HAL
open science

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

Lihong Qiao, Yifan Qie, Zuowei Zhu, Yixin Zhu, Uzair Khaleeq Uz Zaman, Nabil Anwer

## To cite this version:

Lihong Qiao, Yifan Qie, Zuowei Zhu, Yixin Zhu, Uzair Khaleeq Uz Zaman, et al.. An ontology-based modelling and reasoning framework for assembly sequence planning. International Journal of Advanced Manufacturing Technology, 2017, 94 (9-12), pp.4187 - 4197. 10.1007/s00170-017-1077-4 . hal-01653820

**HAL Id: hal-01653820**

**https://hal.science/hal-01653820v1**

Submitted on 1 Dec 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# 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]

*1. School of Manufacturing 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*

Corresponding author: Yixin Zhu, e-mail: chunic@buaa.edu.cn

# An Ontology-based Modelling and Reasoning Framework for Assembly Sequence Planning

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. In order 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 case study related to assembly sequence planning.

Keywords: assembly sequence planning; geometry; ontology; reasoning

## 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. Assembly refers to a product which comprises a set of components and their relationships, or the process the product is created. 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 takes into account available assembly resources to improve 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 to 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 exist when it comes to the knowledge sharing between the product model and assembly process planning. 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, posi-

tion, contact and mobility of parts are main factors that must be considered in an assembly process. ASP needs to incorporate such information to enrich the semantics in model representation and enhance the decision-making capabilities. Moreover, ontology modelling, as one of the commonly used modelling methods in data management, can express unified, structural and semantic information and it 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: Section 2 highlights the research background of the ontology–based modelling, representation and reasoning framework in the context of ASP; Section 3 explains the proposed ontology modelling and reasoning framework for ASP along with the discussion of construction of rules and RUs; a case study is given in Section 4 to validate the framework and the conclusion is drawn in Section 5.

## 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, thus has been applied in many domains such as medical information science, geographic information system, enterprise modelling, organization learning, and software engineering [7, 8, 9, 10, 11, 12].

The traditional approaches for assembly modelling and ASP mainly focused on the relations among parts of a product. The concept of assembly precedence relations (APRs) was proposed to express the precedence relations among parts in an assembly by answering a series of structured questions [13]. Graph-based methodologies, such as liaison sequence graph (LSG) and AND/OR graph [14, 15], have also been proposed to deduce feasible assembly sequences in a graphical way. In order to integrate knowledge in the decision-making of ASP, Zha et al. [16] proposed a novel approach for the automatic generation, selection and evaluation of assembly plans, but the approach was inefficient due to limitations in modelling techniques. It is therefore highly imperative that knowledge is integrated in ASP via an efficient reasoning mechanism to better support decision-making in later design stages.

However, few researches 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 de-

tailed information are drawing increased research attention. As one of the pillars for semantic web, ontology enriches knowledge representation [17]. It enhances a comprehensive information model or knowledge model thereby reducing ambiguity [18]. Ontology also gains its application in the manufacturing domain to formalize domain concepts and processes. NIST (National Institute of Standards and Technology) proposed PSL (Process Specification Language), 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 [19]. A STEP-based ontology named OntoSTEP was proposed to translate the product geometric information in STEP file into ontology [20]. Panetto et al. [21] proposed a product-driven ontology, ONTO-PDM, for product data management interoperability within manufacturing environment. In engineering product design, Chang et al. [22] proposed a graphical modelling tool to support conceptual design. Kitamura et al. [23, 24, 25] proposed an ontology-based framework that can represent product functional design, functional design knowledge and functional structure recognition. While in the domain of process planning, Bock and Gruninger [26] showed how manufacturing knowledge can be expressed by the PSL ontology. Cochrane et al. [27] proposed a PSL-based ontology to indicate process planning knowledge.

