Product Design Ontology System Based on FBS-API

Jihua Wang
College of Information Science and Engineering Shandong Normal University, Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan, China
Email: jihuaaw@126.com

Huanyu Wang
College of Information Science and Engineering Shandong Normal University, Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan, China
Email: whuayu000@163.com

***Abstract*—Design ontology plays an important role in modern multi-party collaborations and knowledge-intensive designs. In this paper, a product design ontology system is proposed by analyzing and deconstructing the design function–behavior–structure (FBS) model and geometry application programming interfaces (APIs). Geometry APIs and surface behavior are considered minimum structure units and concrete function realization, respectively. APIs and their surface behavior are defined to build the semantic connection among functions, behavior, and structures and refine further the ontology of FBS design. According to some inference rules, design ontology can be used to obtain automatically or semi-automatically the preliminary product function and structure. The studied cases prove that the representation of the design knowledge based on the FBS-API ontology overcomes the limit of mere word conventions and is more conducive to the quantization and derivation of product design.**

***Index Terms*—FBS design; function; structure; geometry APIs; surface behavior; design ontology**

I. INTRODUCTION

Modern product design is multi-party collaborative, marketable, innovative, and knowledge intensive because of high product complexity, low development costs, high customer expectation for product functions, and mass customization. The emergence of web based and ontology technology meets these needs by sharing and reusing team design knowledge. In the product design process, which requires the cooperation of groups from different fields, ontology can effectively help people communicate and share knowledge about the current design by retrieving and reusing knowledge from previous designs.

Although computer-aided design (CAD) tools, such as AutoCAD, PRO/E, CATIA, UG, and PHOTOSHOP, provide large-scale electronic design documents as conventional design means, these tools are merely aimed at the integration of geometric data and are not knowledge oriented. The representation, retrieval, reuse, and sharing of design knowledge, including specifications, design rules, and constraints, from comprehensive resources by ontology technology is significant in existing CAD environments.

This paper focuses on the representation of product design ontology and its application on sharing, reusing, and modifying designs. Design ontology is a formal, sharable, and explicitly expressed concept system in the design process that reflects the relationship of concepts and their instances on functions, behavior, shapes, forms, structures, or layouts. Based on the function–behavior–structure (FBS) model, design ontology can be expressed comprehensively by geometry application programming interfaces (APIs) as basic structural ontology, as well as by surface actions in the API bulletin board [1] as basic behavior ontology. The FBS-API ontology system mainly includes the product, feature, geometry API, topology, utility function, decorative function, behavior, and surface classes. Geometry APIs are the logic units that conform to the intuitive thinking of the designer in product modeling. Geometric APIs involve the advanced integration of ordinary geometry ontology [2] and generate functions of the operated objects on CAD platforms. Geometric APIs are also known as CAD services that ensure interoperability among heterogeneous design tools in a distributed environment [3]. On the basis of converting design requirements into standardized functions, the FBS-API ontology can help designers obtain behavior graphs or trees and product structures by rules and inference engines. Ontology contributes to the achievement of design objectives as follows: 1) a number of existing design plans or knowledge retrieved by the similarity or correlation concept are modified as new design plans; 2) the original product design is completed according to the given design functions.

Manuscript received December 12, 2012; revised May 7, 2013; accepted September 7, 2013.
Corresponding author: Jihua Wang, jihuaaw@126.com. This work is supported by the Shandong Provincial Science and Technology Development Plan (Grant No. 2010G0020807) and the National Natural Science Foundation of China (Grant Nos. 69975010, 60374054).

doi:10.4304/jcp.9.3.543-550

This paper aims to refine FBS ontology (FBSO) and achieve a quantitative and semantic correlation between functions, behavior, and structures by geometry APIs and surface behavior to bypass simple word agreements. The rest of the paper is organized as follows. Section 2 reviews previous studies related to product design ontology, particularly design FBSO. Section 3 introduces the FBS-API design ontology. Section 4 provides the inference rules and their application on product design. Section 5 demonstrates a gear pump as an example. Section 6 concludes and discusses future works.

