Advanced Engineering Informatics 39 (2019) 331–346

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Review article

On the role of generating textual description for design intent communication in feature-based 3D collaborative design

Yuan Cheng[a], Fazhi He[b,*], Xiao Lv[b], Weiwei Cai[b]

[a] School of Information Management, Wuhan University, 430072, China
[b] School of Computer Science, Wuhan University, 430072, China

ARTICLE INFO

Keywords:
Feature-based collaborative design
Design intent communication
Semantic network
Natural language generation

ABSTRACT

Modern manufacturing firms are more inclining to promote the product quality, save costs and reduce times of product design by both collaborative designing and model reuse. If CAD components constructed collaboratively have information representing their developers' design intents embedded in the model, people's understanding over the product should be improved and the product model should be best reused. Until now, capturing, recording and presenting design intents still remains a challenge. It has been shown by empirical studies that textual summarisations can lead to improved decision making. In this paper, we propose an approach to generation the natural language description about design intents of collaboratively developed product. The approach brings together techniques from different areas of collaborative designing, ontology and semantic network, and natural language generation. The language generation process is guided by an information model we established to give a structured description about design intents of collaboratively products. In order to record information related to the design intents, we build a common CAD model ontology and then generate a semantic network to describe dependencies, component structures and design history which are components of the design intent information model. The techniques of natural language generation, namely discourse planning and sentence planning, are adopted for the eventual linguistic generation of design intents. Finally, we use several case studies to prove the advantages of natural language in helping people better understanding the design intents.

1. Introduction

In an era when products are more structurally and functionally complicated, various widely geographically dispersed firms operating under different cultures and languages participate in the lifecycle of industrial plants, which increases the quality and competitiveness of design products and also reduces the costs, design flaws, development cycles for new products [1]. Successful firms today are those that are able to manage concurrent activities, gathering knowledge, wisdom and sufficient expertise in an intra- or inter-organizational network and deliver value to the marketplace [2,3]. A broad spectrum of knowledge is used and shared in these modern collaborative but distributed firms. Product knowledge needs to be specified in languages with formal semantics and driven by experts conceptualizations about their domains of expertise [4–6]. Feature-based modeling approaches have played a relevant role for qualitative knowledge specification and integration since 1970's [11,12]. Feature-based CAD software packages, such as Solidworks, Unigraphics NX(UG), Pro/Engineer and CATIA, are currently considered the state-of-art platforms for product modeling [4,13]. In these CAD software packages, design intent is typically conveyed implicitly within the CAD model in the form of relations among features of the model [10], which are typically displayed as a design tree in the interface of the parametric modeling software.

Design intent is known as a special kind of design knowledge. Based on Pena-Mora's [7] summarization of the significance of representing design intent, the importance of recording, structuring, presenting the collaborative design intents in an explicit way lies in:

- It leads to a more intelligent use of resources and knowledge by providing a more available way for review design information. The overall quality of the product increases.
- Changes in complex projects require certain design decisions to be modified during the development process. When the justifications defined during the initial stages are lost, they need to be recreated, which has a negative impact on project costs and development times.
- Efficient communication of design intent is essential for integrating solutions and transferring design knowledge. To benefit from the

* Corresponding author.
E-mail address: fzhe@whu.edu.cn (F. He).

https://doi.org/10.1016/j.aei.2019.02.003
Received 3 October 2018; Received in revised form 31 December 2018; Accepted 5 February 2019
Available online 16 February 2019

functionality provided by modern CAD systems, to quickly and efficiently modify existing designs, users interacting with the model must understand the decisions behind the modeling process [8,9]. Despite major advances in CAD technology, models are difficult, or impossible, to alter when design intent information is lost or not communicated properly [10,15].

Reusing CAD model is considered to be a critical factor for a successful implementation of collaborative product design methodologies. A feature-based 3D CAD model should be machine understandable. This is especially important in the situation of when a model is to be edited within heterogeneous CAD systems, e.g. data exchange. A lot of important information relating to design intents are lost which has a bad effect over the model exchange, sharing and archiving. A study performed by Brunner meier and Martin at National Institute of Standards and Technology (NIST) estimated that the cost for dealing with the data loss problem like this in U.S. automotive supply chain alone was about 1 billion dollars per year.

Both the Ontology-based data exchange methods and the already-put-into-practice STEP approach need manual interventions [14]. This means that, a 3D part should be human understandable. However, as it is suggested and demonstrated by Meulen that “a graph is poorer than 100 words” and people’s understanding of 2-D graphs is rather limited, and that people make numerous errors and require complex cognitive processes when interpreting even simple 2-D graphs [16]. Considering that a 3D part is more structurally, functionally and semantically complicated than a 2D graph, we inferred that people are facing more challenges than what they have encountered in comprehending 2D graphs and text [17]. NLG (Natural language generation) has been widely used in translating signals [18–22], graphs [23–27] and images [28–31] into human understandable texts. These works demonstrated the ability of natural language is assisting people in understanding the complex semantics from different kinds of input.

In this paper, we are going to present an approach to generate linguistic descriptions about design intents of collaboratively constructed products so as to support design intent communications among designers. The contribution of this paper are summarized as follows:

- We built an information model for representing design intents, which were hard to describe formally, contained in 3D CAD parts developed in a collaborative environment. This model tends to cover the following aspects such as what the product is (e.g. the type), who contributed in co-designing the model, the collaboration procedure for developing the product, design features composing the model, sub-components and the dependencies between features and the design history.
- We developed a common CAD ontology and then established a feature semantic network to describe the collaboratively developed models and to represent of the design intent model. According to the established concept hierarchy and property hierarchy, the features are related within the semantic network, which helps to identify features, dependencies, component structure and design history.
- We devised an approach to generate natural language descriptions of collaboratively constructed products’ design intents. The approach takes the Feature Semantic Network as the input. The ultimate description is presented at the feature-level. The discourse planning step determines the structure of the text dynamically by confirming the priorities among features. We build a template base for describing different aspects of an individual feature.

The outline of this paper is organized as follows. Section 2 reviews the related works on design knowledge categorization, representation and communication. Section 3 gives a system overview of the proposed approach. In Section 4, the design information model that outlines the backbone of our natural language based design intent communication approach is introduced. In Section 5, the CAD model ontology developed is introduced. In Section 6, we introduce in detail the approach to generate a Feature Semantic Network of any CAD model and the NLG process. In Section 7, critical issues regarding to the implementation are introduced and an experiment is explained specifically. In Section 8, a discussion over the limitations of this approach is given. Finally, in Section 9, a conclusion is made and the future work is introduced.

2. Related works

2.1. Design knowledge categorization

The classification of design knowledge is of importance in deciding the types of knowledge in need and where they can be captured. Design knowledge can be generally divided into formal knowledge and tactic knowledge. Tactic knowledge refers to those kept in people’s brains and difficult to be codified. Johansson [32] classifies design knowledge into more specifically categorization: heuristic, analytical, numerical and empirical. Owen and Horvth [33] classify representation formats into five categories: pictorial (e.g. sketches, drawings, and pictures), symbolic (e.g. decision tables, and production rules), linguistic (e.g. verbal and textual communications), virtual (e.g. CAD models and simulations), and algorithmic (e.g. computer algorithms, and mathematical equations). Poorkiany [34] claims that in the early stages of design the presentations are more linguistic, e.g. requirements, and pictorial in nature, while by evolving the design process, the presentations are predominantly virtual and symbolic. In the design data representation model CPM (Core Product Model) presented by Fenves and his colleagues [35,36], the knowledge of a product involves its form, function, behavior, material, physical, and functional decompositions, and their relationships in the process of product design. Even designer’s interactive activities with CAD software packages are considered as valuable design knowledge [37].

2.2. Representing design intent

An effort to summarize the characteristics of existing design intent representation approaches is given Table 1. In academia, solutions for representing design intent can be divided into 4 categories:

- *The standardized neutral file based approach.* AP 203 and AP 214, and AP 242 in compliance with JT are representatives in such a category. The combination of AP 242 and JT is in response to the incapability for representing design intents by merging and extending the AP 203 and AP 214. JT for visualization supports common product structure-to-file structure mappings. Translated geometric definitions are globally OK but the exchange of PMI specifications is moderate satisfactory because of its weakness in representing product semantics. Even though, most STEP translators still be viewed as transferring ‘dumb’ shape models representing the final result [14]. They are not able to represent design intent related information such as construction history, parameters, constraints and features are completely lost.
- *The ontology-based semantic modeling approach.* Ontology is the semantic basis of communication among domain entities [57,58]. They can be used to share knowledge among systems or users, and to make knowledge based inference or reasoning [72]. However, a major problem of the ontology models is the ineffectiveness to address the multi-faceted feature of design knowledge as design involves different issues requiring knowledge at different levels and from different sources.
- *The language-based semantic modeling approach.* Knowledge-based engineering is characterised by its language-based, OO approach [70]. Better language based technologies is beneficial in developing expressive tool to understand and analyze at each stage of the design process, and to reason about across stages. Different forms of

Table 1
A summarization of design intent representation approaches.

Design intent representation method	Scope	Literatures, standards, models	Represented design knowledge	Representation format
The standardized neutral file based data approach	CAD model data exchange	AP 203 [38,39] AP 214 [40]	Geometric data	STEP neutral file in EXPRESS language
	Product Manufacturing Information (PMI) exchange	AP242 [41,42] and JT [43]	Geometry structure attributes	JT and AP242 XML file
	Interoperability of plant lifecycle data	ISO 15926 [4]	Specifications of a plant item Logical configuration information 3D shape of a plant item	OWL and RDF
Ontology-based semantic modeling approach	Product data exchange	Dartigues et al. [44]	Process parameters Process description Product description Tool description Equipment description	Ontology developing language
		Patil et al. [45,46]	The form, function and behavior of assemblies and artifacts; Specifications; Features; Constraints	PSRL
		Seo et al. [47]	Features	F-Logic
	Feature recognition and model retrieval	Tessier et al. [48] Kim et al. [49,50] Jeon et al. [51] Lu et al. [52] Qin et al. [53] Abdul-Ghafour et al. [54]	Features with topology Assembly features with constraints Knowledge in Tyer design Features Feature attributes	OWL + SWRL OWL + SWRL OWL + SWRL OWL + SWRL
	System entity mapping in CPS	Petnga et al. [54] Jenson et al. [55] Bhave et al. [56]	System entities from cyber and physical components in a CPS	OWL
Language-based representation approach	CAx system integration	Zhu et al. [59] Sanfilippo [61]	Features	Ontology description language
		Sun et al. [60]	Features	Ontology
	Interoperation between ontologies and the STEP standard	Barbau et al. [59] Sun et al. [60]	Features	Ontology
	Design intent communication	Camba et al. [62] lducin-Quintero et al. [63]	3D annotations	Natural language
	Virtual testing in CPS	Taha et al. [64]	Visual and geometric presentations Basic mechanics and dynamics Control and etc.	Hybrid modeling language
	Geometry and simulation modeling	Gross et al. [65]		
	Parametric CAD Data Exchange	Altidor et al. [66]	Formal semantics of parametric CAD data and analyses	Programming language developed for CAD software packages specifically, e.g. *SWlang*
	Present design rationale	Bracewell et al. [67]	Market information associate with a product model	Graph-based language natural language
Hybrid approach	Knowledge management	Peng et al. [68] Wang et al. [69] Poorkiany et al. [34] Schmidt et al. [71]	User requirements CAD model FMEA Design decisions	The visualized CAD models Spreadsheets Textual description Chars

languages are adopted, such as natural language, graph-based language and modeling language. A language is chosen with the consideration of its expressiveness in describing a certain type of design knowledge.