As the decision-making process is highly related to knowledge modelling, there have been many researches on the knowledge modelling for assembly process planning. Holland et al. [28] established feature models for single-part and assemblies and discussed their application in assembly process planning. Zhao and Liu [29, 30] proposed an Ontology Web Language (OWL) representation methodology for an Express-driven product information model. Rachuri et al. [31] 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. Gruhier et al. [32] introduced a formal ontology-based on spatiotemporal mereotopology in the context of integrated assembly design and sequence planning. Yu et al. [33] also worked on assembly ontology for ASP of a ball valve assembly.

An obvious limitation of the above mentioned researches is their lack of support for reasoning and inference mechanisms. Therefore, some researchers from NIST proposed to transform existing assembly model- CPM and OAM, from UML model to ontology model, and to develop reasoning techniques based on the ontology model [34]. Kim et al. [35] proposed the Assembly Relation Model (ARM) to represent the assembly information based on spatial relations between parts and features, thus a reasoning mechanism is developed through ontology representation of the ARM [36]. Noh et al. [37] proposed a framework for collaborative product engineering environments by

combining a product information model with a rule-based model using Description Frame Logic. Samer et al. [38] enhanced the collaboration among designers by defining feature-based ontology model and Semantic Web Rule Language (SWRL). Zhu et al. [39] 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. However, the description of the geometric information is not deep enough in the reviewed works.

## 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 Figure 1, in the proposed framework, the information from product model is parsed, imported and stored as the ontology data, which contains 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.

{Please insert Fig. 1 about here}

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. 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 [40]. Figure 2 is an example of the directed graph.

{Please insert Fig. 2 about here}

As depicted in Figure 2, the ontology structure is organized with class, property and individual. Concept is represented by class, while individual is an instance of a class. Property includes object property and data property. It founds relationships be-

tween two resources.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 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 Figure 3.

{Please insert Fig. 3 about here}

#### *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.*

{Please insert Tab. 1 about here}

#### *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 assem-

bly operation has logical relations (before, after) and time relations (start time, end time) with another.

Table 2 illustrates the properties defined in the OWL-DL ontology model of assembly process, together with the classes linked by them. Figure 4 further represents the OWL-DL model of assembly process using directed graph. The start and end time of an assembly operation is indicated by data properties- *beginAt* and *endAt* that belong to a *dateTime* data type. *Before* and *after*, as two object properties that connect two assembly operations, define the sequence of operations. 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. 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.

{Please insert Tab. 2 about here}
{Please insert Fig. 4 about here}

### *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 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.

{Please insert Tab. 3 about here}

### *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 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.

{Please insert Tab. 4 about here}

### *3.2. Ontology Reasoning for ASP*

Apart from the explicit relations established in the ontology model, there remains 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 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 Equation 1.

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

In Eq. 1, *Rt(x,y)* is the outcome of reasoning. *OntIndividuals* represent the collection of all individuals. *FactBase* is a collection of all the explicit individuals and properties, while *InferBase* denotes the collection of properties that has been inferred. Each line in the equation represents a constraint, in which *Pij(xij,yij)* is the *jth* premise in *ith* constraint, and *Ri* is the inferred result of the *ith* 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 [41]. 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 exist 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.

Step3: 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 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 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 judgment 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 judgments through computer programs, which greatly reinforces 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 judgments (e.g. identification of assembly constraints, in which a series of judgments 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 are 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 judgment with the built-in programs, inferred results such as assembly constraints, bounding boxes, relative position of parts and components 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 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. 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 *before* properties, the individual that owns the largest number of *before* properties should be assembled first. Thus, the sequence can be determined by sorting the individuals in a descending order of the number of *before* properties that they have. Accordingly, the objective function can be written in SQWRL as:

$$\text{AssemblyOperation}(?x) \wedge \text{before}(?x,?y) \rightarrow \text{sqwrl:select}(?x) \wedge \text{sqwrl:count}(?y) \qquad (2)$$

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

{Please insert Tab. 5 about here}

## 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 Figure 5.

{Please insert Fig. 5 about here}

First, the existing data, including the product structure and geometry information extracted from the STEP files are imported and stored into 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. In 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.