II. Background

Ontology has many definitions in artificial intelligence. Gruber [4] stated that ontology is a formal specification of a concept. This definition is the most broadly used definition of ontology in artificial intelligence. A concept is a simplified abstract view of a domain that describes objects, concepts, and relationships. Formal specifications imply the existence of a representational vocabulary, in which objects in the universe of discourse and relationships can formally be represented. Furthermore, axioms may be used to constrain the interpretation of terms in the vocabulary. Ontology should be reusable and shared across several applications. Borst [5] stated that ontology is a formal specification of a shared concept. A common understanding of a domain is defined by using ontology to support inter-human and inter-organizational communication in machine processes to support the semantic interoperation of different software systems [6, 7]. The ontology on assembly design, manufacturing, function, structure, general design, and so on has extensively been discussed in product design fields.

The product design process can be divided into four phases, namely, conceptual design, embodiment design, detail design, and engineering analysis. The most important work of product design mainly includes requirement-functional design, schematic design (behavioral design), layout design, and shape or preliminary structural design. The form or shape of artifacts is the result of the design. Function is the bridge between the customers' demand and physical behavior of artifacts, which are gradually embodied into behavioral and structural models. Some models for product design, product design ontology and design process ontology based on FBS were proposed [8, 9, 10, 11]. Yasushi [12] proposed a knowledge representation scheme for functions called FBS models. The sharing of designer intention, i.e., design rationale (DR), is important. Thus, Horváth [13] focused on modeling the DR of supplementary functions to establish the ontology and systematically capture designer knowledge for sharing among designers. Component functionalities and undesirable phenomena are integrated into an extended functional model used in multiple tasks, such as design, redesign, design review, and reliability assessment by using such concepts. The primary design concern is the generation of physical solutions to meet design specifications. Wynne [14] resolved two inherent difficulties: the modeling of complex interactions among various facets of a product, and the reasoning on the generation and selection of feasible solutions by expressing their functions, behavior, and structures.

Tudorache [15] referred to components, connections, systems, requirements, and constraints as engineering ontology. The component ontology represents the components and their whole-part decomposition. The connection ontology describes the system topology, i.e., the way in which components are interconnected with one another. The system ontology provides a formal representation of systems and combines concepts from component and the connection ontology. The requirement ontology contains the main concepts needed for the representation of requirements and their properties. The constraint ontology represents the mathematical relationships among model elements. Christel [16] proposed a feature ontology for design stages that involve the mapping of a specified function (or functional specifications) into a (description of a) realizable physical structure. The design artifact and process planning are the intermediate phases between design and manufacturing. Kyoung [17] presented the hierarchy of assembly design ontology classes that include features, spatial relationships, materials, and manufacturing. Li [18] discussed the functional, behavioral, feature, mating relation, and component ontologies in a layered FBSO model. Panetto [19] proposed first-order logic patterns to define the semantics of each construct of the standard conceptual models, such as class, attribute, association, and aggregation, generalization, and hierarchies. The semantics were defined by instantiating the IEC 62264 including the eight existing IEC 62264 models (product definition, material, equipment, personnel, process segment, production schedule, production capability, and production performance) and corresponding STEP-PDM modules. A multi-level assembly model has been proposed to capture the multiple levels of design information. This multi-level assembly model includes information for abstract, skeleton, and detail designs in the top-down assembly design process. Abstract design mainly involves abstract information, such as function and behavior. Skeleton design handles information on the spatial arrangement of components (layout). Detail design represents information on geometry and materials [20]. Kitamura [21] proposed a systematic description that provides fundamental concepts to capture target worlds based on functional ontology. He also proposed a common vocabulary for the description of functional knowledge that is applicable to other domains in the conceptual design phase. Ontology on device centered and process-centered designs have been discussed in artifact-knowledge modeling [22].

In summary, design ontology aims to share vocabularies and taxonomies and provide a conceptual data scheme (and generic/standard data model), semantic constraint for modeling, interoperability and integration, communication support and querying, capture and clarification of implicit knowledge, and basis for knowledge systematization [22]. Design ontology makes domain knowledge more explicit and improves consistency by providing precise semantics by employing formal axioms that constrain the possible interpretations of the meaning of terms. Current designs can be completed through the collaboration of professionals, such as industry designers, engineers, and customers, in various fields and procedures, such as sketching, examination, and appraisal. Web-based customization self-services encourage consumers to co-develop products by allowing anyone to engage in creative design work. Web services also help designers to grasp customer ideas accurately [23].