- *The hybrid approach.* The hybrid approach incorporates different kinds of schemes for knowledge representation in one application. Its biggest advantage lies in that, for each type of design knowledge, it facilitates the most suitable representation scheme. For some occasions, it is capable of representing knowledge from multiple sources but less concentrating on reasoning. The biggest challenge lies in knowledge retrieval and mapping so the different schemes point to one target for representation.

Yet, it is worth mentioning that the deficiency the STEP standard and the ontology-based semantic modeling have in common is that Neither of which is completely capable of exchanging the non-geometric data. The essential elements of the lost information include construction history, constraints, features, dimensional parameters or other values providing an indication of what it is permissible to change.

2.3. Communicating design intent

The essence of communicating design intent is to provide an environment so designers are free to share, discuss their perceptions and eventually come up with an optimal solution. Both plugins for CAD software packages and external systems are developed to implement the communication. External systems involve using knowledge managing (KM) system or knowledge-based engineering (KBE) systems to process information and facilitate knowledge-based engineering analysis as well as sharing knowledge with colleagues to improve decision making. Typical examples are Compendium [73] and DR editor (DRed) [67]. Other approaches include argumentation-based models such as Decision Representation Language (DRL) [74], which was further extended by Software Engineering Using Design RATionale (SEURAT) [75], and the Question, Option and Criteria (QOC) technique which emphasizes discussions of alternatives regarding artifact features [76]. Such external systems are capable of supporting hybrid knowledge representation approach. A smart collaborative system is described to provide useful knowledge for users with various roles within a collaborative design process is developed [68]. It addresses the separation of the personalisation and codification views of design knowledge. Wang et al. [69] has developed a collaborative knowledge management system driven by user requirement actively respond to the dynamic user situation and diverse user experience. Poorkiany [34] has developed a system to support the decision making process across the automobile development process.

3. An overview of our approach

In this paper, we are going to propose a natural language based method for better communication of design intents among designers of heterogeneous CAD software packages. The goal is to prevent the later-coming users of 3D parts from misunderstanding or even ignore critical design intents by viewing-and-guessing the 3D model or comprehending the absurd parametric representation of feature modeling command history. The overall procedure of our approach is shown in Fig. 1. The core parts of this architecture are describe as follows:

1. Design intent information model: This model sets a framework for describing design intents of collaboratively constructed model in a structural fashion by including five components. It determines the content that should appear in the textual description.
2. CAD model ontology: The CAD model ontology defines concepts and relations which are important constituents of a Feature Semantic Network. Moreover, it sets regulations for XML schemas which define the structures and contents for different types of feature modeling operations.
3. Feature semantic network generator: This is the upstream step of the NLG workflow which is responsible for generating the semantic representations of a given artefact at the feature level. Two sub-components are contained within the generator: design history generator and design history processor. The design history generator is to generate the XML-formatted file for each feature modeling command following the rules set by XML schemas. A feature modeling command in its XML format is transmitted to other cooperative sites for execution because the structural representation of an XML file makes it easily machine interpretable. The design history processor receives the design history from any site and extends each XML file to append more information relating a feature. Another important task for it is to execution concept mapping with the CAD model ontology so as to make assertions of individuals and properties and establishing the feature semantic network eventually.
4. Linguistic description generator: This is the downstream the NLG workflow which takes the feature semantic network as the input and generates the ultimate textual description over the design intents of a CAD model. Two sub-components are contained: discourse planner and sentence planner. The discourse planner determines the structure of the output text by fixing the order of composing features automatically. This is indispensable because the features created by concurrent operations need to be serialized to generate a unique order of design history when design histories are inconsistent at cooperative sites.
5. Feature semantic network: The feature semantic network is of great significance because it connects the above mentioned two generators. The details of the feature semantic network are given in Section 5.

4. Modeling design intents

A 3D product constructed in a feature-based collaborative 3D CAD environment is a complex artefact constructed by multiple designers who adds/deletes various types of features incrementally, iteratively and the most important, concurrently. It is represented using B-Rep and its shape is formed by a large quantity of different topological entities such as topological faces, edges and vertexes. Boolean operations such as union, substraction and intersection provide an efficient way to combine features or solids together so as to re-evaluate the boundary representation. Within a collaborative designing environment, when a

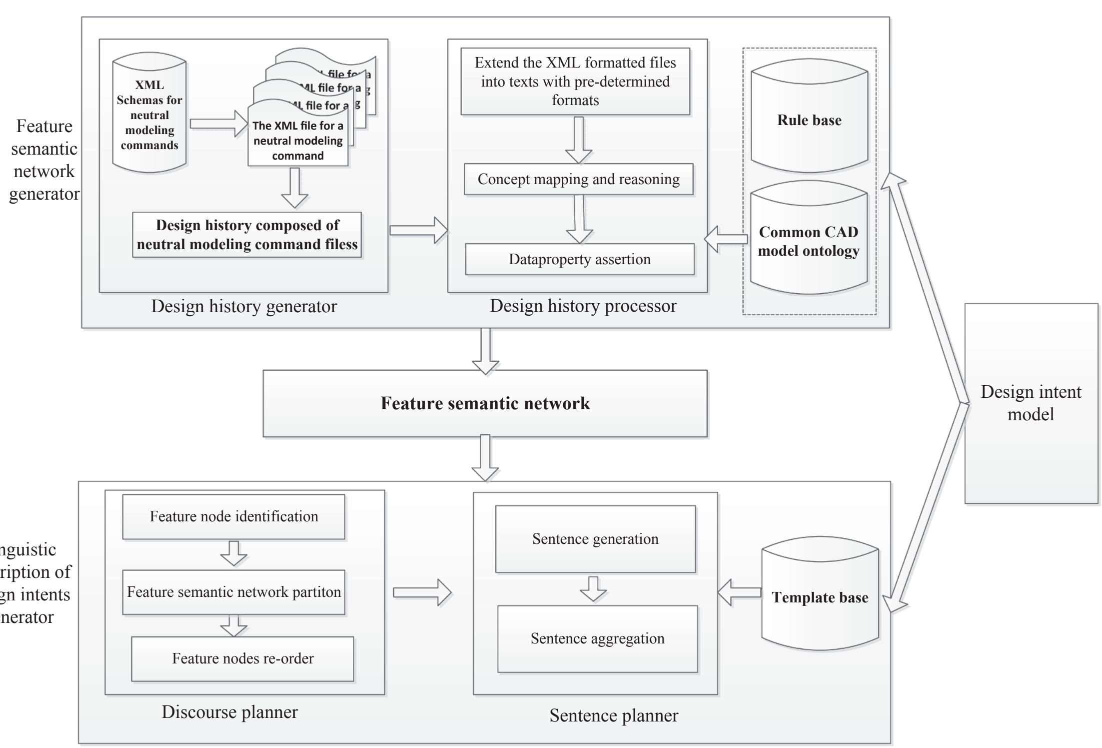

Fig. 1. A system overview of the proposed approach.

complex artefact is constructed in a collaborative manner, each cooperative site surely retains a long historical record of design operations. How can you acquire the important information easily and quickly when the design history is just an arbitrary mixture of symbolic and parameterized commands from multiple sites?

Design intent is a special kind of design knowledge. Iyer and Mills [80] defined design intent contained in legacy CAD as "*the insight into the design variables (design objectives, constraints, alternatives, evolution, guidelines, manufacturing instructions and standards) implicit in the structural, semantic and practical relationships between the geometric, material, dimensional and textual entities present in the CAD representation*". Observing from this widely accepted definition on design intents, we conclude that design intent encompasses a broader context surrounding the product development process. In an environment when artifacts are designed collaboratively at the feature-level and When participants work in parallel to gather their proficiency, knowledge and decisions so products are developed efficiently either by clean sheet design or by model re-use, we define the "design intent" in this background as "a description of the collaboration process, a coordination procedure of design intents from every participant and an optimal design solution as a result as a collective intelligence".

Capturing design intent dedicates in revealing information about modeling decisions, why they have been made, as well as relationships or dependencies that may either link decisions to part of the product representation or to other decisions. In CPM for an artefact, the key object in product design, is an aggregation of three objects representing the artifacts three principal aspects:

- *form* which represents the proposed design solution, namely the physical characteristics, for the design problem specified by the function. In a feature-based 3D collaborative designing environment, *form* corresponds to various form features, such as extrude, revolve, pocket, etc.
- *function* which describes what the artifact is supposed to do to satisfy customer needs and/or engineering requirements. The term function is often used synonymously with the term intended behavior.
- *behavior* which describes how the artifacts form implements its function.

It can be concluded that the *form* of a product determines the functionality a product have and how it behaves. In a feature-based 3D collaborative designing environment, when form features are added incrementally, iteratively and concurrently, the task of design intent capturing process is to record how the form of a product is generated by different participants and why the form is generated in the way it is. We consider an artefact as the coordination result of design intents from the multiple designers. As an on-going research, we are primarily concerned with capturing how different design intents are coordinated and what the eventual design intent is. We satisfy the purpose by creating an information model to describe the designer intents in the artifact developing process. The following concepts are used to describe our design intent model:

– Artefact: The terminology of an artifact in the domain of mechanical design.
– Design history: the purpose of design history is to describe the operations issued by each participant, how these operations are executed at each collaborative site and whether a designer's operation is modified by other designers.
– Participants: use a set of attributes to describe every designer who participated the collaborative designing process, which includes IDs, names, proficiency, design history and etc.
– Artefact structure: in a typical collaborative design scenario, an artefact could be modified over and over again by addressing undo, redo, or by changing parameters of an existing feature. Only a small set of all performed operations are responsible for building the eventual artefact. Then, we define the meaningful design operation sequence as a set of operations with minimum total number needed for constructing an artefact. By mining a meaningful design operation sequence associated to a given artefact, the coordination result of multiple design intents could be clearly represented. It is also helpful for new learners to rapidly acquire knowledge as to how to build.
– Boundary representation: it refers to the eventual boundary representation of the artefact.

The design intent information model explicitly represents with a structural manner the once abstract and hard-to-describe design intents implied in an CAD artefact and in a collaborative designing process. It discovers and extracts valuable design knowledge that is important for the future utilization of a complex artefact. Most important of all, it plays the role of guidance to help with determining the content needs to be included in the linguistic description.

5. Developing the CAD model ontology

The point of view with which we will look at a Feature Semantic Network is: how to model the collaborative feature modeling process and the composition features of an artefact using knowledge representation languages of an automatic system? Therefore, a very important part of our work is to identify the key concepts and relations in the domain of feature-based collaborative modeling by considering the epistemological and ergonomic adequacy carefully, which means a common CAD model ontology should be built at first. An ontology is a formal explicit description of concepts in a domain of discourse, properties of each concept describing various features and attributes of the concept, and restrictions on properties that can be utilized to label both nodes and links in the semantic network. The purpose of the ontology created in this paper is not only for semantic interoperation so an artefact can be reused in a different CAD environment, but also to create a strong foundation of requirements necessary to represent the design intents hidden in the collaborative modeling procedure other than create an exhaustive list of requirements necessary for every application.

