. In the framework, the ontology model of assembly operations is established considering 'product geometry'. A reasoning mechanism is also proposed to infer the sequence of assembly operations based on predefined rules and the ontology model. The overall paper is structured as follows: Sect. 2 highlights the research background of the ontology-based modelling, representation and reasoning framework in the context of ASP, Sect. 3 explains the proposed ontology modelling and reasoning framework for ASP along with the discussion of construction of rules and reasoning units (RUs), Sect. 4 presents a case study to validate the framework, and Sect. 5 presents the conclusion drawn.

2 Literature review

Ontology is a formal way to represent knowledge within a given domain. As it enables concept elicitation and generalization and offers more explicitness to define properties and relationships between the concepts, multiple ontologies and data models have been developed to represent each stage of product life cycle over the past decade. Moreover, it supports inference through deductive reasoning and has been applied in many domains such as medical information science, geographic information system, enterprise modelling, organization learning and software engineering [7–12].

The traditional approaches for assembly modelling and ASP mainly focused on the relations among parts of a product. Graph-based methodologies, such as liaison sequence graph (LSG) and AND/OR graph, have been proposed to deduce feasible assembly sequences [13–15]. The representation, however, is mostly done by precedence diagrams or graphs that include parameters such as process times and levels of assembly by incorporating assembly precedence relations (APRs) derived from a series of structured questions [16]. But since assembly plan generation methods alone are not enough, it is imperative to study assembly plan relations methods that can automatically or semi-automatically generate ASPs by following various approaches such as utilizing neural network algorithms and design structure matrices [17]. Such an approach also helps in reducing the complexity of ASPs wherein assembly precedence denotes assembly order between parts by considering important qualitative assembly constraints like geometrical interference, accessibility, assembly stability and invisibility [18]. This further implies that it is imperative to integrate knowledge in the decision-making of ASP.

Zha et al. [19] proposed a novel approach for the automatic generation, selection and evaluation of assembly plans. Tao and Hu [20] also worked on a contact relation analysis approach to ASP from the perspective of reverse geometric reasoning. However, to make their ASP feasible, assembly tools and decision related to connection types between parts were needed. This also shows that to integrate knowledge into ASP, an efficient reasoning mechanism is required which can better support decision-making in later design stages. However, few types of research can be found on knowledge representation of assembly process planning and the inference of assembly process planning information from geometrical product model. Therefore, ontologies that enable the description of more detailed information are drawing increased research attention. As one of the pillars for the semantic web, ontology enriches knowledge representation [21]. It enhances a comprehensive information model or knowledge model thereby reducing ambiguity [22]. The ontology also gains its application in the manufacturing domain to formalize domain concepts and processes. National Institute of Standards and Technology (NIST) proposed Process Specification Language (PSL), which is an interexchange format designed to help exchange process information automatically among a wide variety of manufacturing applications such as process modelling, process planning, scheduling, simulation, workflow, project management and business process re-engineering tools [23]. A STEP-based ontology named OntoSTEP was proposed to translate the product geometric information in STEP file into ontology [24]. Panetto et al. [25] proposed a product-driven ontology, ONTO-PDM, for product data management interoperability within the manufacturing environment. In engineering product design, Chang et al. [26] proposed a graphical modelling tool to support conceptual design. Kitamura et al. [27–29] also proposed an ontology-based framework that can represent product functional design, functional design knowledge and functional structure recognition. Furthermore, in the domain of process planning, Bock and Gruninger [30] showed how manufacturing knowledge can be expressed by the PSL ontology. Similarly, Cochrane et al. [31] proposed a PSL-based ontology to indicate process planning knowledge.

The paragraph above highlighted the importance of ontology-based reasoning in efficient decision making. However, at the same time, it is also important that the relationship between decision-making process and knowledge modelling should be accounted for. For the instance of knowledge modelling for assembly process planning, Holland et al. [32] established feature models for single-part and assemblies and discussed their application in assembly process planning. Zhao and Liu [33, 34] proposed an Ontology Web Language (OWL) representation methodology for an Express-driven product information model. The ontology concept was also employed on the disassembly decision making process using case based reasoning (CBR) by Chen et al. [35]. If the CBR failed, they utilized rule-based reasoning (RBR). Rachuri et al. [36] established the Open Assembly Model (OAM), which is an assembly model based on the NIST Core Product Model (NIST-CPM). Defined with an object-oriented representation, it covers assembly function, form and behaviour. With an open structure, it could be applied in a collaborative environment. Although the model is relatively complete for representing the geometric assembly information, some implicit geometric information underlying in the product model is not well considered. Ma et al. [37] also proposed a hierarchical graph based on the bill of material (BOM) for open architecture products (OAPs) for the cases of serial and parallel assembly. The authors utilized precedence constraint knowledge among components and modules to generate feasible assembly sequences. Gruhier et al. [38] introduced a formal ontology-based on spatiotemporal mereotopology in the context of integrated assembly design and sequence planning. Finally, Yu et al. [39] worked on assembly ontology for ASP of a ball valve assembly.

Since the above cited works lacked support for reasoning and inference mechanisms, some researchers from NIST proposed not only to transform the existing assembly models—CPM and OAM—from UML to ontology model, but to also develop reasoning techniques based on the generated ontology model [40]. Kim et al. [41] proposed the assembly relation model (ARM) to represent the assembly information based on spatial relations between parts and features, thereby developing a reasoning mechanism through ontology representation of the ARM [42]. Noh et al. [43] also worked on a framework for collaborative product engineering environments by combining product information model with a rule-based model using description frame logic. Samer et al. [44] enhanced the collaboration among designers by defining feature-based ontology model and Semantic Web Rule Language (SWRL). Zhu et al. [45] proposed an ontology reasoning mechanism to infer the implicit information in the product model and implemented a layered semantic application architecture for the reasoning unit to query and reason assembly information from CAD systems. Again, in the mentioned works in this paragraph, the description of the geometric information was not deep enough.