The above-mentioned ontology has not detailed the analysis of the form and behavior entities, which refer to a text descriptions or word conventions similar to function entities. For example, Oren [24] developed a descriptive model of the thinking process in design, including what stimulates creativity and how designers create design. His experiments show that behavior entities are stimulated more than functions and knowledge entities. Functions and knowledge entities involve fixed ideas, whereas form and behavior entities are more ambiguous and stimulate original ideas. Geometry APIs in the design process of CAD platforms can be used as the elementary particles of form or structure decomposition and infrastructure entities. Semantics on API surfaces can be used as the underlying behavior entities to refine and represent behavior in the product design FBS-API ontology.

III. Product Design Ontology Based on FBS-API

A. Design FBS Model

Product functions and structures are the primary factors in the design process. Products can be decomposed into equipment, components, and parts from the combination perspective. A designer does not focus on part assembly and parameter design but focuses on the overall structure or shape to meet customer requirements or functions in the initial stage, which is called the product 3D modeling design. The structure of a 3D model is decomposed into a tree or hierarchical mode according to geometric elements, such as features and geometry APIs in the modeling process (Fig. 1).

Structure decomposition comprises two geometry modeling function APIs, namely, geometric-topology APIs and geometric-body APIs [27]. APIs are smaller, granular, more normative, and more appropriate for exchanging and sharing interdisciplinary design semantics than features.

Decomposition elements from features to surfaces and data exist regardless of particle size to satisfy certain design requirements that need to be summarized. Functions are usually represented by verb–noun pairs, where the verb denotes the function semantics, and the noun represent the objects in the verb. Functions can be standardized to achieve function knowledge sharing with the function vocabulary, which is a lexical symbol and an artificial convention or abstraction. Product functions can be divided into three categories: utility, symbolic, and decorative.

Functions and structure features are artificial convention relations. For example, shape features are primarily named by their function. To describe functions formally, the concept of behavior is introduced to decompose and quantify functions. Behavior refers to how functions work and how structures interact with one another. Functions as abstract descriptions of design intent are specifically expressed and implemented with a series of behavior in accordance with certain principles or rules. Functions are also the subjective abstraction of behavior and have static characteristics. Behavior is the concrete realization of functions and has dynamic characteristics [25]. Zhang [26] summarized the FBS schools in Australia, Japan, the United States, and Europe. Functions with contexts have a certain degree of subjectivity, and behavior as a collection of states are in line with the stimulus-response mode, which are divided into kinematics, kinetics, and static behaviors. Status values and time are the parameters of the former two categories, which are called dynamic behavior. The external force is considered the input, whereas the stress without the finished material is the output of static behavior. A dynamic behavior can be transformed into multiple static behavior frames. Static behavior can only complete appropriate actions that are manifested by physical force, tension, or surface visual on the surface of certain structures. Several types of product behavior are formalized through physical forces. For example, a support force means that objects are placed or fixed, and a friction force denotes relative motion. Tension shows design claims by its curvature, ratio of length and width. Color or decorative patterns show the visual arts.

Definition: A surface behavior is the action bearing on the surface of a product structure and is manifested as a physical force, curvature, or a decorative pattern. Surfaces and surface actions constitute the main structure and functions of a product. If the surface behavior is B, surface is S, action is A, force is F, curvature is C, and decorative pattern is D, then $B = \langle S, A \rangle$ and $A = F|C|D$. S contains a surface shape function and its parameters, and A is a force vector, a curvature scalar, or a resource identifier.

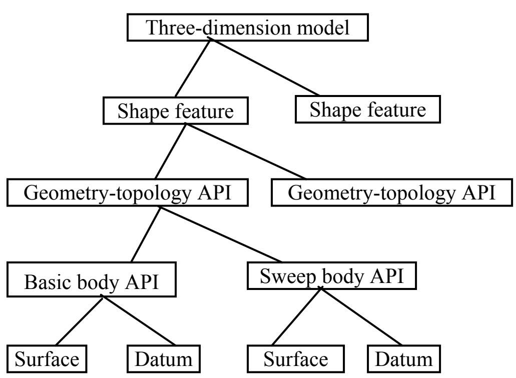

Figure 1. 3D model decomposition

Structures and behavior are coupled with surfaces and their actions. Types of behavior are combined into abstract functions, whereas structure features and functions are linked through the combination of a series of surface behavior. Thus, functions, behavior, and structures exhibit formal semantic and quantifiable relations and are not linked merely by the mandatory convention between functions and structure features.