{Please insert Tab. 6 about here}

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.

## Acknowledgement

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. The authors would like to express their appreciation to the agency.

## References

[1] Samy SN, ElMaraghy H (2012) A Model for Measuring Complexity of Automated and Hybrid Assembly Systems. The International Journal of Advanced Manufacturing Technology 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 Annals - Manufacturing Technology 42 (2): 695–706

[3] Hadj RB, Trigui M, Aifaoui N (2014) Toward an Integrated CAD Assembly Sequence Planning Solution. Journal of Mechanical Engineering Science 0(0): 1-15

[4] Perrard C, Bonjour E (2012) A Priori Checking Inconsistencies Among Strategic Constraints for Assembly Plan Generation. The International Journal of Advanced Manufacturing Technology 63 (5): 817–838

[5] Xu X, Wang L, Newman S T (2011) Computer-aided Process Planning – A Critical Review of Recent Developments and Future Trends. International Journal of Computer Integrated Manufacturing 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. Journal of Biomedical Informatics 39 (3): 252–266

[8] Morbach J, Wiesner A, Marquardt W (2009) OntoCAPE—A (re)usable Ontology for Computer-aided Process Engineering. Computers & Chemical Engineering 33 (10): 1546–1556

[9] Valaski J, Malucelli A, Reinehr S (2012) Ontologies Application in Organizational Learning: A Literature Review. Expert Systems with Applications 39 (8): 7555–7561

[10] Huang N, Diao S (2008) Ontology-based Enterprise Knowledge Integration. Robotics and Computer-Integrated Manufacturing 24 (4): 562–571

[11] Agustina B, Cechich A, Fillottrani P (2009) Ontology-driven Geographic Information Integration: A Survey of Current Approaches. Computers & Geosciences 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. Advanced Engineering Informatics 21 (2): 133–151

[13] Bourjault A (1984) Contributionune approche methodologique de l'assemblage automatise: Elaboration automatique dessequences operatiores. Thesis. Besancon (France): d'Etat Universite de Franche-Comte.

[14] De Fazio TL, Whitney DE (1987) Simplified generation of all mechanical assembly sequences. IEEE Journal on Robotics and Automation 3(6): 640-658

[15] Homem de Mello LS, Sanderson AC 1991. A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences. IEEE Transactions on Robotics and Automation 7(2): 228-240

[16] Zha XF, Samuel YE, Fok SC (1998) Integrated Knowledge-based Assembly Sequence Planning. International Journal of Advanced Manufacturing Technology 14(1): 50-64

[17] Liu Y, Lim SCJ (2011) Using Ontology for Design Information and Knowledge Management: A Critical Review. In Global Product Development, edited by Alain Bernard, Springer Berlin Heidelberg: 427–433

[18] 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

[19] 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

[20] Raphael B, Krima S, Fiorentini X, Rachuri S, Narayanan A, Foufou S, Sriram RD (2012) OntoSTEP: Enriching Product Model Data Using Ontologies. Computer-Aided Design 44 (6): 575–590

[21] Panetto H, Dassisti M, Tursi A (2012) ONTO-PDM: Product-driven ONTOlogy for Product Data Management Interoperability within Manufacturing Process Environment. Advanced Engineering Informatics 26 (2): 334–348

[22] Chang X, Sahin A, Terpenny J (2008) An Ontology-based Support for Product Conceptual Design. Robotics and Computer-Integrated Manufacturing 24 (6): 755–762

[23] Kitamura Y, Sano T, Namba K, Mizoguchi R (2002) A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures. Advanced Engineering Informatics 16 (2): 145–163