Therefore, in view of the over-arching aim of the research and the literature reviewed, it was inferred that there is a need for a reasoning mechanism that can infer assembly relations based on certain rules and guidelines to facilitate decision making in ASP. As knowledge integration is an integral part of such a scenario, geometry-enhanced ontology modelling and reasoning was considered in this paper as a framework to not only enrich knowledge representation but also provide an interaction between the product and process domains for the final ASP. The section to follow will explain the framework in detail.

3 Geometry-enhanced ontology modelling and reasoning framework for ASP

In order to enrich the semantics in ASP and to improve reasoning of the implicit knowledge underlying in the existing data, a geometry-enhanced ontology modelling and reasoning framework is proposed in this paper, which takes advantage of the geometric information and shape representation to better specify product assemblability. The framework is composed of two main parts:

1) An ontology model that depicts the terminology and data structure for ASP
2) A reasoning mechanism that infers the underlying relations among the existing data These contents will be discussed in the following sections.

As illustrated in Fig. 1, in the proposed framework, the information from product model is parsed, imported and stored as the ontology data, which contain many implicit relations between product structure, product geometry and the assembly sequence. Based on predefined rules and reasoning units established according to the assembly process knowledge, new relations will be inferred from the implicit relations to support efficient decision-making for ASP.

As the foundation of the modelling and reasoning framework for ASP, the ontology model contains the key concepts of assembly operations and the relations between them. These concepts and relations should be clarified and well organized within the ontology model in order to ensure the completeness. Ontology Web Language Description Logics (OWL-DL) is applied in this paper as the modelling language for building up the ontology model. As one of the most widely used ontology modelling languages proposed by World Wide Web Consortium (W3C), OWL-DL ensures the universality of the model. The directed graph is also utilized to graphically represent the OWL-DL model [46]. Figure 2 is an example of the directed graph.

As depicted in Fig. 2, the ontology structure is organized with class, property and individual. The concept is represented by a class, while the individual is an instance of the class. Property includes object property and data property. It finds relationships between two resources.

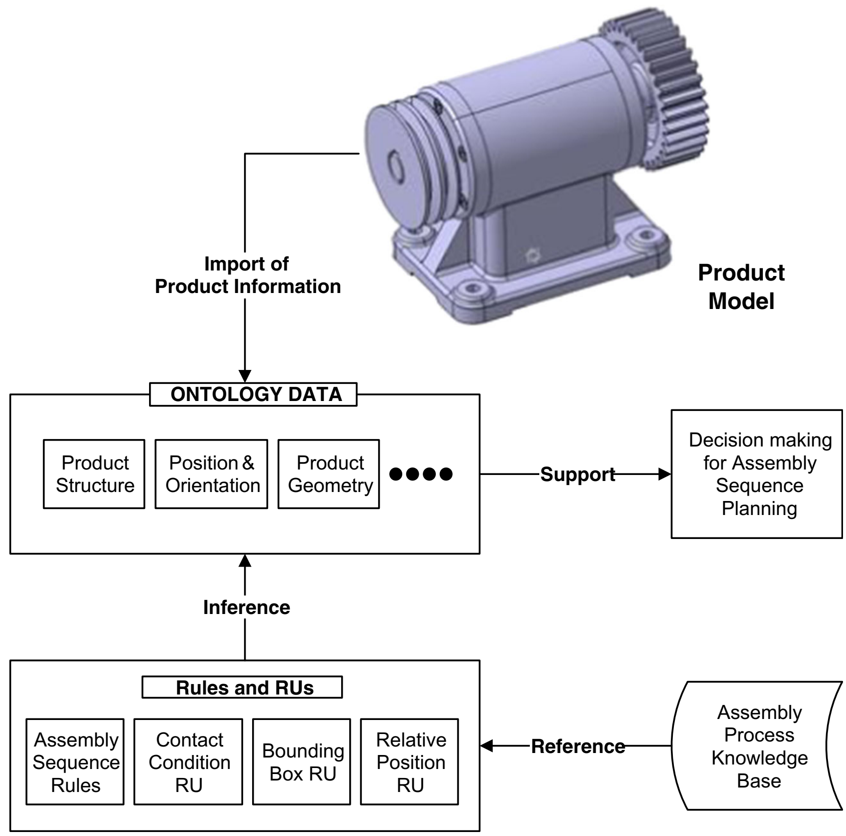

Fig. 1 Ontology-based modelling and reasoning framework

The object property links two individuals and the data property links an individual with a specific type of data.

What distinguishes the OWL-DL model from object oriented model is the ability to describe properties and their restrictions. Property descriptions describe properties in a global context while property restrictions describe properties within the context of a specific class.

Property description and property restriction together enrich the semantics within the relationships between concepts. In later stages, through the processing and reasoning of inference engines, some underlying relationships will be inferred to support decision-making.

3.1 OWL-DL representation of the ontology model for ASP

In order to represent the ontology model for ASP, some concepts should be clarified. Classes representing these concepts can be sorted as (a) assembly structure, (b) assembly process, (c) assembly position and orientation, and (d) assembly entity geometry, with each class having its subclasses, as shown in the class hierarchy in Fig. 3.

3.1.1 Assembly structure

Assembly structure conveys the structure information of the assembly from product design to assembly process planning. A product is generally composed of parts and components, while components can be further decomposed into parts or lower-level components. The relative position and orientation of parts and components are defined by geometric constraints. Table 1 illustrates the properties defined in the OWL-DL ontology model of assembly structure as well as the classes linked by them. According to the hierarchy of the assembly structure, the relations between classes like *Product*, *Part* and *Component* are established through properties like *isComposedOf* and *compose*. Constraints are defined between these classes by specifying a *hasConstraint* or *constrains* property. *isComposedOf* and *compose* are defined as inverse properties by imposing the property description *owl:inverseOf* between them, so it is with *hasConstraint* and *constrains.*

3.1.2 Assembly process

Assembly process conveys the information that connects assembly structure, assembly resources and other information together in a logical and time-sequenced way. It consists of a series of basic elements, namely assembly operations. In an assembly operation, a specific part or component is assembled using certain resources. Each assembly operation has logical relations (before, after) and time relations (start time, end time) with another.