B. Design Ontology based on FBS

Ontology is represented by classes, attributes, and instances and is a directed graph structure with concept nodes and relationship arcs. Concepts (sometimes called as resources) are associated with a predicate or attribute (arc). An attribute is also a concept or resource. The hierarchical design knowledge system expressed with the ontology based on FBS is shown in Fig. 2.

Product ontology is concerned about the concept of product categories. The structures of product categories need to be broken into parts or features and described individually because of structure complexity. Product functions are also the highest level of abstraction. Product ontology has at least four types of attributes, namely, has-function for practical functions, has-symbol for symbolic functions, shape for the supplement of a product shape or structure, and whole-part for product composition. The has-function is a data-type attribute that corresponds to a word in the function vocabulary. The has-symbol belongs to an object attribute whose object is a brand, a totem pattern, and an animal or plant pattern. Shape has multiple sub-attributes, such as pictures, performance parameters, and maintenance manuals. The whole-part attribute considers feature instances as the compositions of a product structure. Features are geometry constructions that independently complete functions that map symbols and physical functions. For example, the design of semi-circular instrument panels needs to meet ergonomic visual requirements and confer speed and passion to customers. Thus, the function attributes of features need to include at least the has-function and has-symbol. The parameters used by the parametric feature design components in current CAD systems are the shape attribute values of features.

Geometry APIs as micros and mathematical expressions in geometric disciplines are the basic geometric particles according to designers. The concepts of APIs easily reach connotation unity in cross-CAD systems and interdisciplinary collaboration. APIs are expressed by function formulas and their parameters, wherein surfaces are marked with unique sequence numbers of the entity list in the API bulletin. Surfaces are entities expressed by shape functions, normal functions, and so on, and surfaces with actions are the surface-behavior attributes of APIs. Every API instance involves the has-function attribute according to the lowest level of the function vocabulary and shape attribute which are API parameters.

Action entities contain forces, curvatures, colors, and planar patterns, among which the planar pattern is an object attribute. Action depends on surfaces, and a behavior is a behavior-surface pair.

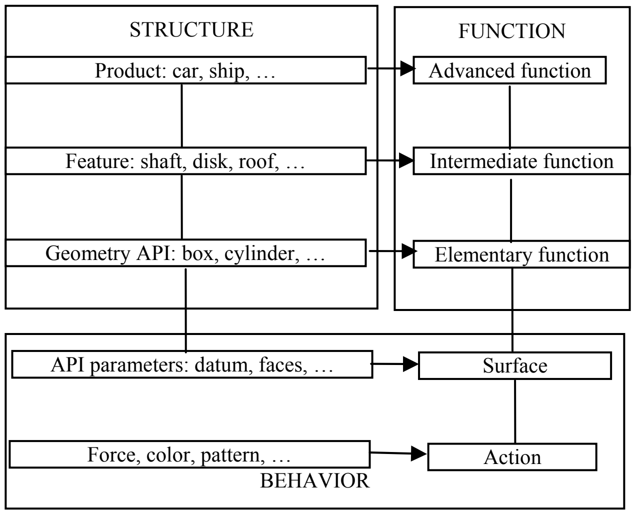

Figure 2. FBS-based ontology

The structure instances on the left of the above figure from top to bottom present a whole-part relationship, whereas the upper level involves whole-part attributes. The functions on the right layers expressed by words state the purpose of the corresponding structure and also express a whole-part relationship from top to bottom. For example, features consisting some API instances are geometric constructions with certain functions and involve at least whole-part and has-function attributes.

In addition to the above-mentioned attributes, the structure in every level has instance-of and topology (TOPO) attributes. The individual structures are the instances of geometry structure classes expressed by instance-of. The design semantics (e.g., structure individuals that replicate themselves and change their own parameters) are expressed by TOPO attributes, such as circular array.

C. Geometry API Ontology

Geometry API and surface behavior are the bases of the above design FBSO. Furthermore, APIs are easier to be unified, shared, and exchanged in different enterprises and areas than features because of the unique connotation.