5.1. Objectives and requirements

An ontology, in general, is not only a "representation of a shared conceptualisation" of a specific domain, it also envisages the use of descriptions of resources connected into "semantic networks". Ontologies are used to describe constraints on the meanings of terms used in these networks and to provide a basis for interoperability. It provides a shared and common vocabulary, including important concepts, properties and their definitions, and constraints, sometimes referred to as background assumptions regarding the intended meaning of the vocabulary. Ontologies are specified in (for many humans) complex formal languages, and knowledge engineers, domain experts and also casual users need good ways to compare ontologies and understand how to use them. As the logical structure of ontologies becomes richer, so it becomes increasingly hard to devise appropriate graphical means of presentation that do not require special training on the part of the users. In this scenario, presentation in natural language is becoming increasingly attractive. Following, we are going present an common CAD ontology for describing the feature-based collaborative product developing process and then a detailed description of the semantic network for presenting multiple design intents, generating the natural language description of a collaboratively developed product is introduced.

There are several requirements for us to establish the ontology in the feature-based collaborative CAD domain. The first one concerns with facilitating an over-all descriptor of the collaborative designing procedure of an artefact. However, there should not be any unnecessary or useless definitions as well. Besides, the ontology is used both to represent domain knowledge, and to support linguistic processing for the reason that a feature semantic network serves as the input to the language generator in terms of natural language generation. Generating linguistic output from this originally non-linguistic input requires instantiations of the conceptual unit content, i.e. concepts, properties and instances. Verbalization of properties together with lexicalization of concepts and instances are the core component with regard to the semantic statements based natural language generation. To enable effective human intervention, the linguistic description of the multi-user design intents should respect the following requirements:

- The terms used to describe design features should be understood by users of heterogeneous CAD software packages without any semantic uncertainty and ambiguity. Different CAD applications have their taxonomies and definitions over features which means a neutral and widely recognized and accepted terminologies for features is to be utilized in building the ontology.
- The description should cover all the aspects used to describe design intents as mentioned above. This asks for certain concepts, properties and relations to be established in the ontology we are going to build.
- When it comes to describe a specific feature, all the necessary information in defining the feature should be included as well. Otherwise, misunderstanding will be caused.

5.2. Capturing the ontology

5.2.1. Concepts

In order to specify the meaning or semantics for the lexicalization of nodes and links in a Feature Semantic Network for further, we take the priority of refining and extending existing sources for the CAD domain ontologies by referring to international industrial standards STEP, multiple CAD domain ontologies defined in literatures, the Core Product Model and general upper ontology WordNet.

The concepts are defined to describe a design feature in terms of the collaboration and various parameter values at its definition. Main upper concepts include:

1. Feature concept. The feature concept acts as a general place holder.
2. SketchPoints concept. It is an external concept defined to describe the critical points and parameters that form the shapes of 2D-sketches and pathes.
3. Sketch concept. The instance nodes of concept sketch link to the instance nodes of external concept, Points, to specify the various types of 2D sketches.
4. AxisandPath concept. The concept is to present a fundamental and mutual description over various types of paths or axis for sweeping, extruding and rotating the 2D-sketches.
5. Solid concept. The solid concept aims at describing the critical information for creating volumes such as cuboid, sphere, cylinder and etc.
6. Topology concept. The topology concept is to define the topological structure of an artefact. It helps to connect features within a feature semantic network.
7. ReferenceInfo concept. The ReferenceInfo concept pertains to information that is necessary to a feature's definition, but is defined externally outside a feature.
8. Metadata concept. The metadata concept is to describe the critical collaboration information concerning to a specific feature.

Finally, the major class hierarchy is shown in Fig. 2, where the arrows illustrate "is-a" relationships. Sibling classes are disjoint with one another and mutually exclusive.

5.2.2. Properties

The definition of relations is also of significant importance for the generation of linguistic description of two concepts it links lies in that the types of relationships can be directly verbalized. This requires the property hierarchy describe the relations especially between sub-classes as specific as possible. It can be observed that the upper-level feature concept is connected to the other upper-level classes with object properties mostly inherited from the "has-a" relationship, which covers the "has-reference-attribute," the "has-parameter-attribute," the "has-sketch-points," the "has-boolean-operation," and the "has-Axis-Path" relationships. One exception is that the *Feature* class is connected to the *Part* class with the "part-of" relation. Each concept connects to the feature concept with a "has-a" relation is intuitively a descriptor for representing a specific characteristic of a feature. Each subclass is also connected to another subclass through object property inherited from the 'has-a' relationship. Table 2 presents us with a quick view of properties linking a pair of subclasses.

A Datatype property which links an instance to a specific value of integer, float, string, or Boolean value are to define parameter attributes. They are important because dimensions of each feature are controlled by a set of parameters. Additionally, parameters are recorded to make it easier to integrate quantitative design knowledge into the model such that it is possible to change the product model with alterations of values [78]. An example of a flange coupling and its Feature Semantic Network are exemplified in Figs. 3 and 4 perspectively.

6. Generating linguistic description of design intents in feature based collaborative design using feature semantics

6.1. Developing Feature Semantic Network

6.1.1. What is a Feature Semantic Network?

Natural language has developed good ways of conveying complex logical structures and requires no special training. A semantic network is definitely a good solution for NLG. A semantic network is a vision of the web of the future consisting of resources with machine-readable semantics. It is a graphic notation for representing concepts, represented by nodes and the corresponding instances which are both represented by nodes and relations which are represented by links (arcs or arrows) between instances. Its definition is presented below:

Definition 1 (*Feature Semantic Network, FSN*). A Feature Semantic Network (SN) is a directed, node-edge-labeled graph $G = \{N, \phi_N, E, L_E, \ \phi_E, \ \psi\}$, where:

- $N = n_1, n_2, \ldots, n_m$ is a set of labeled nodes, the label of a node n is denoted as l(n), ϕ_N: $N \rightarrow L_N$ is the label function that mapping each node n to its label.
- $E = e_1, e_2, \ldots, e_m$ is a set of labeled edges, L_N is a label set of E, the label of an edge e is denoted as l(e), ϕ_E: $E \rightarrow L_E$ is the label function that mapping each edge e to its label. Each node or edge is called an element of a Feature Semantic Network.
- ψ is the relationship mapping from edges to nodes ψ: $E \rightarrow V \times V$. Each edge is directed. Its starts from a source node, denoted as S(e), and points to a target node, denoted as T(e).

It is very important to set the labels for edges and nodes of a Feature Semantic Network because the elements or objects of a semantic network for representing discourse meanings are concepts and relations. We label each edge with a semantic relation based on the object property we defined between instances and objects and the datatype properties we defined between an instance and its attributes. The label enables a linguistic generator understand the relations between a node couple. A concept node is labeled with a lexicon equivalent to its conceptual meaning. The nodes constitute a Feature Semantic Network fall into three categories: the concept nodes, the instance nodes and the value nodes. The labels of instance nodes varies according to the node

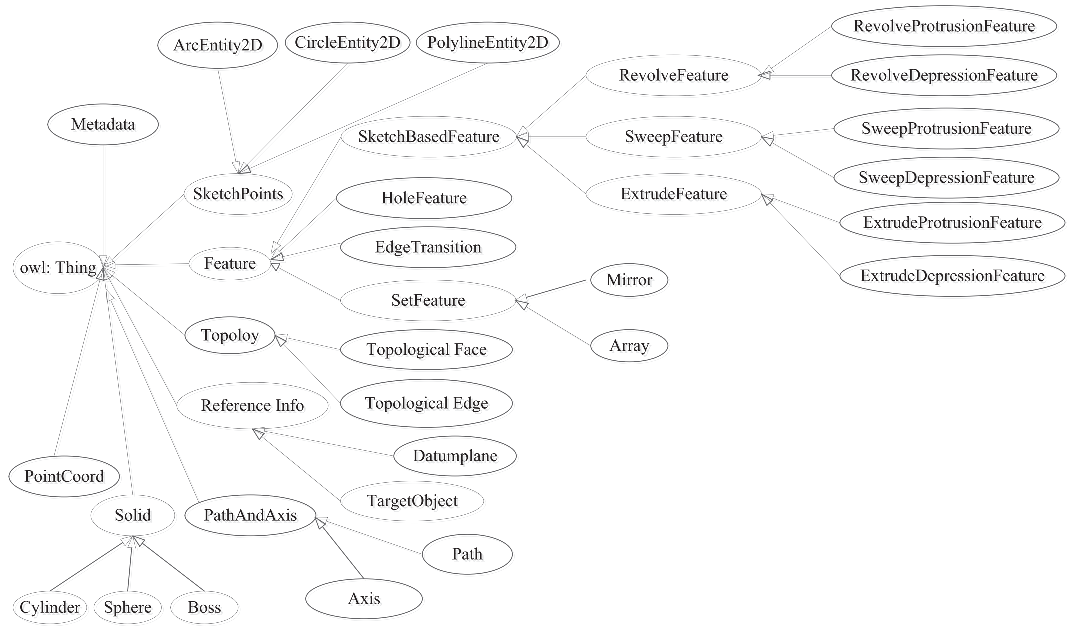

Fig. 2. A brief preview of the concept hierarchy of the common CAD ontology.

Table 2
Object property hierarchy in the feature-based collaborative CAD ontology.

Parent Property	Child Properties	Domain	Range
hasBooleanOperation	hasAdditionOperation	Feature	Feature
hasSketchPoints	hasPolylineEntity2D	Sketch, SweepPath	PolylineEntity2D
hasPolylinePoints	–	PolylineEntity2D	LocalPoint2D
hasAxisandPath	hasExtrusionPath	Feature	LocalNormal, SketchPoints
hasAxisandPath	hasSweepPath	Feature	LocalNormal, SketchPoints
hasTopology	hasTopologicalFace	Feature	TopologicalFace
hasReferenceTarget	hasTargetFeature	SetFeature	SketchBasedFeature, HoleFeature
hasReferenceTarget	hasDatumPlane	Feature	TopologicalFace
hasMetadata	hasCreator	Metadata	String
hasLocalCoord	hasXposition	LocalPoint2D	Double
hasLocalNormal	hasXposition	LocalPoint2D	Double
hasShape	–	Sketch	String
hasUniqueName	hasDatumPlaneName	DatumPlane	String
LocateOn	–	Feature	Topologicalface

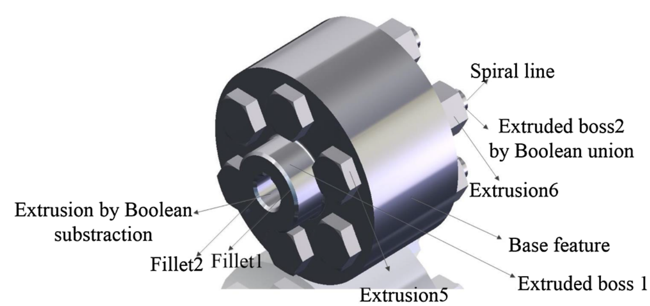

Fig. 3. An example of a flange coupling.

types. There are seven types of concept nodes are distinguished, which are feature nodes N_f, metadata nodes N_m, reference information nodes N_r, parameter attribute nodes N_{para}, topology nodes N_t, *sketchnodes*N_s and path nodes N_{path}. Features, topological entities together with paths and sketches are entities that are given unique identities by any CAD package. Thus, the labels of feature nodes N_fs, topology nodes N_ts, *sketchnodes*N_ss and path nodes N_{path}s all inherit their unique identities perspectively. We paid more attention to the labels of each value nodes. The value nodes represent a variety of parameters at a feature instance's definition. A value node is linked to an instance node through a datatype property, e.g. haslength, hasradius. The label of a value node can be a float to present, e.g., the length, width and height of a cuboid. The data type string is also employed, for example, to represent the state vector of a feature in the form of "01100111" because the array data type is not supported by ontology manipulating language. Another example of the string format attribute value presentation is the norm of a topological face for the same reason.