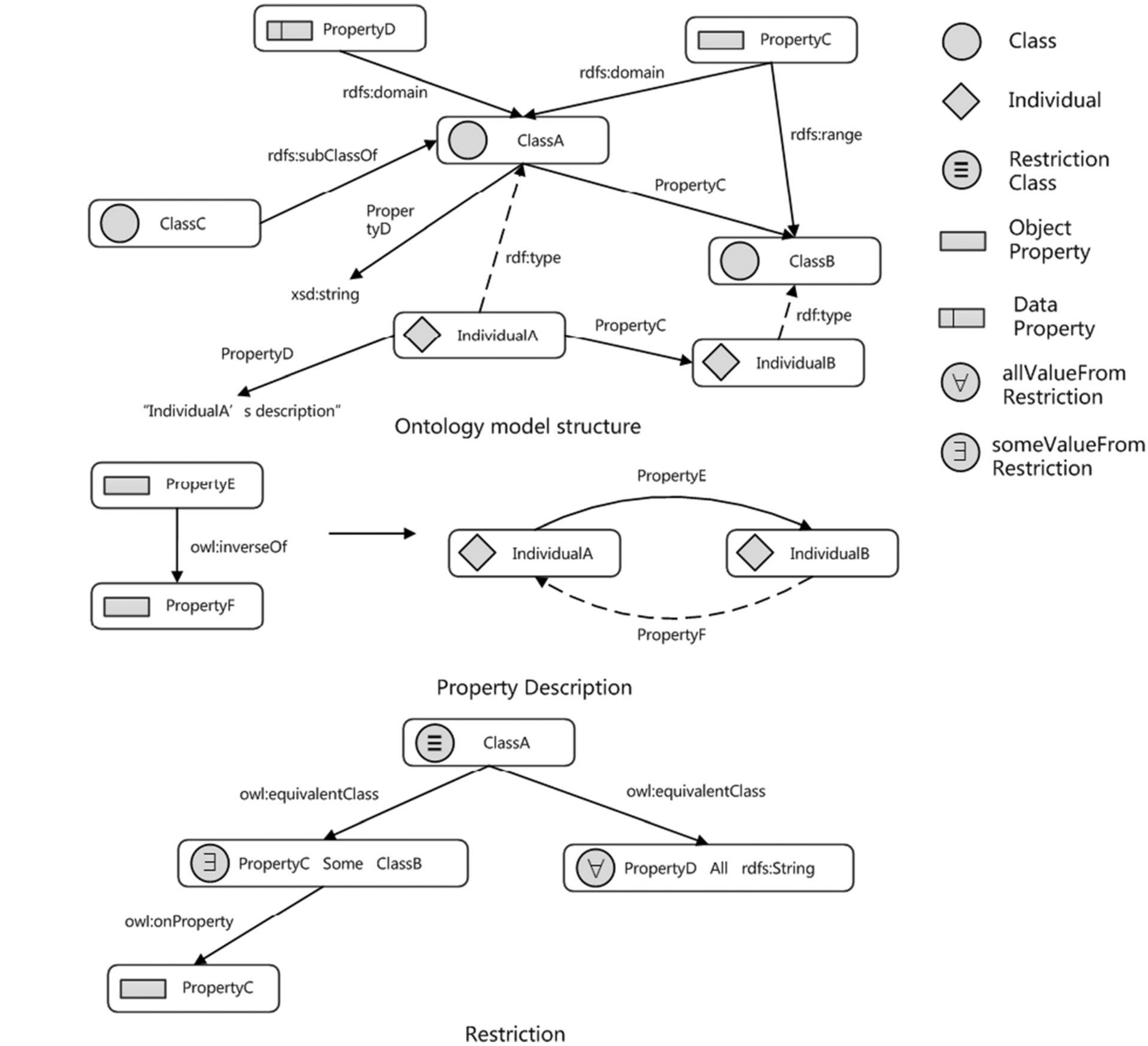

Fig. 2 An example of the directed graph representation for OWL-DL model

Fig. 3 Class hierarchy of the ontology model

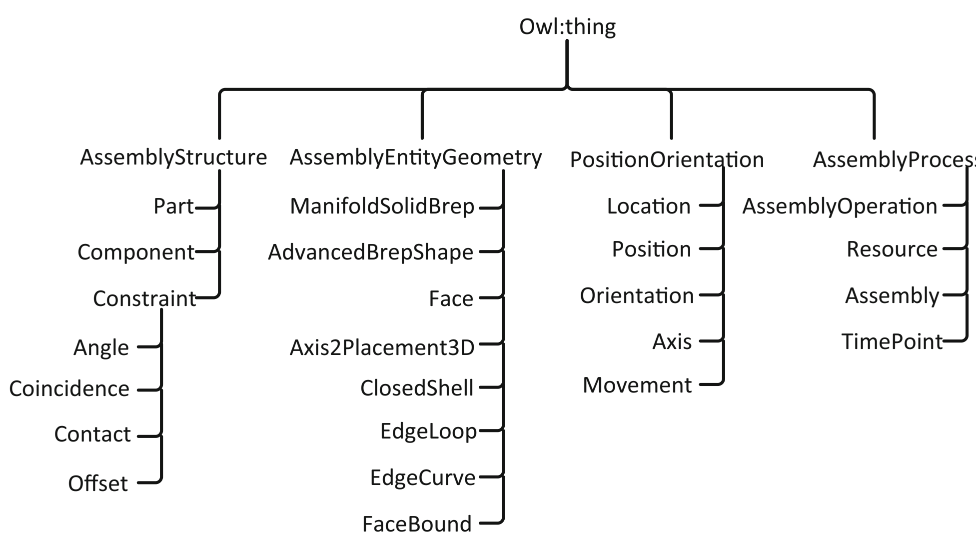

Table 1 Properties defined in assembly structure

Property	Domain	Range
isComposedOf[a]	Product	Part/Component
compose[a]	Part/Component	Product
hasConstraint[a]	AssemblyStructure	Constraint
constrains[a]	Constraint	AssemblyStructure