[24] Kitamura Y, Kashiwase M, Fuse M, Mizoguchi R (2004) Deployment of an Ontological Framework of Functional Design Knowledge. Advanced Engineering Informatics 18 (2): 115–127

[25] Kitamura Y, Mizoguchi R (2004) "Ontology-based Systematization of Functional Knowledge. Journal of Engineering Design 15 (4): 327–351

[26] 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

[27] Cochrane S, Young R, Case K, Harding J, Gao J, Dani S, Baxter D (2009) Manufacturing Knowledge Verification in Design Support Systems. International Journal of Production Research 47 (12): 3179–3204

[28] Holland WV, Bronsvoort WF (2000) Assembly Features in Modeling and Planning. Robotics and Computer-Integrated Manufacturing 16 (4): 277–294

[29] Zhao W, Liu JK (2008a) OWL/SWRL Representation Methodology for EXPRESS-driven Product Information Model: Part I. Implementation Methodology. Computers in Industry 59 (6): 580–589

[30] Zhao W, Liu JK (2008b) OWL/SWRL Representation Methodology for EXPRESS-driven Product Information Model: Part II: Practice. Computers in Industry 59 (6): 590–600

[31] Rachuri S, Han YH, Foufou S, Feng SC, Roy U, Wang F, Sriram RD, Lyons KW (2005) A Model for Capturing Product Assembly Information. Journal of Computing and Information Science in Engineering 6 (1): 11–21

[32] 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. Journal of Advanced Engineering Informatics 29(3): 495-512

[33] Yu M, Tianlong GU, Liang C, Fengying L (2016) Assembly Ontology for Assembly Sequence Planning. Pattern Recognition and Artificial Intelligence 29(3): 203-215

[34] 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, Accessed 30th January 2017: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=822740

[35] Kim KY, Wang Y, Muogboh OS, Nnaji BO (2004) Design Formalism for Collaborative Assembly Design. Computer-Aided Design 36 (9): 849–871

[36] Kim KY, Manley DG, Yang H (2006) Ontology-based Assembly Design and Information Sharing for Collaborative Product Development. Computer-Aided Design 38 (12): 1233–1250

[37] 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

[38] Samer AG, Ghodous P, Shariat B, and 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: 195-203

[39] Zhu L, Jayaram U, Kim O (2011) Semantic Applications Enabling Reasoning in Product Assembly Ontologies-Moving Past Modeling. Journal of Computing and Information Science in Engineering, doi: 10.1115/1.3647878

[40] 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

[41] Sirin E, Parsia B, Grau B C, Kalyanpur A, Kats Y (2007) Pellet: A practical OWL-DL reasoner. Web Semantics Science Services & Agents on the World Wide Web 5(2):51-53.