6.1.2. Relation reasoning

Part of the rules to reason the *hasReferenceInfo* and *hasBooleanOperation* relations are presented in Table 3. In reasoning the object properties derived from the *hasReferenceInfo* relation, a *ReferenceInfo* instance node is temporarily inserted when there existence a *Topology* instance node which has an equivalent unique name.

Although there is not an explicit object property to specify the dependency relationship between two features, the relationship can be inferred by the following lemmas.

Lemma 1. *Two feature concept nodes, noted as FCN_a and FCN_b, FCN_a is considered to be dependent on FCN_b, noted as $FCN_a \overset{D}{\rightarrow} FCN_b$ if one of the following conditions are satisfied:*

1. *There exits a ReferenceInfo node RIN_c and topology node TPN_d that*

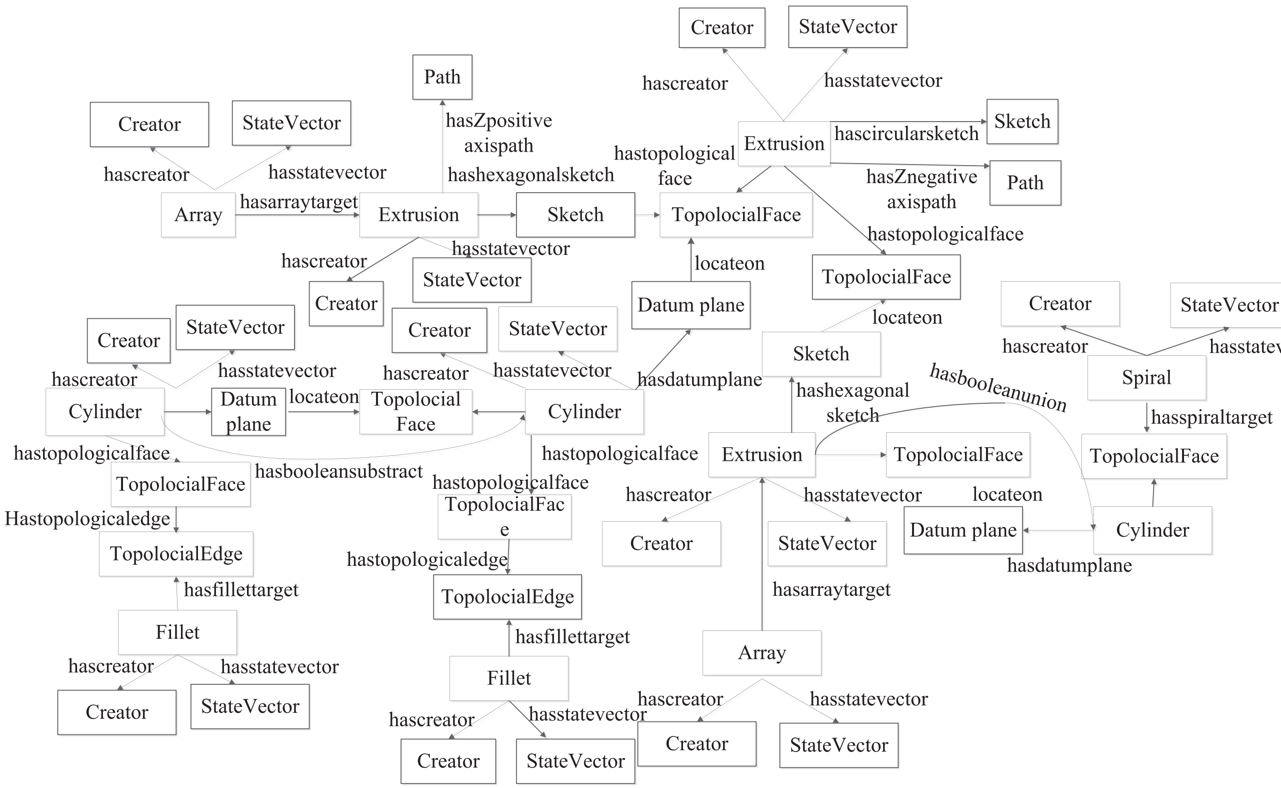

Fig. 4. An example of the semantic network description of a flange coupling.

satisfy $FCN_a \xrightarrow{hasReferenceObject} TPN_d$, $RIN_c \xrightarrow{locateon} TPN_d$, $FCN_b \xrightarrow{hasTopology} TPN_d$.

2. *There exists a SketchPoints node* SN_c *and topology node* TPN_d *that satisfy* $FCN_a \xrightarrow{hasSketch} SN_c$, $FCN_b \xrightarrow{hasTopology} TPN_d$, $SN_c \xrightarrow{locateon} TPN_d$.
3. *There exits another feature concept node* FCN_c *that satisfy* $FCN_a \xrightarrow{F} CN_c$ *and* $FCN_c \xrightarrow{F} CN_b$.

Lemma 2. *Given two operations* O_a *and* O_b, N_a *and* N_b *are feature concept nodes from FSN representing the features created by* O_a *and* O_b *respectively, it can be asserted that* O_b *has attribute dependency on* O_a *if* $N_a \xrightarrow{D} N_b$.

The dependency among features can be further categorized into the following two types according to in which manner the dependencies among features are established:

– Reference dependency. If the topological entities created by an operation, say O_a, are referenced by a later generated operation, say O_b, it can be asserted that O_b has reference dependency on O_a. The reference dependency decides if a feature can be successfully added or if the added feature preserves designers' intentions. For example, when O_a and O_b are concurrently issued while the parts at both operations' generation are identical, O_a adds a hole on face *f* and O_b select *f* to chamfer,
– Attribute dependency. Attribute dependency is established if certain parameters for the definition of a later generated operation, say O_c, are affected by the dimension or location of a previously created operation, say O_d.

Manifesting these two relationships is the task of significance in the situation of capturing design intents for reusing a CAD artefact.

6.1.3. Feature Semantic Star

Within a Feature Semantic Network, the relationships of between a single node with other nodes form a star topology called a semantic star. Its definition is given below:

Definition 2 (*Semantic star*). G = $\{N, E\}$ is a semantic network, n ∈ N is a node of G, E′ ⊂ E(n) is a subset of edges connected with n. n and all nodes of other end of each e E from a node set V. Then G = $\{V, E\}$ is a feature semantic network called a semantic star, where: n is the central node of a semantic star, other nodes connected to n are margin nodes, each edge e ∈ E owns a link type specifies a property of n, and each margin node satisfies v = other–end(e,v) maintains the attribute value.

There are whole a lot of semantic stars within a Feature Semantic Network to represent the collaborative modeling procedure from different perspectives according to the above definition. One example is, when a topological face f is selected as the datum plane for multiple features, the reference attribute node N_f representing f and all feature nodes connected to f by has-reference-attribute relation constitutes a semantic star. The star topology clearly represents the set of features selecting f as the datum plane. Another example is, given a concept node *Extrusion*, it has an "is-instance-ofr¨elation with all feature nodes connecting to it. These nodes also compose a semantic star. Nevertheless, there is one kind of semantic stars really attract our attention which we call Feature Semantic Star.

Definition 3 (*Feature Semantic Star*). G = $\{N, E\}$ is a feature semantic network, n ∈ N is a node of G, E′ ⊂ CES(n) is a subset of edges connected with n. n and all nodes of other end of each e ∈ E from a node set V. Then G = $\{N, E\}$ is a feature semantic network called a feature semantic star, where: the central node n of G is in the type of feature concept node.

6.2. Generating linguistic descriptions of design intents

6.2.1. A detailed description of the NLG process for design intent representation

NLG, which is concerning with transforming a given formal content input into a natural language output, has provided us with a useful tool to serve the purpose of generating easily readable and understandable

Table 3
SWRL rules defined for reasoning.

Property	Rule
hasDatumPlane	feature(?f1) ∧ hasTopolocialFace(?f1,?t1) ∧ hasUniqueName(?t1,?n1) ∧ feature(?f2) ∧ hasDatumPlane(?f2,?t2) ∧ hasUniqueName(?t2,?n2) ∧ swrlb:equal(?n1,?n2) → hasDatumPlane (f2,t1)
LocateOn	feature(?f1) ∧ hasTopologicalFace(?f1,?t1) ∧ feature(?f2) ∧ hasDatumPlane(f2,t1) → LocateOn(f2,t1)
hasBooleanUnionOperation	SketchBasedFeature(?f) ∧ hasLocalNormal(?f,?n) ∧ hasExtrudePath(?f,?p) ∧ hasDirection(?p,?d) ∧ swrlb:notEqual (?n,?d) → hasBooleanSubstractionOperation(?f2,?t1)

plain texts for presenting design intents. In this paper, the process takes the Feature Semantic Network as the input and executes analysis over the input data to generate the output description. We adopted the six activities architecture proposed by Reiter and Dale [77] for natural language generation. The initial activity of the pipeline is content determination to determine which information should be mentioned in the text by filtering and summarizing the system input data, encapsulate them into a formal language. We skip this step because the by constructing the feature semantic network, we have already filtering out the content that should appear in the output textual description. Finally, a two-stage natural language generation process in this paper is summarized in the following process.

– Discourse planning: the process to determine the order and structure of the output text.
– Sentence planning: this is a general process that combines the sentence aggregation, lexicalization and referring expression generation processes. Sentence aggregation is a process groups several messages together in a sentence. Lexicalization is a process to decide which words and specific expressions must be used to express the concepts and relationships of the domain that appear in the design operations. And, linguistic realization with purpose to apply grammatical rules to produce a text which is syntactically, morphologically and orthographically correct.

6.2.2. Discourse planning

A good structure makes a text much easier to read. The task of discourse planning is to determine the given order of feature descriptions appearing in the text automatically. The task can be divided into two steps. On one hand, we should fix the appearance orders of descriptions about concurrently generated features in the output text. One cannot expect to use the design sequence from a random site, because the design histories are inconsistent at cooperative sites. However, we can reach this goal by confirming the priorities of features. Within the Feature Semantic Network, each feature node is linked to a metadata node. The datatype property StateVector of the metadata node is employed to confirm the priorities for features. A state vector, proposed by Lamport, is with the motivation of detecting concurrency among operations. It is in the form of an N-tuple, where N is the number of cooperative sites involved. Integers start from 1 to identify the participated sites. For each newly joined site, its identifier is creased by 1 based on the largest integral identifier of the current collaborative designing group. An element of a state vector, initiating from 0, presents the number of features already created by that site. For example, a state vector $\langle 1, 0, 0 \rangle$ attached to a feature means that at the feature's generation moment. Given a newly generated feature f and its related metadata node m, its timestamp is generated according to the following equation:

$$f.\ timestamp = m.\ SiteID + \sum_{i=1}^{N} StateVector[i] * Identifier_i \tag{1}$$

The priority between concurrent features is determined given Definition 4.