The methods of geometry modeling in current CAD engines mainly include constructive solid geometry, boundary representation, parameter representation, cell representation, and their combinations. Designers complete product modeling by clicking on the menus in the design process. Menus that act as method macros are geometry body APIs and geometry topology operation APIs that have whole geometry shape semantics. Thus, APIs can be regarded as elementary particles that constitute a product model, and each particle has the same design and geometry semantics. A series of geometric entities (e.g., surfaces) and topology entities (e.g., loops) generated by APIs constitute the collection of independent objects. The collection can either be integrally operated or individually operated. These APIs are called geometry modeling functions or the feature functions of minimum granularity. The common API entities are shown in Table Ⅰ [27].

TABLE I.
API ENTITIES

API entities	Basic APIs	Box, pyramid, wedge, dome, sphere, cone, torus, poly body
	Sweep APIs	Extrude, revolve, sweep, loft

Each API is a subclass that is inherited from the parent class, namely, GEO_API, which contain the following attributes:

1) Shape attribute: local coordinate, datum, boundary curve, center, moment of inertia, and weight. The attribute values are object types or data types such as a real value, point coordinate, line, spline, and spiral.

2) Surface-behavior attribute: every surface with an action is an API attribute that contains the name of a surface entity and the sequence number in the API bulletin board, including planar(#), sphere(#), cylinder(#), pface(#), rulesurf(#), tabsurf(#), revsurf(#), and torus(#). The attribute values are object types or data types such as patterns, forces, colors, and curvatures.

3) Has-function attribute: the overall API instance as the minimum function unit corresponds to the elementary function vocabulary.

4) TOPO attribute: API instances are combined with a geometry body through self-replication, arraying, and Boolean operations. The attribute values are data types (e.g., real) or object types (e.g., APIs). Such attributes reflect spatial topological relationships, such as the relative position of the API instances (Table II).

D. Function Ontology

Functions are the abstract vocabulary of verb–noun pairs. A part of function words in the mechanical field is shown in Table III.

A higher function level indicates a higher level of abstraction and generalization. A high-level function is composed of a number of low-level functions. The lowest-level functions are implemented through surface behavior according to certain physical theorems. Function decomposition is the main method used to analyze complex functions. Top-down decomposition is a tree mode, and root nodes are the overall function of a product.

TABLE II.
API-TOPO ATTRIBUTES

Topology	Modification	Presspull, chamfer, fillet, move, rotate, mirror, array, scale_array, scale, align
	Boolean operation	Union, intersect, subtract
	Euler operation	face_extrude, _move, _offset, _rorate, _taper, _sperate, _shell

TABLE III.
EXAMPLE OF FUNCTION WORDS

Sort	Verb and its symbol	
Joiner	Bolt assembling	BA
	welding	WD
	riveting	RV
	suspending	SP
Transmission	Screw driving	SD
	Belt driving	BD
	Gear driving	GD
…		

By contrast, leaf nodes are the lowest-level functions. High-level functions involve multiple whole-part attributes for subordinate functions, which have inverse attributes such as the whole-part attributes of corresponding low-level functions. Similarly, the lowest-level functions involve multiple whole-part attributes for surface behavior.

In addition to the lowest-level functions, high-level functions that serve as black boxes are marked with one or several of typical surface behavior of input or output. Thus, such high-level functions have input and output attributes.

Aside from the hierarchical relationship of functions, a correlation exists among all functions: some functions are attached to the core function. Product design generally starts with the most important core functions. Other support and auxiliary functions are designed around the core functions. Therefore, the core function has multiple support attributes supported by values that are auxiliary function instances.

IV. APPLICATION OF THE DESIGN ONTOLOGY

Design knowledge represented by ontology is stored in the knowledge base, which needs to increate and operate through the semantic framework including field knowledge modeling, storage, retrieval, validation, analysis, and reasoning. The design knowledge base is a collection of statements (resource description framework triples) whose main contents are explicit facts. The inference engines entail the implicit facts in the semantics and rules by forward and backward link reasoning. The purpose of establishing design ontology is to obtain satisfactory ontology elements to constitute design plans by querying and reasoning according to the rules and requirements.

The four types of rules expressed with web ontology language triples and semantic web rule language are proposed to elucidate different design goals.