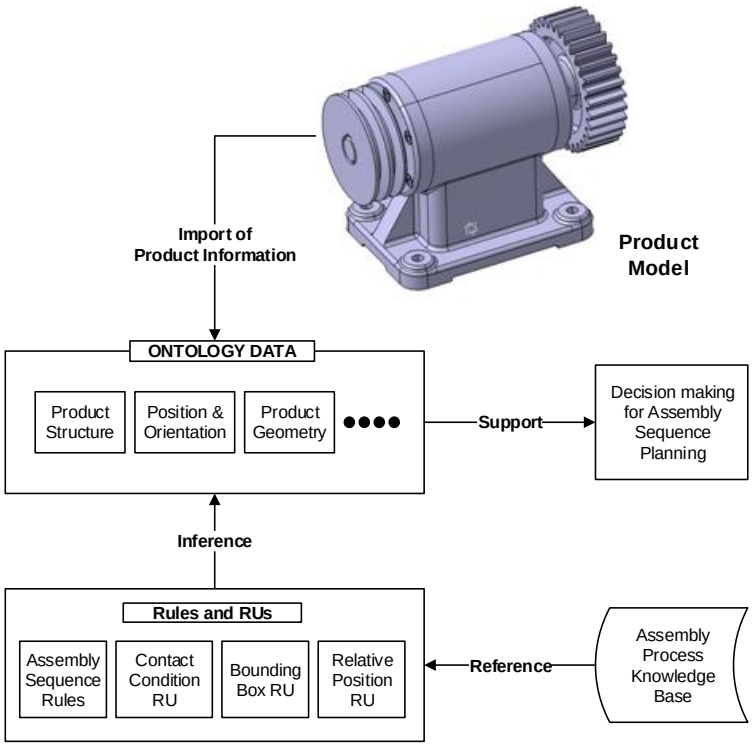


Figure 1. Ontology-based modelling and reasoning framework

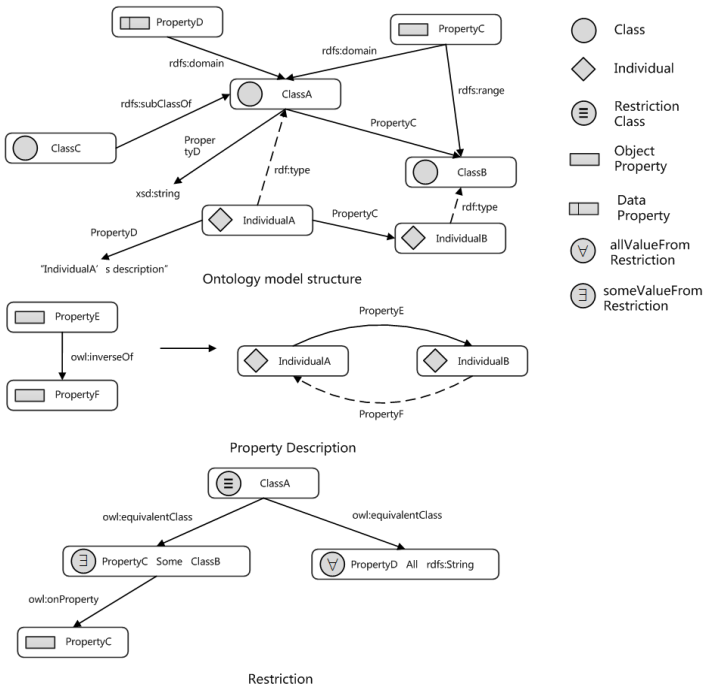


Figure 2. An example of the directed graph representation for OWL-DL model

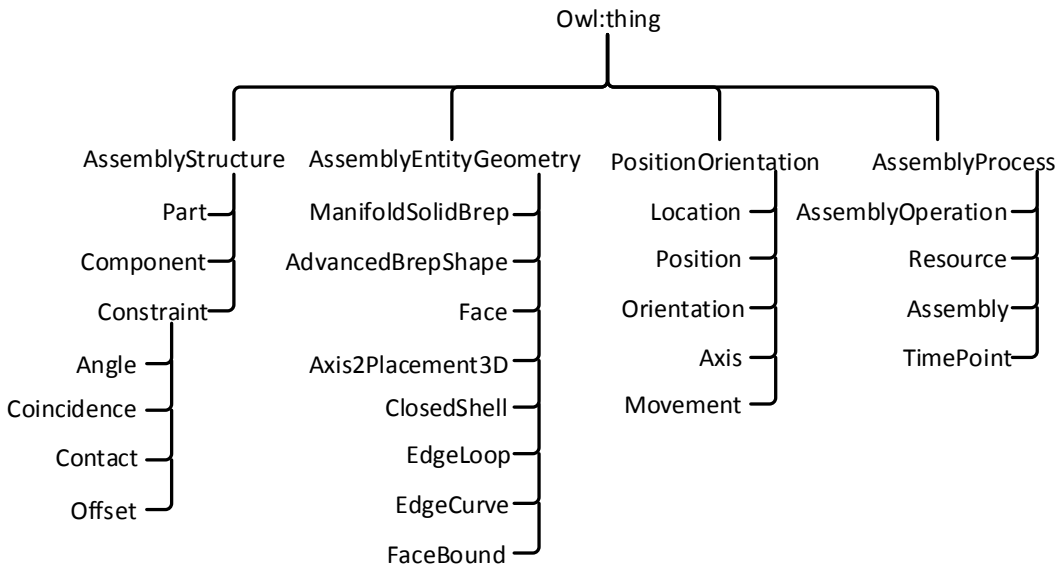


Figure 3. Class hierarchy of the ontology model

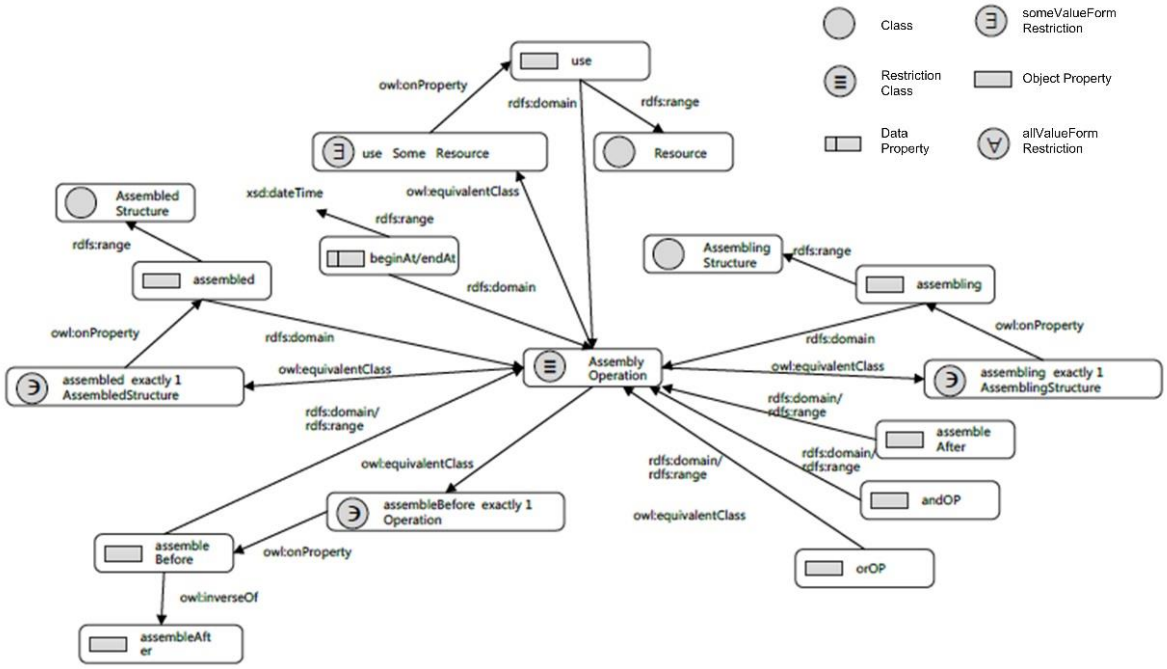


Figure 4. The OWL-DL model of Assembly Process in directed graph representation

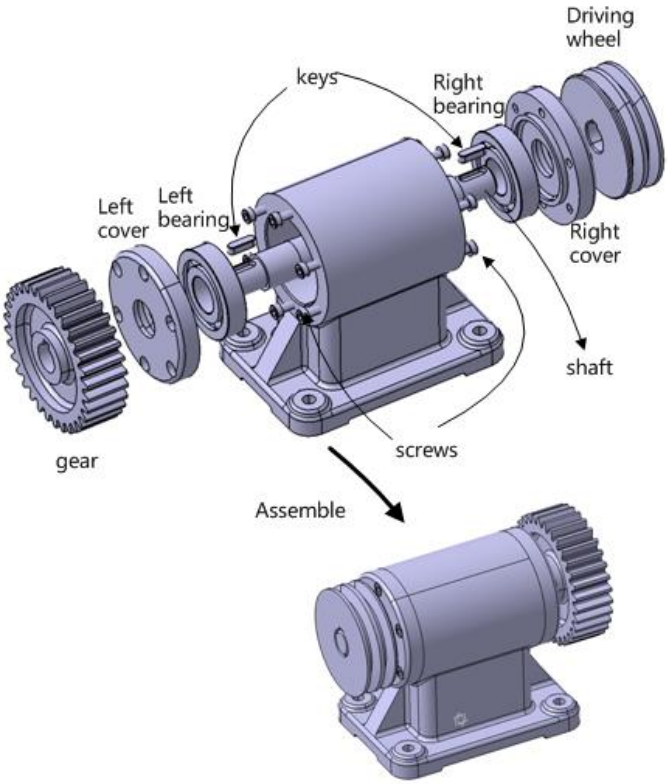


Figure 5. Structure of the transmission

Table 1. Properties defined in Assembly Structure

| Property | Domain | Range |
|---|---|---|
| isComposedOf[#1] | Product | Part/Component |
| compose[#1] | Part/Component | Product |
| hasConstraint[#2] | AssemblyStructure | Constraint |
| constrains[#2] | Constraint | AssemblyStructure |

*Note: Properties marked with #1 and #2 are two pairs of inverse properties*

Table 2. Properties defined in Assembly Process

| Property | Domain | Range |
|---|---|---|
| assembleBefore[#1 *1] | AssemblyOperation | AssemblyOperation |