[a] Properties that are two pairs of inverse properties

Table 2 illustrates the properties defined in the OWL-DL ontology model of an assembly process, together with the classes linked by them. Figure 4 further represents the OWL-DL model of the assembly process using directed graph. The start and end time of an assembly operation are indicated by data properties, *beginAt* and *endAt*, which belong to a *dateTime* data type. The *assembleBefore* and *assembleAfter* are the two object properties that connect two assembly operations and define their sequence. The assembly structure to be assembled in each operation is defined with the *assembling* property, while the assembly resource to be used is defined with the *use* property. When assembled, the assembly structure is defined as *assembled* property in each operation. Property descriptions and restrictions are added to fully represent the relations between different assembly operations, as well as the corresponding assembly structure and assembly resource used in them.

3.1.3 Assembly position and orientation

Many aspects of assembly process planning, such as relative positioning of parts and assembly path planning, depend on the position and orientation of parts or components. In order to express the position and orientation of a rigid body geometrical entity, two coordinate systems (CS) are defined: an absolute CS in a 3D environment and a relative CS fixed on the entity. The coordinate of the relative CS origin in the absolute CS defines the position of the entity while its orientation is defined by the yaw, pitch and roll angles of the relative CS.

Hence, the position and orientation ontology in OWL-DL could be built, as shown in Table 3. All the 4 defined properties are used to describe assembly position and orientation. For example, 12 parameters (a_0~a_{11}) are used to define the location information of an assembly resource, in which a_9~a_{11} are the coordinates of the relative CS origin in the absolute CS while a_0~a_2, a_3~a_5 and a_6~a_8 are used as the vector coordinates of the X1, Y1 and Z1 axes in the relative CS, respectively. Therefore, there are 12 data properties (*hasA0~hasA11*) in the *Location* class, which is the range of the *hasLocation* property. *AssemblyStrcture* class establishes relation with certain *Location* class by

Table 2 Properties defined in assembly process

Property	Domain	Range
assembleBefore[a,b]	AssemblyOperation	AssemblyOperation
assembleAfter[a,b]	AssemblyOperation	AssemblyOperation
assembling[b]	AssemblyOperation	AssemblingStructure
assembled[b]	AssemblyOperation	AssembledStructure
beginAt	AssemblyOperation	xsd:dateTime
endAt	AssemblyOperation	xsd:dateTime
use[c]	AssemblyOperation	Resource

[a] Properties that are a pair of inverse properties

[b] Properties that have cardinality restriction with the cardinality as 1

[c] Properties that have someValuesFrom restriction

hasLocation property. Meanwhile, other three properties in the table are used to establish relation among different classes.

3.1.4 Assembly entity geometry

Assembly entity geometry contains detailed geometric information of the product, part or component. Based on STEP AP 203, assembly entity geometry is represented by some basic geometric elements such as closed shell, face bound, edge loop and so on. The OWL-DL model of assembly entity geometry can be indicated by the properties and classed as shown in Table 4. The property of assembly entity geometry is mainly composed of functional relation. For instance, an *AssemblyStrcture* consists of an *AdvancedBrepRepresentation*. The property *hasGeometry* is the functional relation that connects these two classes. The geometric information can be obtained from a STEP file, which is supported by most mainstream CAD software packages, and then organized as such in ontology models.

3.2 Ontology reasoning for ASP

Apart from the explicit relations established in the ontology model, there remain substantial implicit relations within the model, especially those underlying between the geometric elements. These relations could be inferred through a reasoning mechanism to support decision-making in assembly process planning. The ontology and assembly sequence reasoning mechanism will be discussed in the following sections.

3.2.1 The ontology reasoning mechanism

The ontology model based on OWL-DL is composed of a set of triples, with each triple represented as (x,R,y) or $R(x,y)$, in