A. Function-oriented Design

The design of function in forward engineering is to find auxiliary functions based on the core function and then decomposes these auxiliary functions into the lowest-level functions. The design result is a tree with nodes of function words.

Whole-part is simplified as Part, where ?core-x is a core function variable, and RULE1 is the rule of function design:

Supported-by(?core-x(i))∧…→Part(?core-x(i))∧… ∧Part(supported-by(?core-x(i)))∧…→function-tree(?core-x(i)).

B. Design from Behavior to Basic Function

Designers cannot miss basic lowest-level functions as minimum units of product functions to link the structure directly and obtain satisfactory structure design. The basic functions as vocabulary terms are quantitatively expressed with some surface behavior in the APIs. To avoid the redundancy of the definition of terms, ontology engineers need to query the system according to the combination of surface behavior and provide new names to basic functions when the same basic function does not exist.

?b is a surface-behavior variable, and RULE2 is the rule of basic function design:

$$API(surface\text{-}behavior(?b1) \wedge surface\text{-}behavior(?b2) \wedge \ldots) \rightarrow has\text{-}function(?b1 \wedge ?b2 \wedge \ldots).$$

C. Design from Function to Structure Feature

Features are structures that correspond to mid-level functions, which are the combination of basic API functions. The feature design decomposes mid-level functions according to RULE1 and achieves API combination according to RULE2.

?f is a function variable, and RULE3 is the rule of feature design:

$$Part(?f1) \wedge Part(?f2) \wedge Part(?f3) \wedge \ldots \rightarrow has\text{-}function(Part(?f1)) \wedge has\text{-}function(Part(?f2)) \wedge has\text{-}function(Part(?f3)) \wedge \ldots \rightarrow API1 \wedge API2 \wedge \ldots.$$

D. Design from Critical Behavior to Product Structure

The machine will automatically reason and export other behaviors, functions, and structures as long as critical behavior of typical marking functions are provided by designers in CAD. The structure plans are manually selected and determined. The machine deduces high- and mid-level functions based on typical behaviors. Thereafter, the machine finishes the function design and structure design according to RULE1 and RULE3 (i.e., API combination), respectively.

?tb is a typical behavior variable, and the following equation expresses RULE4:

$$Has\text{-}function\ [input1(?tb11) \wedge output1\ (?tb12)] \wedge \ldots \rightarrow RULE1 \rightarrow RULE3 \rightarrow API1 \wedge API2 \wedge \ldots.$$

The ultimate goal of product design is to obtain the product structure or shape. Designers summarize core functions from complicated requirements and obtain auxiliary functions by machine reasoning with the correlation supported-by attributes. The above analysis illustrates that functions are decomposed into a tree and that basic functions as leaf nodes are mapped with a series of surface behaviors on APIs. A function is changed into a set of surface behavior, and APIs that constitute a product structure are obtained by the backward link inference according to these behaviors. The above four rules are interrelated, and the machine uses these rules automatically to export the preliminary designs for manual selection.

V. Case Demonstration

Product design ontology is established in a variety of software integration environment, including the compiler tool Java 1.6, the editing tool Protégé, the inference engine Hermit 1.3.6, and the geometry modeling tools AUTOCAD and ACIS. The design knowledge-acquisition method is semi-automated. The surface behavior and functions of APIs are produced with the help of field experts, and other knowledge is automatically obtained by machines. The design knowledge of our subject is based on the cases of mechanical products and can be extended to the product design of other fields.

The design based on ontology is specifically addressed by a gear pump as an example. The design ontology related to gears is shown in Fig. 3. Such design ontology is generated from the concepts in Chapter 3 at our subject experimental platform.

The liquid is pressurized and is delivered by a pair of meshing gears in a gear pump. The meshing gears that complete the core function are the design focus, whereas other structures complete the auxiliary functions around them. For example, housing needs to support the gears, seal the liquid, and decorate the appearance, whereas the gear shaft provides the driving power. The meshing gear structures comprise the active and passive gears with the shafts at both ends. A longer input shaft needs to be designed for the active gear to install the power gear.