| assembleAfter[#1 *1] | AssemblyOperation | AssemblyOperation |
| assembling[*1] | AssemblyOperation | AssemblingStructure |
| assembled[*1] | AssemblyOperation | AssembledStructure |
| beginAt | AssemblyOperation | xsd:dateTime |
| endAt | AssemblyOperation | xsd:dateTime |
| use[*2] | AssemblyOperation | Resource |

*Note: Properties marked with #1 are a pair of inverse properties. Properties marked with *1 have cardinality restriction with the cardinality as 1, while those with *2 have someValuesFrom restriction*

Table 3. Properties defined in Assembly Position and Orientation

| Property | Domain | Range |
|---|---|---|
| hasLocation | AssemblyStructure | Location |
| hasAxis | Location | Axis |
| hasPosition[*1] | Axis | Position |

| | | |
|---|---|---|
| hasOrientation*1 | Axis | Orientation |

*Note: Properties marked with *1 have cardinality restriction with the cardinality as 1*

Table 4. A portion of the properties defined in Assembly Geometry Entity

| Property | Domain | Range |
|---|---|---|
| haGeometry | Part | AdvancedBrepRepresentation |
| hasManifoldSolidBrep | AdvancedBrepRepresentation | ManifoldSolidBrep |
| hasClosedShell | ManifoldSolidBrep | ClosedShell |
| hasAdvancedFace | ClosedShell | AdvancedFace |

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) → before (?y, ?a) |
| 2 | Base part is assembled first | Base (?x) ∧ AssemblyOperation (?y) ∧ assemble (?y, ?x) ∧ AssemblyOperation (?z) → before (?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 | Assembly (?x) ∧ AssemblyOperation (?y) ∧ assemble |

| | | |
|---|---|---|
| | before external parts (Judging from the distance between the part bounding box and the product bounding box) | (?y, ?x) ∧ enhancedOnto:hasDistance (?x, ?u) ∧ Assembly (?z) ∧ AssemblyOperation (?a) ∧ assemble(?a, ?z) ∧ enhancedOnto:hasDistance (?z, ?v) ∧ greaterThan (?u, ?v) → before (?y, ?a) |

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) → before (?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) → before (?y, ?c) ∧ before (?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) → before (?y, ?c) ∧ before (?y, ?d) |