Definition 4. Given two concurrent operations O_a and O_b and the generated features f_a and f_b, f_a is prior to f_b iff the following condition is satisfied f_a.timestamp $\leqslant f_b$.timestamp.

Obviously, a feature with higher priority should appear in the output text before its concurrent generated feature, as it is illustrated in Fig. 5-(a).

On the other hand, to reveal the dependencies among features and sub-configurations of an artefact, features with dependency relationship should be combined into groups for representation. This is accomplished by considering the time stamp of a feature and the dependency relationship both.

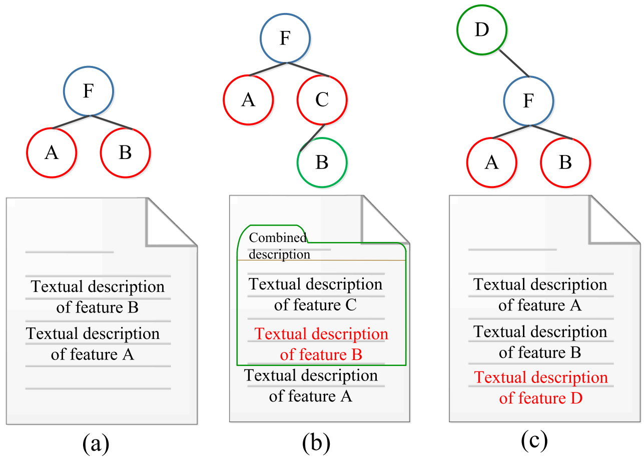

Fig. 5. An explanation of discourse planning process.

Definition 5. Given two operations O_a and O_b and the generated features f_a and f_b where f_a.timestamp $\leqslant f_b$.timestamp, f_b is prior to f_a iff there exists another feature f_c where f_c is prior to f_a and $O_b \xrightarrow{D} O_c$.

Fig. 5-(b) illustrated the appearance of the description of feature with larger time stamp but higher priority. Descriptions of sequentially generated features are appended at the of the current text output, as illustrated in Fig. 5-(c).

6.2.3. Sentence planning

In the phase of sentence planning, several questions are proposed firstly, specifically:

- Which nodes from a feature semantic star should be selected for linguistic representation?
- Which order of these nodes is well formed?

With the above considerations, the sentence planning task can be decomposed into the following two sub-tasks:

- To resolve the first problem, a decomposition of a feature semantic star is performed. A feature semantic star covers nearly all the information concerning to a feature, e.g. parameters, topology and metadata. Should all these information be included for the linguistic representation? Or, is there any possibility to contain all the information in one sentence while the readability can still be promised? Therefore, the task of this step is to decompose the star structure into several parts, by eliminating the parts that are unrelated to the feature generation description, it is suitable to generate phrases or sentences for the expression of a specific characteristic.
- As to the second problem, a sorting over the decomposed sub-structures is due to be performed. This work contains two steps. The first one is to assort the decomposed sub-structure-based feature descriptors into a proper order so as to clearly present the feature generation procedure linguistically. The second one is to properly order the nodes composing a decomposed sub-structure for generating sentences specifically.

All language elements of a well-formed piece of language form a tree structure. Basically, a discourse is composed by several logically organized sentences, sentences are formed by phrases, phrases are formed by words. All these words, phrases, sentences, discourses including punctuation are elements of natural language. A feature semantic star can be viewed as a group of feature semantic trees that connects to the central feature node $CN_{feature}$. The central node of a feature semantic star connects to other types of nodes by edges labeled with lexicalized semantic relation. The decomposition work is undertaken by identifying the labels of edges that starts from the feature node. An example is given in Fig. 6. A decomposed feature semantic tree can be abstracted as a hierarchical structure with four levels. The first level is the root of the tree structure which is constantly the central feature node. The second level is a node delegates one of the properties defined in our ontology. The third level are some attributes of the property. The fourth level is the value or range of the attribute. The root node is connected to the property nodes with a "hasproperty" relation that derives from the "has-a" relation, its semantical meaning is to represent the fact that the feature owns the property. There exists a 1:1 relationship between the feature node and the property nodes. A property node is connected to its value nodes with a "has attribute" relation that also derives from the "has-a" relation, its semantical meaning is to represent the different attributes that a property object owns. There exist a 1:N relationship between a object property node and its attribute nodes. Finally, every property attribute has its value, this is represented by the datatype property relation which is a 1:1 relationship. The value nodes are terminal elements of a decomposed feature semantic tree. One exception is the ReferenceInfo node connecting to the central feature node with object properties derived from the "hasReferenceObject" property. This is because for most of the time, the ReferenceInfo node points to nodes from other feature semantic starts.

Considering the fact that all nodes connected to the central node are equally treated until now, the decomposed feature semantic trees are equally treated as well. Very differently, the generation of a modeling

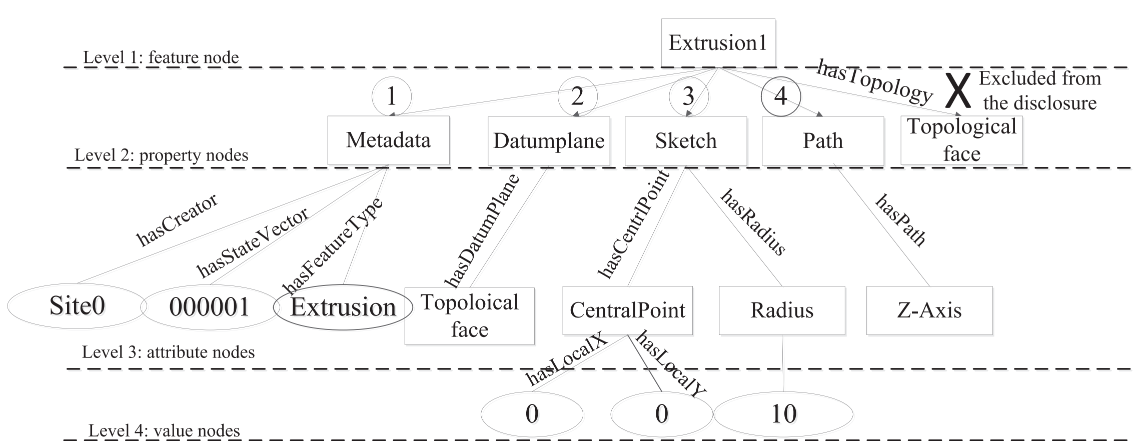

Fig. 6. An example of decomposing a feature star.

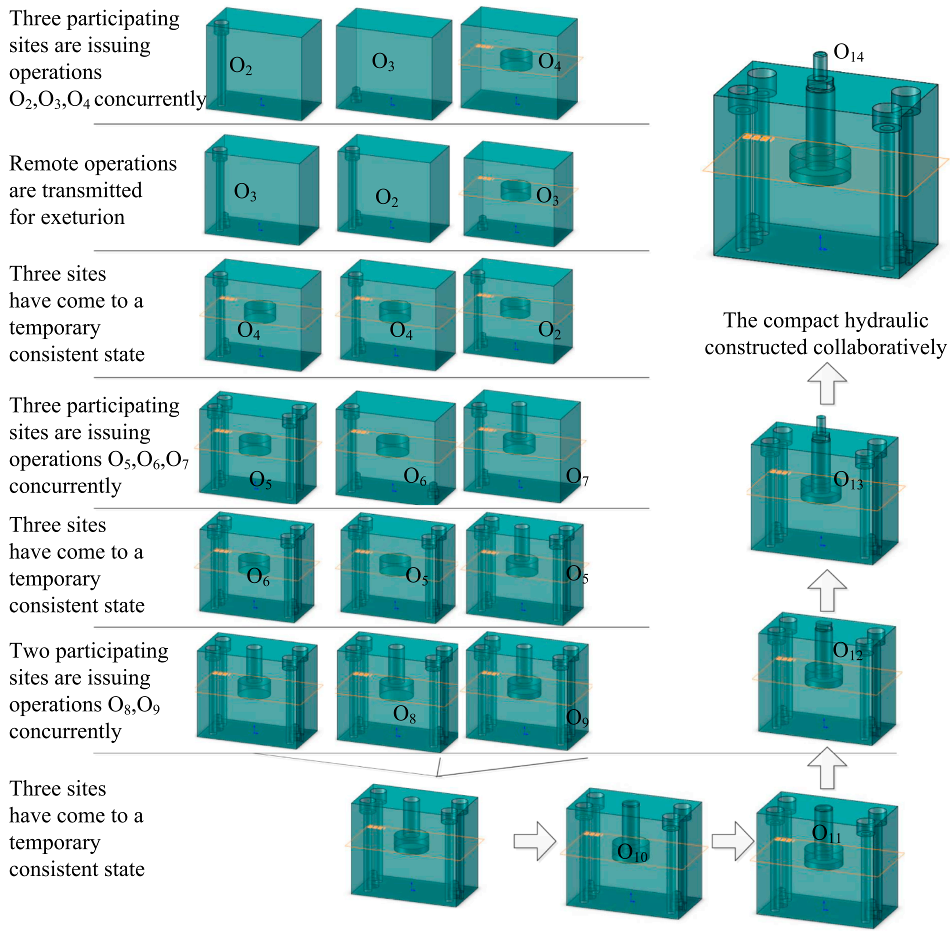

Fig. 7. The collaborative designing process of the example compact hydraulic.

feature follows a strict procedure which means when there exists a priority relationship when describing different steps of the feature generation process in the textual presentation.

- The very first step is to decide in which way a feature is to be constructed. In other words, to decide the type of the feature, e.g. the sketch-based features such extrusion or non-sketch-based features such as hole.
- As soon as the type of the feature is determined, the datum plane for locating a feature is selected. It is in fact not a separate step. For sketch-based features, this work is treated as a part of sketch drawing. For non-sketch-based features such as holes, the datum-plane is selected at the same time a volume for substraction is placed on the part. Nevertheless, with the consideration of its importance, in our work of linguistic description generation, it is a necessary work to point out the datum plane by generating a single sentence.
- For sketch-based features, 2D-sketch drawing is one essential phase. We address our emphasis on the shape of the 2D sketch and its critical coordinates.
- Path drawing is another essential phase for sketch-based features. Descriptions for pathes in *Extrusion* feature and *Sweep* feature should be presented differently.

Fortunately, the lexicalized semantic relations between the central feature node and property nodes can be an effective tool for determining the sentence sequence that a feature semantic tree should appear in the text, as it is also illustrated in Fig. 6.

Therefore, the challenge remains for sentence generation is generating a readable textual description that expresses the relevant information of the creation of each design feature. The essential step is automatic lexicalization of object properties. Lexicalization of concepts and relations are fundamental to generate the linguistic output of a feature semantic tree. One advantage of our feature semantic network is that some concepts and relations can be directly represented by language words or phrases. Therefore, the work remains to do is lexicalization of the a property instance node and the data type links between its value nodes. The template-based sentence generation is adopted. We create different templates for representing the relationships implied by the hierarchy. This achieved by including the terminology of the operation, key attributes, conflicts, designers using specific templates. The basic template for describing common information for all types of feature modeling operations is given below, while the templates for describing dimensions varies given semantics of different operations.