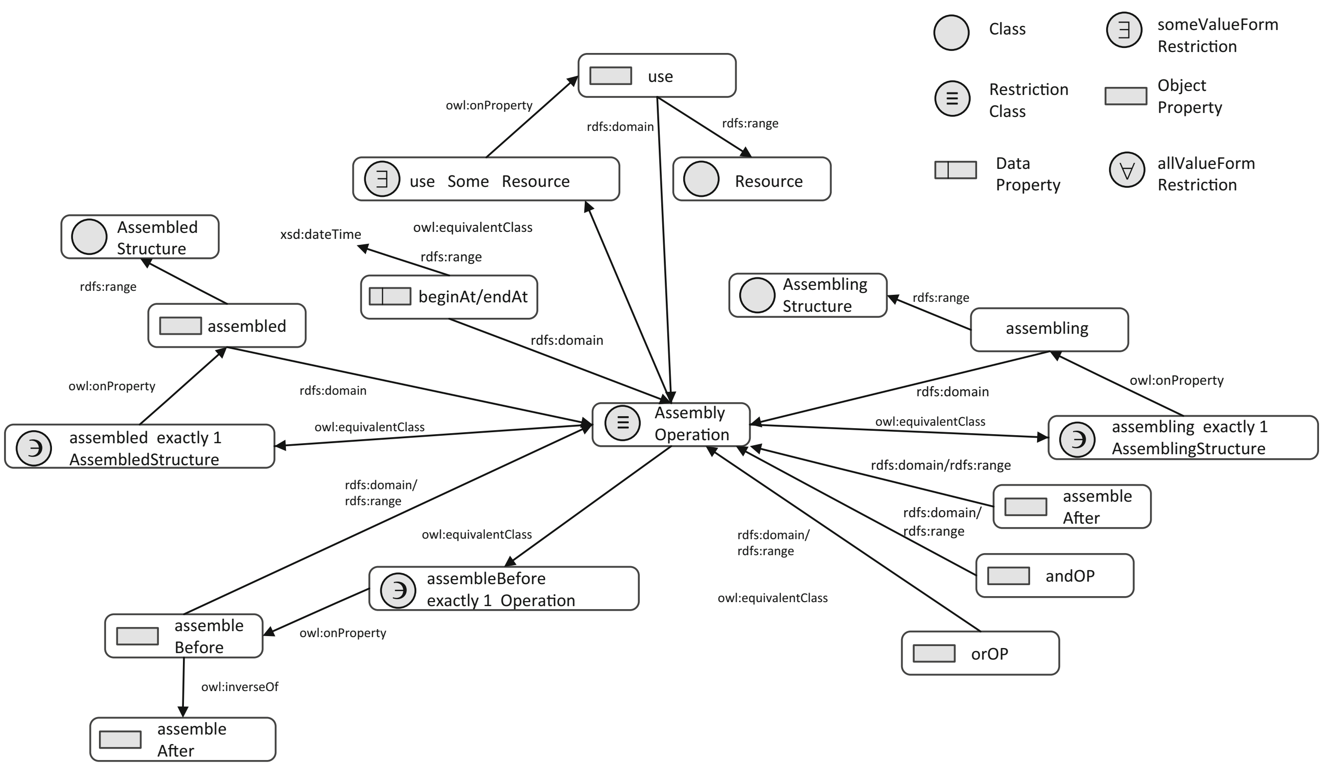

Fig. 4 The OWL-DL model of assembly process in directed graph representation

which x and y denote two individuals related to property R. Thus, the reasoning process of implicit relations can be described by the mathematical model as explained in Eq. 1.

$$
\begin{aligned}
&\textit{Objective}: \\
&Rt\ (x,y)\ ,\ x\ ,\ y\ \in\ \textit{OntIndividuals},\ \ Rt\ \in\ \textit{FactBase} \vee\ \textit{InferBase} \\
&S\ .\ T\ . \\
&P_{11}(x_{11},y_{11}) \wedge P_{12}(x_{12},y_{12}) \wedge \ldots \wedge P_{1n_1}(x_{1n_1},y_{1n_1}) \rightarrow \exists\, x_{r1} \exists\, y_{r1} R_1(x_{r1},y_{r1}) \\
&P_{21}(x_{21},y_{21}) \wedge P_{22}(x_{22},y_{22}) \wedge \ldots \wedge P_{2n_2}(x_{2n_2},y_{2n_2}) \rightarrow \exists\, x_{r2} \exists\, y_{r2} R_2(x_{r2},y_{r2}) \\
&\ldots \qquad \ldots \\
&P_{m1}(x_{m1},y_{m1}) \wedge P_{m2}(x_{m2},y_{m2}) \wedge \ldots \wedge P_{mn_m}(x_{1n_m},y_{1n_m}) \rightarrow \exists x_{rm} \exists y_{rm} R_m(x_{rm},y_{rm}) \\
&P_{ij} \in \qquad \textit{FactBase} \\
&R_i \in \qquad \textit{InferBase}
\end{aligned}
\tag{1}
$$

In Eq. 1, $Rt(x,y)$ is the outcome of reasoning. *OntIndividuals* represents the collection of all individuals. *FactBase* is a collection of all the explicit individuals and properties, while *InferBase* denotes the collection of properties that have been inferred. Each line in the equation represents a constraint, in which $P_{ij}(x_{ij},y_{ij})$ is the j_{th} premise in i_{th} constraint, and R_i is the inferred result of the i_{th} constraint. In each constraint, all premises shall be true to ensure the correct result of inference.

To elaborate, the reasoning mechanism can be depicted with the following steps:

Step 1: Instantiate all the explicit individuals and the properties between them in the existing information, since these individuals and properties compose the facts for reasoning, as denoted by *FactBase* in the mathematical model.

Step 2: Define the semantic web rules according to the assembly knowledge. The assembly knowledge can be expressed as constraint equations in Eq. 1, which serve as semantic web rules and can be interpreted with the Semantic Web Rule Language (SWRL). SWRL can be recognized by a rule-based reasoning engine such as Pellet [47]. Based on the rules defined in SWRL, the engine can infer the implicit relations between individuals, thus creating new triples to compose the *InferBase* as defined in Eq. 1. For instance, a rule in SWRL defines that if there exists a relation between two individuals A and B, and another relation between individuals B and C, a new triple will be formed inferring a relationship between A and C.

Step 3: Query the result by defining an objective function as shown in Eq. 1. Then, the Semantic Query-Enhanced Web Rule Language (SQWRL) could be utilized to query

Table 3 Properties defined in assembly position and orientation

Property	Domain	Range
hasLocation	AssemblyStructure	Location
hasAxis	Location	Axis
hasPosition[a]	Axis	Position
hasOrientation[a]	Axis	Orientation

[a] Properties that have cardinality restriction with the cardinality as 1

Table 4 The properties defined in assembly geometry entity

Property	Domain	Range
hasGeometry	AssemblyStructure	AdvancedBrepRepresentation
hasManifoldSolidBrep	AdvancedBrepRepresentation	ManifoldSolidBrep
hasClosedShell	ManifoldSolidBrep	ClosedShell
hasAdvancedFace	ClosedShell	AdvancedFace
hasFace	AdvancedFace	Face
hasFaceBound	AdvancedFace	FaceBound
hasEdgeLoop	FaceBound	EdgeLoop
hasCartesianPoint	VertexPoint	CartesianPoint
hasVector	Line	Vector
hasDirection	Vector	Direction

the triples that satisfy the objective function. Based on this ontology reasoning mechanism, this paper will discuss the reasoning strategies in the planning of assembly sequence.

3.2.2 Enhanced reasoning with RU

Traditional SWRL can represent simple rules and logic, while the reasoning of relations between geometric information of a product and its assembly process calls for complex computational and decision-making efforts. Therefore, in this paper, the concept of geometry-enhanced RU is proposed to enhance the knowledge representation and reasoning capabilities of the ontology-based mechanism.