"Liquid pressure" as a core function links its auxiliary functions with the attribute "supported-by," including the functions of liquid seal, power input, and visual appearance. A number of auxiliary functions are obtained by querying the core function with RULE1. Each function is decomposed into types of surface behavior, such as input power and liquid pressure and their surfaces, with RULE2. The power gear transmits the force on the spline surface <peditsurf(#20), (1,0,0)> into the active gear shaft. The force of the gear surface <peditsurf(#300), (−1,0,0)> overcomes the liquid resistance to increase the liquid pressure. The cylinder surface of the gear shaft supports the gear by the force <cone(#150), (0,1,0)>; thus, the gear pair as a feature completes the "liquid pressure" function. The central cylinder cone-subtract removed from each gear is aligned with the reference cylinder cone-datum to reduce the weight and moment of inertia. The API combination is obtained by backward link reasoning based on the surface behavior with RULE3. Designers change the attribute values of APIs, such as the datum diameter, to determine the appropriate plan, which is then stored in the knowledge base to enrich the design knowledge.

Designers can find related data according to any function, behavior, and structure attributes in the geometry API ontology-based product design platform. This approach is a great solution for querying the plans designed by other designers according to the design requirements of sharing and reusing design knowledge.

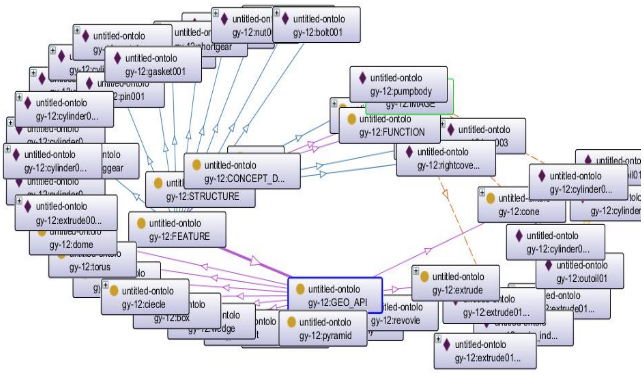

Figure 3. FBS-API-based ontology of gear pump.

VI. Future Works

This study solves the representation, sharing, and reuse of design knowledge by further analyzing and deconstructing products designed with FBS models based on ontology.

However, a number of problems remain that must be solved. Instances are difficult to create and improve automatically. The conversion from design intent to function artificial support is needed. This approach will produce knowledge inconsistency. Structure design elements are regulated by APIs but standards need to guide the further development of entities on function vocabulary and behavior classification because of the lack of a unified specification.

Our future works would include the establishment of the unified and standardized constituents of functions and behavior and the development of algorithms to build automatically the design ontology from 3D CAD model data files, extend the design ontology into the product life cycle, and unify APIs from heterogeneous CAD tools in ontology concept systems.

Acknowledgment

This work is supported by the Shandong Provincial Science and Technology Development Plan. This study aims to establish the feature models of products and represent design knowledge in a collaborative design environment. Design efficiency and quality can be improved significantly by retrieving, reusing, and sharing the design knowledge represented by ontology.

References

[1] Zhan H S, Li G X, MA Z X. *Geometry Modeling Technology and System Development Based on ACIS*. Beijing: Tsinghua University Publisher, 2002, pp.107-116.

[2] Zhong X Q, Fu H G, She L, Huang B. "Geometry knowledge acquisition and representation on ontology," *Chinese Journal of Computers*, vol. 33, pp. 167-174, 2010.

[3] Gao S M, He F Z. "Survey of distributed and collaborative design," *Journal of Computer-aided Design & Computer Graphics*, vol. 16, pp. 149-157, 2004.

[4] Gruber T R. "Towards principles for the design of ontologies used for knowledge sharing," in *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Guarino N and Poli R, Eds. Netherlands: Kluwer Academic, 1993.

[5] Borst P. "Construction of Engineering Ontologies for Knowledge Sharing and Reuse," PhD thesis, Universiteit Twente, 1997.

[6] Noy N F. "Semantic integration: A survey of ontology-based approaches," *SIGMOD Record, Special Issue on Semantic Integration*, vol. 33, pp. 65-70, 2004.

[7] Noy N F, McGuinness D L. "Ontology development 101: A guide to creating your first ontology," *KSL-Report KSL-01-05, Stanford Knowledge Systems Laboratory*, 2001.

[8] Hao Y T, Dai P, Lou D M. "The complex product design modeling based on spatial sequence colored Petri-nets," Journal of Software, vol. 8.8, pp. 1818-1826, 2013.