Still, when generating the sub-sentence for a datum plane which is also the surface of some existing feature, the most critical issue is determine the proper word to describe the datum plane. The lexicons from the set {*TOP*, *BOTTOM*, *LEFT*, *RIGHT*, *FRONT*, *BACK*, *SIDE*} are the options for us to choose. Considering a part will be orbiting according to a designer's action, when using the word from above set for datum plane description, the designers are required to stay with the isometric view.

7. Experiment

In this section, we are going to illustrate the workflow of transforming a 3D artefact into natural language and its advantage in fostering design intent communication. The experiment is undertaken within the prototype we developed using C++, Protege 4.3, Java with OWL API and the CAD modeling system API, a Solidworks 2014 based collaborative designing prototype. Protege 4.3 is to construct the common CAD ontology. For the ontology visualization, NavigOwl plugin for Protege is adopted. Java with OWL API is to help build up the Feature Semantic Network for specific CAD models using both the ontology and the feature descriptions by extending their associating XML-based neutral modeling commands. The Solidworks 2014 based collaborative designing prototype contributes considering the following 3 aspects: firstly, it allows for the co-designing of CAD models; secondly, as soon as a designer issues a feature modeling command, the Solidworks APIs are called to generate XML-based neutral modeling commands constrained by XML schemas regulated by the CAD model ontology. These commands are then sent to other cooperative sites for execution; and thirdly, it helps to provide an overall description by acquiring the topological information for every feature. The experiment is carried out with the following 3 stages:

Collaborative designing. An example part, a Compact Hydraulic, as shown in Fig. 7 is constructed by designers from 3 sites within our Solidworks based collaborative designing environment. Designers can login the system after a simple registration. The participating sites are identified as *Site1*, *Site2* and *Site3* respectively. Totally, there are 14 feature modeling operations issued and details concerning to the concurrency scenario and the effect of each modeling feature are presented in Fig. 8. In Table 4, the concurrent and dependent relations between operations, denoted by $\|\|$ and $\xrightarrow{D}$ separately, the State Vectors, and the calculated priorities are given.

Feature semantic network generation. If a co-designer is motivated to check the textual description of the collaboratively developed model, it is necessary to choose the modeling history record, always represented in several XML files contained in the same folder, at the local site to incorporate features, attributes and relations into the Feature Semantic Network. We use Dom4j API to clarify the hierarchy of the XML files and obtain information from nodes at different levels to make assertions of individual and properties. The instantiated OWL file in its RDF/OWL format of the Compact Hydraulic is shown in Fig. 9 and the visualized Feature Semantic Network is shown in Fig. 10. For the sake of exemplify a Feature Semantic Network is composed of a series of Feature Semantic Stars, an area marked with the red lines and zoomed in.

Natural language generation. This is the last step of the work flow. Fig. 10 presents the textual descriptions of the example part. With the proposed approach, it can be guaranteed that descriptions generated at dispersed sites are identical. The output text is shown in Fig. 10 below.

8. Discussion

A method to generate linguistic description to foster effective communication of design intents through collaboratively developed CAD models is proposed in this paper. The contributions lie in the following aspects: (1) the linguistic description is generated automatically. Human intervention is needed as a connector to import the output of one software, e.g. to import the feature descriptors by extending the XML formatted advanced modeling commands generated in the SolidWorks, into another software manually, e.g. Protege, to generate the Feature Semantic Network of a CAD model; (2) it presents features, parameters and the topological entities of a CAD artefact; and (3) it reveals the implicit relations among modeling features and combines features with interdependencies automatically so a model is robust for e.g. re-use. Explicit representation of these knowledge is important because a lot of questions could be asked for designers when creating CAD models [76]. For example, which sketch plane to use, what kind of complexity should it be, what references need to be used to create constraints, when to apply theBooleanoperations and which one of Boolean operation should be used, how to choose the sequence of the modeling, etc. These questions are tricky because the answers are the keys for the varieties of ways to create a geometrical model in CAD (see Table 5).

However, as an ongoing research on design intent communication, the method has limitations considering the following aspects:

At first, from the perspective of the CPM model, the method is limited in expressing the functions and behaviors of an artefact. The relations between the form features and functional features are not established yet. It addresses part of the know-what issue, e.g. what the artefact is, and know-how issue, e.g. how it is constructed or how the conflicts between concurrent operations are resolved. Knowledge addressing the “why” issues is not presented, e.g. why the conflicts are resolved in such a way and why the artefact is constructed in such a manner. Therefore, the method is inadequate in presenting the descriptions in the form of “to do” or “to prevent”. As it is point out by Cheng et al. [79] that functional considerations of design have multiple sources, most well-known of which include customer requirements and

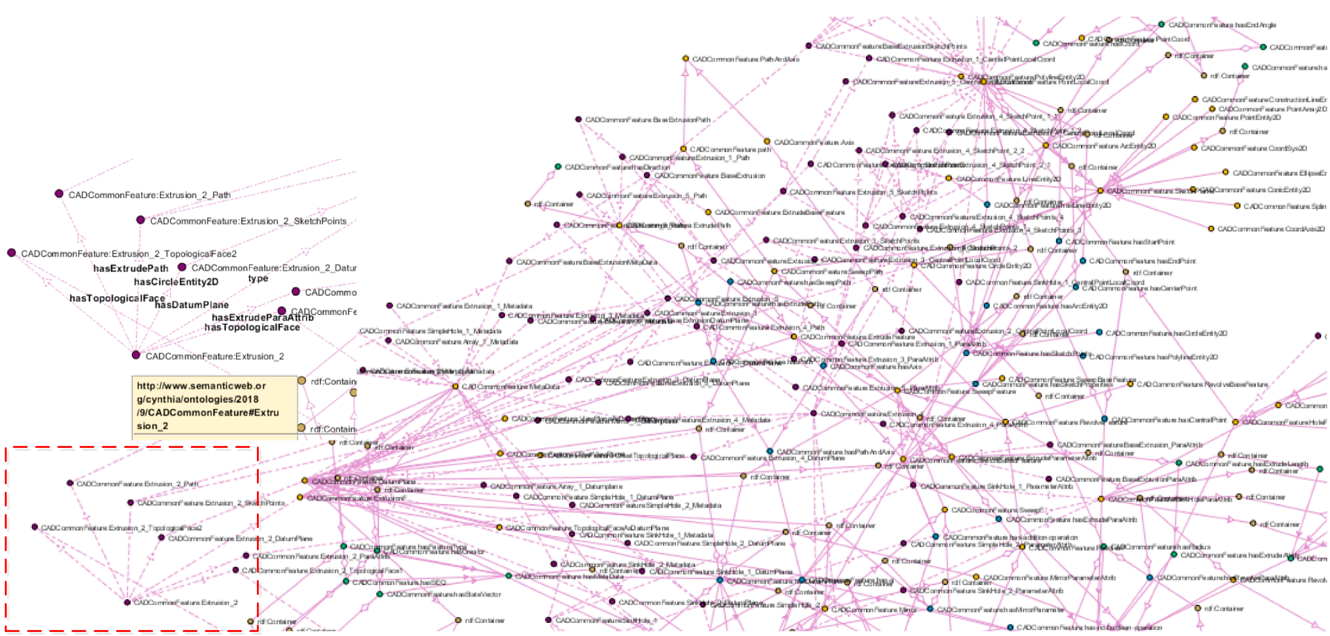

Fig. 8. The visualized Feature Semantic Network for the example compact hydraulic.

Table 4
The templates.

Procedure	Content	Template
Linguistic description of feature creator	To present the creator of a certain feature	A/An [LEXICALIZED FEATURE NODE LABEL] is created by [CREATOR_IDENTITY]
Linguistic description of datum plane	To specify the datum plane a feature locates on	Alternative 1:The [LEXICALIZED DATUMPLANE NODE LABEL] is created as the datum plane
Linguistic description of sketches	To present the type of the 2-D sketch together with the key points or angles in the datum plane's local coordinate with which information a sketch can be re-created manually	A [LEXICALIZED SKETCH NODE LABEL] is created with [THE KEY POINTS OF THE SKETCH] in the local coordinate of the datum plane
Linguistic description of pathes	To present to key points or angles with which information a path can be re-created or identified manually	A [LEXICALIZED PATH NODE LABEL] is created with [THE KEY POINTS OF THE PATH]
Linguistic description of constructing procedure of sketch-based feature	To state how the form of a feature is determined by using a sketch and a path	The sketch [REVOLVES/SWEEPS/EXTRUDES] along the direction [IDENTITY OF THE PATH] for [RADIUS/DISTANCE]
Linguistic description of a solid	To present the type of a solid (generally the fundamental system defined volumes) and its dimensional parameters	A [LEXICALIZED SOLID NODE LABEL] is created with [LOCATION PARAMETERS]in the local coordinate of the datum plane with [DIMENSIONAL PARAMETERS]
Linguistic description of constructing procedure of boolean operation	To state how a part is changed by using a boolean operation	The [TYPE OF THE SOLID] is [SUBTRACTED FROM/UNITED WITH/DIFFERENTIATED WITH] the current part

engineering considerations. Descriptive words cannot be generated simply because a feature is a Hole or a Groove because features owning the same form, e.g. a Groove or a Hole, have different functions and behaviors when composing into different types of artifacts. One solution to facilitate automatic generation of function descriptions is to establish a knowledge base and use knowledge retrieval to map the feature/sub-configuration under description to some existing functional features of the same type. In addition, the researches in the domain of CPS give us clues as how to associate a sub-configuration or an artefact to the real engineering phenomena so as to describe the behavior.

Secondly, the method has not been able to serve a well-experienced communication because we have not yet designed an integrated system yet. Accomplishing the design intent communication activities in this paper entails using different independent software. In literature [67], according to the author's experience, designers are typically reluctant to accept and add new applications for archiving and annotation of design knowledge to their current portfolio of software. Ideally, the KBS capabilities regarding knowledge capture, knowledge representation and reasoning are to be merged with CAX software to provide engineers with automated assistance in geometry manipulation, data processing and analysis. Moreover, user friendly interface is needed to support intuitive human-computer interaction (HCI) to enhance the whole knowledge management process. Fox example, an artefact should be able to interact with the text. Whenever a feature is selected, its descriptive text block can be highlighted.

9. Conclusion and future work

In order to communicate design intents embedded in collaboratively constructed products for model reuse, the paper proposes a natural language based design intent communication approach for later users of product model to gain better understandings over the model reused. The design intent communication happens between CAD software packages, that is product data exchange between heterogeneous CAD packages. The STEP standard and the technique of ontology are two representative and prominent approaches to exchange geometry information and model semantics perspectively between heterogeneous CAD packages. After a series of literature review and experimental work, we observed that the information generated during the product developing process and plays a relevant role in the course of model

Fig. 9. The instantiated OWL file in OWL + RDF.