RU is an extension of rules. As shown previously in Eq. 1, rules are defined based on the judgement of several premises. Each premise has input and output parameters, returning a Boolean value to signify whether the premise is true. Similar to the structure of a premise, an RU also has these parameters and returns a value. Taking advantage of the custom built-in mechanism of SWRL, RU encapsulates complex algorithms, equations and logical judgements through computer programs, which greatly reinforce the reasoning capabilities of SWRL and simplifies the representation. According to their functionalities, the geometry-enhanced RUs can be classified into complex algorithms (e.g. calculation of bounding boxes), complex logical judgements (e.g. identification of assembly constraints, in which a series of judgements are made according to the given conditions to decide whether certain constraints exist between two assembly components) and complex equations (e.g. equations for the calculation of relative position of assembly components, in which the distance between the bounding boxes of the components is calculated to decide their relative position).

The RU applied in this paper utilizes the geometric information extracted from STEP files as input parameters, while the computation and logical judgements with the built-in programs inferred results such as assembly constraints, bounding boxes, the relative position of parts and components that are output to support the decision-making in ASP.

3.2.3 Assembly sequence reasoning

Assembly sequence, as a fundamental information in assembly process planning, specifies the logical and time sequence of assembly operations. The design of assembly sequence should be carried out under comprehensive consideration of

Table 5 Part of the common rules for ASP

Sr.	Rules	Rule description in SWRL
1	If components exist in a product, components should be assembled before single parts.	Component (?x) ∧ AssemblyOperation (?y) ∧ assemble(?y, ?x) ∧ Part (?z) ∧ AssemblyOperation (?a) ∧ assemble (?a, ?z) → assembleBefore (?y, ?a)
2	Base part is assembled first.	Base (?x) ∧ AssemblyOperation (?y) ∧ assemble (?y, ?x) ∧ AssemblyOperation (?z)→ assembleBefore (?y, ?z)
3	The part having the most contact with other parts is the base part.	Assembly (?x) ∧ enhancedOnto:isBase (?x)→ Base(?x)
4	Internal parts are assembled before external parts (judging from the distance between the part bounding box and the product bounding box).	Assembly (?x) ∧ AssemblyOperation (?y) ∧ assemble (?y, ?x) ∧ enhancedOnto:hasDistance (?x, ?u) ∧ Assembly (?z) ∧ AssemblyOperation (?a) ∧ assemble(?a, ?z) ∧ enhancedOnto:hasDistance (?z, ?v) ∧ greaterThan (?u, ?v) → assembleBefore (?y, ?a)

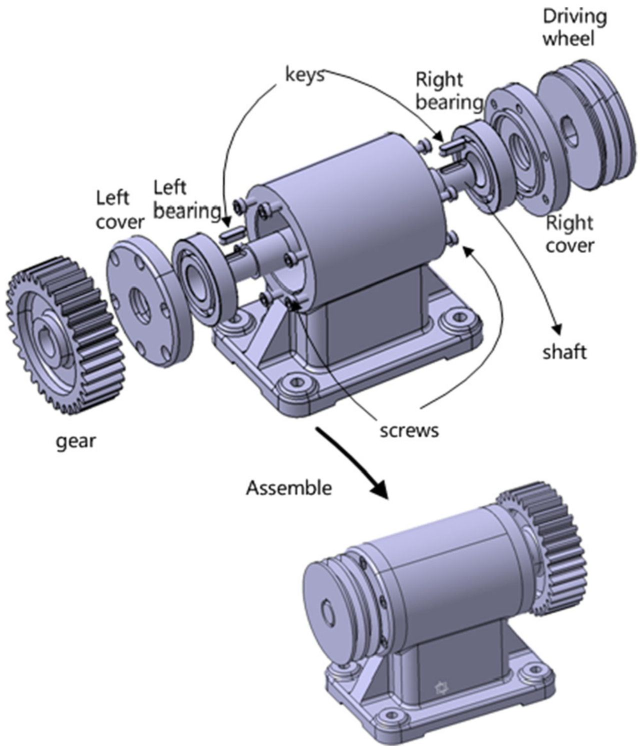

Fig. 5 Structure of the transmission

assembly structure, assembly geometry and relative position of assembly components.

For the proposed framework, the inference engine infers all the implicit relations between individuals according to the rules and existing information. Some of the rules are listed in Table 5. Variables are indicated using the standard convention of prefixing them with a question mark (e.g. ?*x*), while both antecedent and consequent of the rules are conjunction of atoms written as $a_1 \wedge \ldots \wedge a_2$.

Thereafter, an objective function defined by SQWRL queries the inferred sequence between assembly operations. Since the sequence relations between *AssemblyOperation* individuals are identified according to their *assemblebefore* properties, the individual that owns the largest number of *assemblebefore* properties should be assembled first. Thus, the sequence can be determined by sorting the individuals in a descending order of the number of *assemblebefore* properties that they have. Accordingly, the objective function can be written in SQWRL as follows:

$$AssemblyOperation(?x) \wedge assembleBefore\ (?x, ?y) sqwrl\ : select(?x) \wedge sqwrl : count(?y) \quad (2)$$

By executing Eq. 2, a list of assembly operation instances will be returned showing all the *AssemblyOperation* individuals and the number of *assembleBefore* properties that each individual owns in descending order. Thus, the assembly sequence could be identified from the list.

4 Case study

The ASP of a transmission is shown as an example to illustrate how the geometry-enhanced ontology modelling and reasoning framework works. The structure of the transmission is shown in Fig. 5.