[9] Zhou J, Watanuki K. "Skill ontology for mechanical design of learning contents," Journal of Software, vol. 7.1, pp. 61-67, 2012.

[10] Yuan C F, Pang G b, Wang W L. "Approach of customer requirement analysis based on requirement element and improved HoQ in product configuration design," Journal of Software, vol. 7.3, pp. 691-698, 2012.

[11] Gero J S, Kannengiesser U. "A function-behavior-structure ontology of processes," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 21, pp. 166-197, 2007.

[12] Umeda Y, Ishii M, Yoshioka M, Shimomura Y, Tomiyama T. "Supporting conceptual design based on the function-behavior-state modeler," *Artificial Intelligent for Engineering, Design, Analysis and Manufacturing*, vol. 10, pp. 275–288, 1996.

[13] Horváth, Xirouchakis. "Towards modeling design rational of supplementary functions in conceptual design," *Proceedings of the TMCE 2004*, Switzerland, 2004.

[14] Wynne H, Irene M Y. "Current Research in the conceptual design of mechanical products," *Computer-Aided Design*, vol. 30, pp. 377-389, 1998.

[15] Tudorache T. "Employing ontology for an improved development process in collaborative engineering," PhD thesis, University of Berlin, 2006.

[16] Christel D, Parisa G, Michael G, Denis P, Ram S. "CAD/CAPP integration using feature ontology," *Concurrent Engineering*, vol. 15, pp. 237-249, 2007.

[17] Kyoung Y K, Hyungjeong Y, David G M. "Assembly design ontology for service-oriented design collaboration," *Computer- Aided Design & Applications*, vol. 3, pp. 603-613, 2006.

[18] Li Z J, David A, Karthik R. "Ontology-based design knowledge modeling for product retrieval," *ICED05 Melbourne*, pp. 1-15, 2005.

[19] Panetto H, Dassisti M. Tursi A. "ONTO-PDM: Product-driven ontology for Product Data Management interoperability within manufacturing process environment," *Advanced Engineering Informatics*, vol. 26, pp. 334–348, 2012.

[20] Chen X, Gao S M, Yang Y D, Zhang S T. "Multi-level assembly model for top-down design of mechanical products," *Computer-Aided Design*, vol. 44, pp. 1033-1048, 2012.

[21] Kitamura Y, Mizoguchi R. "Ontology-based systematization of Functional knowledge," *Journal of Engineering design*, vol. 15, pp. 327–351, 2004.

[22] Yoshinobu K. "Roles of ontologies of engineering artifacts for design knowledge modeling," *Proceeding of the 5th international seminar and workshop engineering design in integrated product development, Poland,* pp. 59-69, 2006.

[23] Elizabeth M. Gerber, Caitlin K. "Supporting creativity within Web-based self-services," *International Journal of Design*, vol. 6, pp. 85-100, 2012.

[24] Oren B. "Creative stimulation in conceptual design," *Design Engineering Technical Conferences and Computer and Information in Engineering Conference, Canada*, 2002.

[25] Tomiyama T, Umeda Y, Yoshikawa H. "An CAD tool for functional design," *Annals of the CIRP*, vol. 42, pp. 143-146, 1993.

[26] Zhang W J, Lin Y, Sinha N. "On the function-behavior-structure model for design," *Proceeding of the Canadian Design Engineering Network (CDEN), Alberta,* 2005.

[27] Wang J H, Liu H. "Part genome based on geometry features in evolutionary design," *Computer Integrated Manufacturing Systems*, vol. 15, pp. 21-27, 2009.

Jihua Wang was born in Shandong Province in China in 1966. He received the B.E. degree in Tsinghua University in 1990, the M.S. degree in Shandong University in 2005 and the Ph.D. degree in Shandong Normal University in 2009, respectively. He is currently an associate professor in Information Science and Engineering College in Shandong Normal University. His research interests are design ontology, and intelligent design.

Huayu Wang was born in Shandong Province in China in 1964. He received the B.E. degree in Shandong University in 1986, the M.S. degree in Shandong University of Science and Technology in 2002 respectively. He is currently an associate professor in Information Science and Engineering College in Shandong Normal University. His research interests are computer graphics and intelligent design.