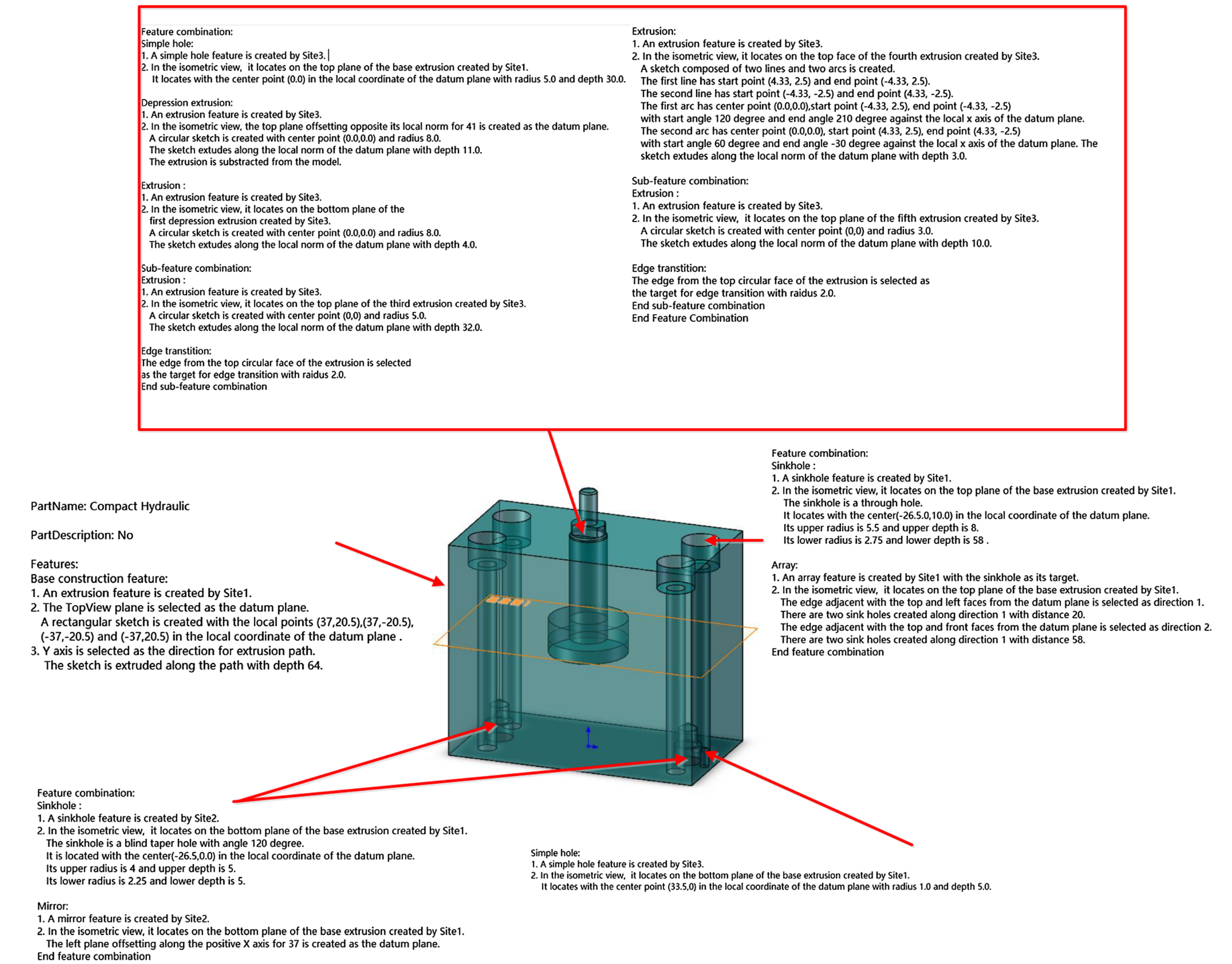

Fig. 10. The output text for the compact hydraulic.

Table 5
A description of the attributes of feature modeling operations and relations.

Operation	Concurrent relationship	State vector	Time stamp	Dependent relationship
O_1	–	000	1	
O_2	$O_2 \| \| \| O_3 \| \| \| O_4$	100	2	$f(O_2) \xrightarrow{D} f(O_1)$
O_3	$O_2 \| \| \| O_3 \| \| \| O_4$	100	3	$f(O_3) \xrightarrow{D} f(O_1)$
O_4	$O_2 \| \| \| O_3 \| \| \| O_4$	100	4	$f(O_4) \xrightarrow{D} f(O_1)$
O_5	$O_5 \| \| \| O_6 \| \| \| O_7$	111	8	$f(O_5) \xrightarrow{D} f(O_4)$
O_6	$O_5 \| \| \| O_6 \| \| \| O_7$	111	9	$f(O_6) \xrightarrow{D} f(O_2)$
O_7	$O_5 \| \| \| O_6 \| \| \| O_7$	111	10	$f(O_7) \xrightarrow{D} f(O_4)$
O_8	$O_8 \| \| \| O_9$	222	14	$f(O_8) \xrightarrow{D} f(O_1$
O_9	$O_8 \| \| \| O_9$	222	15	$f(O_9) \xrightarrow{D} f(O_4)$
O_{10}	–	233	24	$f(O_{10}) \xrightarrow{D} f(O_7)$, $f(O_{10}) \xrightarrow{D} f(O_9)$
O_{11}	–	234	25	$f(O_{11}) \xrightarrow{D} f(O_{10})$
O_{12}	–	235	26	$f(O_{12}) \xrightarrow{D} f(O_{11}$
O_{13}	–	236	27	$f(O_{13}) \xrightarrow{D} f(O_{12})$
O_{14}	–	237	28	$f(O_{14}) \xrightarrow{D} f(O_{13})$

reuse, such as construction history, feature dependency, are usually lost even using these two approaches. The design intent communication happens between humans. A product model is a medium for its later users to understand to intents of its creators. Although existing CAD packages provide feature recognition tools to extract features from the model, these tools are quite limited in re-storing the lost design intent-related design information.

To solve the problem, an design intent information model is built firstly. The model provides a structural description of design intent which is hard to be presented formally. It also plays the role of content determination so the natural language generation process would know what content to present. Secondly, a common CAD ontology is built and a semantic network so as to record not only the feature-based product developing information but also the collaboration information. An algorithm is devised to partition the semantic network and so helps to detect the dependency relationships between features and also the design history. Finally, a natural language generation approach is devised so the textual description of a collaboratively developed product can be generated. The text was presented at the feature-level according to the captured design history. In order to improve people's understanding over the product during the model reuse, the text was divided into small sections where each section is the description of features with dependency relations.

Ongoing research efforts evolves the following aspects: (1) A knowledge repository is needed where the knowledge of certain type of artefact is structured; (2) A knowledge mapping mechanism should be devised so the function of a form feature can be captured and descried; (3) An integrated system with a user friendly interface in presenting design intents in expressive language is urgently needed and is going to be developed; (4) Using advanced optimization [81,82] and multi/many-core [83,84] techniques to enhance the computational efficiency of the proposed method.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (61502353 and 61472289), the Youth Innovation Corps Fund of Humanities and Social Sciences of Wuhan University, and the Soft science research program of Hubei Innovation fund for Technology (2016ADC073). The authors would like to thank all the reviewers for their constructive comments and suggestions.