First, the existing data, including the product structure and geometry information extracted from the STEP files, are imported and stored in the ontology model as the facts for reasoning. Thereafter, rules are defined according to the available knowledge and related RUs are developed for the processing of product geometry information. The geometry information of product is transferred into information that is directly related to assembly sequence planning reasoning, including part contact type, assembly constraints, geometric features, bounding boxes and/or relative position of assembly components by RU-enhanced reasoning. In

Table 6 Part of the case study-specific rules for ASP

Sr.	Implicit relation description	Rule description in SWRL
1	The base part needs to be assembled first.	Base (?x) ∧ AssemblyOperation (?y) ∧ assemble (?y, ?x) ∧ AssemblyOperation (?z) → assembleBefore (?y, ?z)
2	For the screw fastening, assemble the part to be fasten first, then assemble the bolt.	Screw (?x) ∧ AssemblyOperation (?y) ∧ assemble (?y, ?x) ∧ Component (?z) ∧ hasPart (?z, ?x) ∧ hasPart (?z, ?a) ∧ hasPart (?z, ?b) ∧ AssemblyOperation (?c) ∧ assemble (?c, ?a) ∧ AssemblyOperation (?d) ∧ assemble (?d, ?b) → assembleBefore (?y, ?c) ∧ assembleBefore (?y, ?d)
3	For the key connection, assemble key into the key hole first, then assemble the other part.	Slot (?x) ∧ AssemblyOperation (?y) ∧ assemble (?y, ?x) ∧ Component (?z) ∧ hasPart (?z, ?x) ∧ hasPart (?z, ?a) ∧ hasPart (?z, ?b) ∧ AssemblyOperation (?c) ∧ assemble (?c, ?a) ∧ AssemblyOperation (?d) ∧ assemble (?d, ?b) → assembleBefore (?y, ?c) ∧ assembleBefore (?y, ?d)

addition to the common rules illustrated in Table 5, some specific rules applied in this case study are defined in Table 6. Based on the existing data, rules and reasoning units, the inference engine Pellet is used to infer the implicit sequence relations between assembly operations and the inferred the results are queried through SQWRL.

The inferred assembly sequence of the product is as follows:

Shaft→right and *left bearing→right cover→screws* for *the right cover*→key for *the driving wheel→driving wheel →left cover→screws* for *the left cover*→key for *the gear→gear*

The sequence is then validated by an assembly process simulation software—DELMIA. The simulation result shows that the inferred sequence is acceptable without causing any clash or interference during the assembly process, thus proving the effectiveness of the proposed framework.

5 Conclusion

In this paper, a geometry-enhanced ontology modelling and reasoning framework is proposed for ASP, which includes an ontology model for assembly operation and an ontology reasoning mechanism for the inference of assembly sequence based on established rules and RUs. The ontology model is highly flexible and customized, due to its extendibility. Main concepts and relations in the domain of product modelling and assembly process planning, such as product structure and assembly process, are described in this framework. Information, such as assembly entity geometry, assembly position and orientation information, is also included in this model, which enhances its ability for geometric representation.

An ontology reasoning mechanism is further proposed using SWRL as a rule description language to make the implicit knowledge explicit. The RUs are also incorporated in this mechanism to enhance its ability in the processing of geometric information. Thus, the implicit relations underlying in the geometric information can be inferred from the ontology data to support automatic decision-making in ASP. A case study is provided to implement the models and methods proposed in this framework in the ASP of a transmission. The result proves the validity of the proposed framework. Moreover, the proposed framework overcomes some inherent limitations in existing models by taking into consideration of geometric information. However, more efforts are needed to complete the representation of assembly process and to incorporate more practical knowledge for ASP.

Acknowledgements The authors would like to express their appreciation to the respective agencies.

Funding information This work is supported by the National Natural Science Foundation of China (Grant 51575031), the National Hi-Tech R&D Program (863 Program) (Grant 2015AA043702) and the Graduate Student Innovation Fund of Beihang University.

References

1. Samy SN, ElMaraghy H (2012) A model for measuring complexity of automated and hybrid assembly systems. Int J Adv Manuf Technol 62(5):813–833
2. Krause FL, Kimura F, Kjellberg T, Lu SC-Y, Wolf VD, Alting L, ElMaraghy HA, Eversheim W, Iwata K, Suh NP, Tipnis VA, Weck M (1993) Product modelling. CIRP Ann Manuf Technol 42(2): 695–706
3. Hadj RB, Trigui M, Aifaoui N (2014) Toward an integrated CAD assembly sequence planning solution. J Mech Eng Sci 0(0):1–15
4. Perrard C, Bonjour E (2012) A priori checking inconsistencies among strategic constraints for assembly plan generation. Int J Adv Manuf Technol 63(5):817–838
5. Xu X, Wang L, Newman ST (2011) Computer-aided process planning—a critical review of recent developments and future trends. Int J Comput Integr Manuf 24(1):1–31
6. Huang Z, Qiao LH, Anwer N, Mo Y (2014) Ontology model for assembly process planning knowledge. Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014, Zhuhai, China, 419–423
7. Alexander CY (2006) Methods in biomedical ontology. J Biomed Inform 39(3):252–266
8. Morbach J, Wiesner A, Marquardt W (2009) OntoCAPE-A (re)usable ontology for computer-aided process engineering. Comput Chem Eng 33(10):1546–1556
9. Valaski J, Malucelli A, Reinehr S (2012) Ontologies application in organizational learning: a literature review. Expert Syst Appl 39(8): 7555–7561
10. Huang N, Diao S (2008) Ontology-based enterprise knowledge integration. Robot Comput Integr Manuf 24(4):562–571
11. Agustina B, Cechich A, Fillottrani P (2009) Ontology-driven geographic information integration: a survey of current approaches. Comput Geosci 35(4):710–723
12. De Almeida Biolchini JC, Mian PG, Natali ACC, Conte TU, Travassos GH (2007) Scientific research ontology to support systematic review in software engineering. Adv Eng Inform 21(2): 133–151
13. De Fazio TL, Whitney DE (1987) Simplified generation of all mechanical assembly sequences. IEEE J Robot Autom 3(6):640–658

14. Homem de Mello LS, Sanderson AC (1991) A correct and complete algorithm for the generation of mechanical assembly sequences. IEEE Trans Robot Autom 7(2):228–240
15. Bourjault A (1984) Contributionune approche methodologique de l'assemblage automatise: Elaboration automatique dessequences operatiores. Thesis. Besancon (France): d'Etat Universite de Franche-Comte
16. Pintzos G, Triantafyllou C, Papakostas N, Mourtzis D, Chryssolouris G (2016) Assembly precedence diagram generation through assembly tiers determination. Int J Comput Integr Manuf. https://doi.org/10.1080/0951192X.2015.1130260
17. Pandremenos J, Chryssolouris G (2011) A neural network approach for the development of modular product architectures. Int J Comput Integr Manuf 24(10):879–887
18. Wang Y, Tian D (2015) A weighted assembly precedence graph for assembly sequence planning. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-015-7565-5
19. Zha XF, Samuel YE, Fok SC (1998) Integrated knowledge-based assembly sequence planning. Int J Adv Manuf Technol 14(1):50–64
20. Tao S, Hu M (2017) A contact relation analysis approach to assembly sequence planning for assembly models. Comput Aided Des Appl. https://doi.org/10.1080/16864360.2017.1287674
21. Liu Y, Lim SCJ (2011) Using ontology for design information and knowledge management: a critical review. In: Bernard A (ed) Global product development. Springer, Berlin Heidelberg, pp 427–433
22. Lee JH, Suh HW (2005) OWL-based hybrid product knowledge model for collaborative engineering environment. In proceedings of ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 3:877–885
23. Schlenoff C, Gruninger M, Tissot F, Valois J, Road TC, Steptools Inc, Lubell J, Lee J (2000) The process specification language (PSL) overview and version 1.0 specification. NISTIR 6459. Gaithersburg, MD., National Institute of Standards and Technology
24. Raphael B, Krima S, Fiorentini X, Rachuri S, Narayanan A, Foufou S, Sriram RD (2012) OntoSTEP: enriching product model data using ontologies. Comput Aided Des 44(6):575–590
25. Panetto H, Dassisti M, Tursi A (2012) ONTO-PDM: product-driven ONTOlogy for product data management interoperability within manufacturing process environment. Adv Eng Inform 26(2):334–348
26. Chang X, Sahin A, Terpenny J (2008) An ontology-based support for product conceptual design. Robot Comput Integr Manuf 24(6): 755–762
27. Kitamura Y, Sano T, Namba K, Mizoguchi R (2002) A functional concept ontology and its application to automatic identification of functional structures. Adv Eng Inform 16(2):145–163
28. Kitamura Y, Kashiwase M, Fuse M, Mizoguchi R (2004) Deployment of an ontological framework of functional design knowledge. Adv Eng Inform 18(2):115–127
29. Kitamura Y, Mizoguchi R (2004) Ontology-based systematization of functional knowledge. J Eng Des 15(4):327–351
30. Bock C, Gruninger M (2004) Inputs and outputs in the process specification language. NISTIR 7152, NIST, Gaithersburg, MD. Web Accessed 30th January 2017: http://www.nist.gov/msidlibrary/doc/nistir7152.pdf
31. Cochrane S, Young R, Case K, Harding J, Gao J, Dani S, Baxter D (2009) Manufacturing knowledge verification in design support systems. Int J Prod Res 47(12):3179–3204
32. Holland WV, Bronsvoort WF (2000) Assembly features in modeling and planning. Robot Comput Integr Manuf 16(4):277–294
33. Zhao W, Liu JK (2008a) OWL/SWRL representation methodology for EXPRESS-driven product information model: part I: implementation methodology. Comput Ind 59(6):580–589
34. Zhao W, Liu JK (2008b) OWL/SWRL representation methodology for EXPRESS-driven product information model: part II: practice. Comput Ind 59(6):590–600
35. Chen S, Yi J, Jiang H, Zhu X (2016) Ontology and CBR based automated decision-making method for the disassembly of mechanical products. Adv Eng Inform 30:564–584
36. Rachuri S, Han YH, Foufou S, Feng SC, Roy U, Wang F, Sriram RD, Lyons KW (2005) A model for capturing product assembly information. J Comput Inf Sci Eng 6(1):11–21
37. Ma H, Peng Q, Zhang J, Gu P (2016) Assembly sequence planning for open-architecture products. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-017-0160-1
38. Gruhier E, Demoly F, Dutartre O, Abboudi S, Gomes S (2015) A formal ontology-based spatiotemporal mereotopology for integrated product design and assembly sequence planning. J Adv Eng Informat 29(3):495–512
39. Yu M, Tianlong GU, Liang C, Fengying L (2016) Assembly ontology for assembly sequence planning. Int J Pattern Recognit Artif Intell 29(3):203–215
40. Fiorentini X, Gambino I, Liang VC, Rachuri S, Mani M, Bock C (2007) An ontology for assembly representation. NIST Interagency/Internal Report (NISTIR)—7436. U.S. Department of Commerce. http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=822740. Accessed 30th January 2017
41. Kim KY, Wang Y, Muogboh OS, Nnaji BO (2004) Design formalism for collaborative assembly design. Comput Aided Des 36(9): 849–871
42. Kim KY, Manley DG, Yang H (2006) Ontology-based assembly design and information sharing for collaborative product development. Comput Aided Des 38(12):1233–1250
43. Noh JD, Suh HW, Lee H (2009) Hybrid knowledge representation and reasoning with ontology and rules for product engineering. In Proceedings of ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 2: 409–418
44. Samer AG, Ghodous P, Shariat B, Perna E (2008) Towards an intelligent CAD models sharing based on semantic web technologies. In: Collaborative Product and Service Life Cycle Management for a Sustainable World, Advanced Concurrent Engineering, Springer London, pp 195–203
45. Zhu L, Jayaram U, Kim O (2011) Semantic applications enabling reasoning in product assembly ontologies-moving past modeling. J Comput Inf Sci Eng. https://doi.org/10.1115/1.3647878
46. Allemang D, Hendler J (2008) Semantic web for the working ontologist: effective modeling in RDFS and OWL. 1st edition, Morgan Kaufmann, ISBN-13: 978–0123735560
47. Sirin E, Parsia B, Grau BC, Kalyanpur A, Kats Y (2007) Pellet: a practical OWL-DL reasoner. Web Semant Sci Serv Agents World Wide Web 5(2):51–53