References

[1] A. Sivanathan, T. Lim, J.M. Ritchie, The application of ubiquitous multimodal synchronous data capture in CAD, Comput. Aided Des. 59 (2015) 176–191.
[2] R.D. Sriram, S. Szykman, D. Durham, Special issue on collaborative engineering, J. Comput. Inf. Sci. Eng. 6 (2) (2006) 93–95.
[3] Y.Q. Wu, F. He, D.J. Zhang, X.X. Li, Service-oriented feature-based data exchange for cloud-based design and manufacturing, IEEE Trans. Serv. Comput. 11 (2) (2018) 341–353.
[4] B.C. Kim, Y. Jeon, S. Park, Toward standardized exchange of plant 3D CAD models using ISO 15926, Comput. Aided Des. 83 (2017) 80–95.
[5] J. Stark, Product lifecycle management, Product Lifecycle Management, vol. 1, Springer, Cham, 2015, pp. 1–29.
[6] C. Bock, X.F. Zha, H. Suh, Ontological product modeling for collaborative design, Adv. Eng. Inform. 24 (4) (2010) 510–524.
[7] F. Pena-Mora, D. Sriram, R. Logcher, SHARED-DRIMS: SHARED design recommendation-intent management system, The Second Workshop Enabling Technologies: Infrastructure for Collaborative Enterprises, Second Workshop, 1993, pp. 213–221.
[8] X. Lv, F. He, W.W. Cai, Y. Cheng, Supporting selective undo of string-wise operations for collaborative editing systems, Future Gener. Comput. Syst. 82 (2018) 41–62.
[9] W.C. Regli, X. Hu, M. Atwood, W. Sun, A survey of design rationale systems: approaches, representation, capture and retrieval, Eng. Comput. 16 (2000) 209–235.
[10] J.D. Camba, M. Contero, Assessing the impact of geometric design intent annotations on parametric model alteration activities, Comput. Ind. 71 (2015) 5–45.
[11] X. Lv, F. He, Y Cheng, Y.Q. Wu, A novel CRDT-based synchronization method for real-time collaborative CAD Systems, Adv. Eng. Inform. 2018 (38) (2018) 381–391.
[12] E.M. Sanfilippo, S. Borgo, What are features? An ontology-based review of the literature, Comput. Aided Des. 80 (2016) 9–18.
[13] Y. Chen, F. He, Y. Wu, N. Hou, A local start search algorithm to compute exact Hausdorff Distance for arbitrary point sets, Pattern Recogn. 67 (2017) 139–148.
[14] Y. Qin, W. Lu, Q. Qi, Status, comparison, and issues of computer-aided design model data exchange methods based on standardized neutral files and web ontology language file, J. Comput. Inf. Sci. Eng. 17 (1) (2017) 010801.
[15] D.J. Zhang, F. He, S. Han, X.X. Li, Quantitative optimization of interoperability during feature-based data exchange, Integr. Comput.-Aided Eng. 23 (1) (2016) 31–51.
[16] M. Van Der Meulen, R.H. Logie, Y. Freer, When a graph is poorer than 100 words: a comparison of computerised natural language generation, human generated descriptions and graphical displays in neonatal intensive care, Appl. Cognit. Psychol.: Off. J. Soc. Appl. Res. Memory Cognit. 24 (1) (2010) 77–89.
[17] X. Lv, F. He, W.W. Cai, Y. Cheng, A string-wise CRDT algorithm for smart and large-scale collaborative editing systems, Adv. Eng. Inform. 33 (3) (2017) 397–409.
[18] A. Gatt, E. Krahmer, Survey of the State of the Art in Natural Language Generation: core tasks, applications and evaluation, J. Artif. Intell. Res. 61 (2018) 65–170.
[19] Y. Pan, F. He, H. Yu, A novel enhanced collaborative autoencoder with knowledge distillation for top-N recommender systems, Neurocomputing 332 (7) (2019) 137–148.
[20] S. Pauws, A. Gatt, E. Krahmer, Making effective use of healthcare data using data-to-text technology, 2018. arXiv preprint arXiv: 1808.03507.
[21] K. Li, F.Z. He, H.P. Yu, X. Chen. A parallel and robust object tracking approach synthesizing adaptive Bayesian learning and improved incremental subspace learning, Front. Comput. Sci. https://doi.org/10.1007/s11704-018-6442-4.
[22] P. Conde-Clemente, J.M. Alonso, G. Trivino, Toward automatic generation of linguistic advice for saving energy at home, Soft. Comput. 22 (2) (2018) 345–359.
[23] X. Chen, F. He, H. Yu, A matting method based on full feature coverage, Multimedia Tools Appl. https://doi.org/10.1007/s11042-018-6690-1.
[24] D. Sanchez-Valdes, G. Trivino, Computational perceptions of uninterpretable data. A case study on the linguistic modeling of human gait as a quasi-periodic phenomenon, Fuzzy Sets Syst. 253 (2014) 101–121.
[25] H.P. Yu, F. He, Y. Pan, A novel segmentation model for medical images with intensity inhomogeneity based on adaptive perturbation, Multimedia Tools Appl. https://doi.org/10.1007/s11042-018-6735-5.
[26] S. Mahamood, W. Bradshaw, E. Reiter, Generating annotated graphs using the nlg pipeline architecture, 8th International Natural Language Generation Conference (INLG), 2014, pp. 123–127.
[27] K. Li, F. He, H.P. Yu, Robust visual tracking based on convolutional features with illumination and occlusion handing, J. Comput. Sci. Technol. 33 (1) (2018) 223–236.
[28] H. Xu, B. Li, V. Ramanishka, Joint Event Detection and Description in Continuous Video Streams, 2018. arXiv preprint arXiv: 1802.10250.
[29] S. Zhang, F. He, W. Ren, J. Yao, Joint learning of image detail and transmission map for single image dehazing, The Visual Comput. https://doi.org/10.1007/s00371-018-1612-9.
[30] K. Chen, C.B. Choy, M. Savva, Text2Shape: Generating Shapes from Natural Language by Learning Joint Embeddings, 2018. arXiv preprint arXiv:1803.08495.
[31] K. Li, F. He, H. Yu, X. Chen, A correlative classifiers approach based on particle filter and sample set for tracking occluded target, Appl. Math.-A J. Chin. Univ. 32 (3) (2017) 294–312.
[32] J. Johansson, D.C. Brown, Automated Computer Systems for Manufacturability Analyses and Tooling Design: Applied to the Rotary Draw Bending Process, [Doctoral dissertation] School of Engineering, Jonkoping University, 2011.
[33] R. Owen, I. Horvath, Towards product-related knowledge asset warehousing in enterprises, Proceedings of the 4th International Symposium on Tools and Methods of Competitive Engineering, TMCE, 2002, p. 155.
[34] M. Poorkiany, J. Johansson, F. Elgh, Capturing, structuring and accessing design rationale in integrated product design and manufacturing processes, Adv. Eng. Inform. 30 (3) (2016) 522–536.
[35] S.J. Fenves, Core Product Model for Representing Design Information, US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.
[36] R. Sudarsan, S.J. Fenves, R.D. Sriram, A product information modeling framework for product lifecycle management, Comput.-Aided Des. 37 (13) (2005) 1399–1411.
[37] K. Myers, N. Zumel, P. Garcia, Acquiring design rationale automatically, artificial intelligence for engineering design, Anal. Manufact. 14 (2) (2000) 115–135.
[38] M.J. Pratt, Introduction to ISO 10303 - the STEP standard for product data exchange, J. Comput. Inf. Sci. Eng. 1 (1) (2001) 102–103.
[39] ISO 10303-23803, Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 203: Application Protocol: Configuration Controlled 3D Design of Mechanical Parts and Assemblies, ISO, Geneva, Switzerland, 2011.
[40] ISO 10303-214, Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 214: Application Protocol: Core Data for Automotive Mechanical Design Processes, ISO, Geneva, Switzerland, 2010.
[41] ISO 10303-242 2014, Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 242: Application Protocol: Managed Model-Based 3D Engineering, ISO, Geneva, 2014.
[42] STEP AP242 Project organization, STEP AP242 Project, 2014. http://www.ap242.org/ [last accessed Sept.18, 2018].
[43] http://pdteurope.com/wp-content/uploads/2016/11/4a-AP242-and-JT-Standards-Assessment-A-focus-on-management-of-GDT-and-PMI.pdf.
[44] C. Dartigues, P. Ghodous, Product data exchange using ontologies, Artificial Intelligence in Design 2002, Springer, Dordrecht, The Netherlands, 2002, pp. 617–637.
[45] L. Patil, D. Dutta, R.D. Sriram, Ontology-based exchange of product data semantics, IEEE Trans. Autom. Sci. Eng. 2 (3) (2005) 213–225.
[46] L. Patil, Interoperability of Formal Semantics of Product Data Across Product Development Systems, University of Michigan, Ann Arbor, MI, 2005.
[47] T.S. Seo, Y.S. Lee, S.U. Cheon, S.H. Han, P. Lalit, D. Debasish, Sharing CAD models based on feature ontology of commands history, Int. J. CAD/CAM 5 (1) (2005) 39–47.
[48] S. Tessier, Y. Wang, Ontology-based feature mapping and verification between CAD systems, Adv. Eng. Inform. 27 (1) (2013) 76–92.
[49] K.Y. Kim, D.G. Manley, H. Yang, Ontology-based assembly design and information sharing for collaborative product development, Comput. Aided Des. 38 (12) (2006) 1233–1250.
[50] J. Kim, M.J. Pratt, R.G. Iyer, R.D. Sriram, Standardized data exchange of CAD models with design intent, Comput. Aided Des. 40 (7) (2008) 760–777.
[51] S.M. Jeon, J.H. Lee, G.J. Hahm, Automatic CAD model retrieval based on design documents using semantic processing and rule processing, Comput. Ind. 77 (2016) 29–47.
[52] W. Lu, Y. Qin, Q. Qi, W. Zeng, Y. Zhong, X. Liu, X. Jiang, Selecting a semantic similarity measure for concepts in two different CAD model data ontologies, Adv. Eng. Inform. 30 (3) (2016) 449–466.
[53] F. Qin, S. Gao, X. Yang, M. Li, J. Bai, An ontology-based semantic retrieval approach for heterogeneous 3D CAD models, Adv. Eng. Inform. 30 (4) (2016) 751–768.
[54] L. Petnga, M. Austin, An ontological framework for knowledge modeling and decision support in cyber-physical systems, Adv. Eng. Inform. 30 (1) (2016) 77–94.
[55] J.C. Jensen, D.H. Chang, E.A. Lee, A model-based design methodology for cyber-physical systems, Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2011, pp. 1666–1671.
[56] A. Bhave, B. Krogh, D. Garlan, B. Schmerl, Multi-domain Modeling of CyberPhysical Systems Using Architectural Views, Dept. of Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15217, 2010.
[57] S. Abdul-ghafour, P. Ghodous, B. Shariat, Semantic interoperability of knowledge in feature-based CAD models, Comput.-Aided Des. 56 (2014) 45–57.
[58] R. Barbau, S. Krima, S. Rachuri, OntoSTEP: enriching product model data using

ontologies, Comput. Aided Des. 44 (6) (2012) 575–590.
[59] L. Zhu, U. Jayaram, S. Jayaram, O. Kim, Ontology-driven integration of CAD/CAE applications: strategies and comparisons, ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2009, pp. 1461–1472.
[60] H. Sun, W. Fan, W. Shen, Ontology-based interoperation model of collaborative product development, J. Network Comput. Appl. 5 (1) (2012) 132–144.
[61] E.M. Sanfilippo, Ontological foundations for feature-based modeling, Procedia CIRP 70 (2018) 174–179.
[62] J. Camba, M. Contero, M. Johnson, Extended 3D annotations as a new mechanism to explicitly communicate geometric design intent and increase CAD model reusability, Comput. Aided Des. 57 (2014) 61–73.
[63] G. Alducin-Quintero, A. Rojo, F. Plata, 3D model annotation as a tool for improving design intent communication: a case study on its impact in the engineering change process, ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2012, pp. 349–356.
[64] W. Taha, R. Philippsen, Modeling basic aspects of cyber-physical systems, 2013. arXiv preprint arXiv:1303.2792.
[65] J. Gross, S. Rudolph, Geometry and simulation modeling in design languages, Aerosp. Sci. Technol. 54 (2016) 183–191.
[66] J. Altidor, J. Wileden, J. McPherson, I. Grosse, S. Krishnamurty, F. Cordeiro, A.L.S. John, A programming language approach to parametric CAD data exchange, ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2011, pp. 779–791.
[67] R. Bracewell, K. Wallace, M. Moss, D. Knott, Capturing design rationale, Comput. Aided Des. 41 (2009) 173–186.
[68] G. Peng, H. Wang, H. Zhang, Y. Zhao, A.L. Johnson, A collaborative system for capturing and reusing in-context design knowledge with an integrated representation model, Adv. Eng. Inform. 33 (2017) 314–329.
[69] Y. Wang, S. Yu, T. Xu, A user requirement driven framework for collaborative design knowledge management, Adv. Eng. Inform. 33 (2017) 16–28.
[70] J. Stjepandi, W.J.C. Verhagen, H. Liese, P. Bermell-Garcia, Knowledge-based engineering, Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges, 2015, pp. 255–286.
[71] J. Schmidt, M. Borsato, E. Hinckel, E. Onofre, F. Maccari, A framework for capturing and applying design knowledge in new product development, Int. J. Agile Syst. Manage. 11 (1) (2018) 23–40.
[72] S. Staab, R. Studer, Handbook on Ontologies, Springer Science & Business Media, 2010.
[73] S.J.B. Shum, A.M. Selvin, M. Sierhuis, J. Conklin, C.B. Haley, B. Nuseibeh, Hypermedia support for argumentation-based rationale: 15 years on from Gibis and Qoc, Rationale Manage. Softw. Eng. (2006) 111–132.
[74] J. Lee, Decision representation language (DRL) and its support environment, 1989.
[75] J.E. Burge, D.C. Brown, Software engineering using rationale, J. Syst. Softw. 81 (2008) 395–413.
[76] A. MacLean, R.M. Young, V.M. Bellotti, T.P. Moran, Questions, options, and criteria: elements of design space analysis, Hum.-Comput. Interact. 6 (3–4) (1991) 201–250.
[77] E. Reiter, R. Dale, Building applied natural language generation systems, Nat. Lang. Eng. 3 (1) (1997) 57–87.
[78] Y. Bodein, B. Rose, E. Caillaud, Explicit reference modeling methodology in parametric CAD system, Comput. Ind. 65 (1) (2014) 136–147.
[79] Z.R. Cheng, Y.S. Ma, A functional feature modeling method, Adv. Eng. Inform. 33 (2017) 1–15.
[80] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Shape-based searching for product lifecycle applications, Comput. Aided Des. 37 (13) (2005) 1435–1446.
[81] X. Yan, F. He, N. Hou, H.J. Ai, An efficient particle swarm optimization for large scale hardware/software co-design system, Int. J. Cooper. Inform. Syst. 27 (01) (2018) 1741001.
[82] X. Yan, F. He, Y.L. Chen, A novel hardware/software partitioning method based on position disturbed particle swarm optimization with invasive weed optimization, J. Comput. Sci. Technol. 32 (2) (2017) 340–355.
[83] Y. Zhou, F. He, Y.M. Qiu, Dynamic strategy based parallel ant colony optimization on GPUs for TSPs, Sci. China Inform. Sci. 60 (6) (2017) 068102.
[84] Y. Zhou, F. He, N. Hou, Y. Qiu, Parallel ant colony optimization on multi-core SIMD CPUs, Future Gener. Comput. Syst. 79 (2) (2018) 473–487.