Advanced Engineering Informatics 33 (2017) 1–15

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

A functional feature modeling method

Zhengrong Cheng, Yongsheng Ma *

Department of Mechanical Engineering, University of Alberta, Canada

ARTICLE INFO

Article history:
Received 22 November 2016
Received in revised form 13 March 2017
Accepted 12 April 2017
Available online 27 April 2017

Keywords:
CAD
Functional modeling
Feature-based design
Modeling methodology

ABSTRACT

With the advances in CAD technology, it has been increasingly convenient to model product shapes digitally. For example, in a feature-based parametric CAD system, the product shape could be parameterized and thus altered with the change of parameters. However, without a consistent and systematic CAD modeling method, CAD models are not robust enough to capture functional design knowledge and cope with design changes, especially functional changes. A poorly constructed CAD model could result in erroneous or inconsistent design that requires a lot of expertise, manpower and repetitive computation to rebuild a valid and consistent model. The situation can be worse if the model is complex. The gap between functional design considerations and procedural CAD modeling demands an integrated CAD modeling approach. This paper proposes a functional feature-based CAD modeling method to guide designers building CAD models that are valid and yet agile to represent functional design considerations. A case study is presented to demonstrate the feasibility of the proposed research.

1. Introduction

CAD tools are helpful in the modern engineering design. They accelerate product development by creating virtual product models with highly flexible geometrical features that are easy to be manipulated, for example, blocks, holes, and fillets features, which help to maintain consistency of lower level geometric entities like faces, edges, and vertices based on Euler operators [26]. With the commonly seen procedural modeling approach through manipulating a number of intermediate operations, the desired form of the design artefact can be obtained. CAD systems also have the ability to reuse and make modifications to existing models; hence further extending their usability in the engineering design [9]. Reusability in the CAD domain means that CAD models can be altered to adapt to new use cases with little effort. The reusability of CAD models foster the design reusability because more and more design information is stored in the CAD models and they are becoming indispensable for downstream engineering activities, such as manufacturing [51,38,21], engineering analysis [34,27,31,45,59], and optimization [60]. Reusability requires CAD models to be robust. By saying robustness of a CAD model, the authors mean that it should have the quality of reusability that it is modifiable to certain extent without rendering the model into inconsistency or jeopardizing the model. Therefore, effective representation, expression, and communication of design intents are critical.

Feature-based parametric modeling is widely applied in the industry to create product parts and assembly models. Product models could be parameterized to the extent that each building scheme or pattern of a product, that is to say any feature, form feature, manufacturing feature, and detailed design feature, can be parameterized. The models can then be updated with the changes of parameters; hence the parts and assemblies could be regenerated without designers manually going through the remodeling process. Fig. 1 shows an example of implementing parameterizing feature with expressions in Siemens NX®. Expressions are named parameters with mechanisms to interact with features in NX®, e.g., by remembering their owning and using features. Features provide a manner of representing semantic patterns of design intent. They can be constructed at higher associative assembly level [41,40], and also detailed up to a low level of granularity, e.g., hole features, edge blend features.

By maintaining the feature parentships during the model creation, modeling history could be preserved such that when the model needs to be regenerated due to design changes the changes could propagate downward from feature to feature, thus creating a form of dependencies [5]. Fig. 2 shows some examples of dependencies in CAD modeling, e.g., datum dependencies, parameter dependencies, and geometry dependencies.

However, some issues may arise from the feature dependencies in CAD. Designers might not be fully aware of the feature dependencies and the constructed model is fragile. Some of the

---

* Corresponding author.

E-mail address: yongsheng.ma@ualberta.ca (Y. Ma).

http://dx.doi.org/10.1016/j.aei.2017.04.003
1474-0346/

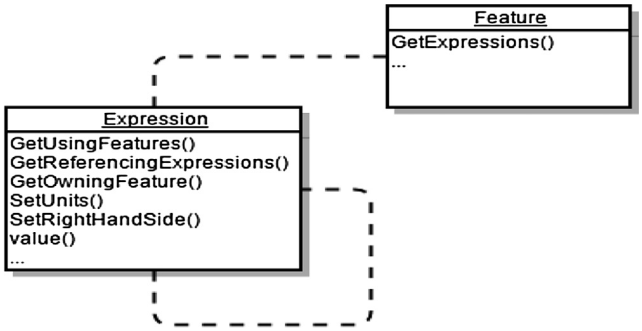


Fig. 1. An example implementation of parameterized feature with expressions.

misbehaviors are easily observable if the model ends up visually and explicitly wrong. Other errors are harder to be detected with human eyes when they are visually less obvious but functionally critical. There is no systematic way to manage the dependencies and designers usually have to redo part of the model operations. Namely, the burden of managing the cumbersome interdependencies of the feature operations lies on the designers [6,9]. A procedural CAD modeling approach is not close to designers' *way of thinking* due to the gap between features, patterns, or shapes designers have in mind, and the modeling functions and operations provided to them [26]. For example, Fig. 3(a) presents the shape of connection rod and (b) shows the corresponding modeling operations. The gap is the one between the engineering design intents behind the features and the applied procedural CAD modeling operations.

A lot of questions could be asked for designers when creating CAD models [6], 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 the *Boolean* operations 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. Sadly designers are often content in creating the shape of the design artefact without giving much thought on the robustness of the model, which is, based on above discussion, clearly insufficient. The authors believe that the question is less of the procedural modeling approach itself, but more on how to apply the procedural modeling more effectively for functional modeling from the angle of engineering innovation.

The approach adopted by this research work is to tackle the CAD modeling efficiency problem from functional perspective in a top-down manner [12]. Top-down design is an assembly modeling approach that can drive multiple part designs by using a single "parent" part, where users create geometry at the assembly level (the parent part) and then move or copy the geometry to one or more components (children parts). The generic idea of top-down design is taken as a starting point for current research, instilled with functional flavor. With the understanding of multiple possibilities to create a specific product model, a functional understanding of the design is not only important in the conceptual design stage but also critical to provide modeling guidance during the process of model detailing and the subsequent derivation of other downstream engineering models and activities. By incorporating the functional design considerations into CAD models, the authors believe that the functional usability of the CAD model can be significantly improved – this belief led to this research effort, i.e. improving robustness of CAD models by conveying design intents explicitly in the model construction.

The rest of the paper is organized as follows. Section 2 presents the related literatures to this research, including feature modeling, communication of design intent, representing function in engineering design, and other intelligent methods to build robust CAD models. Section 3 introduces the proposed method within general framework of functional features, some of the key elements of functional features that are pertinent to the current research, and CAD modeling procedure to build robust function-oriented CAD models. A case study is demonstrated in Section 4, which incorporates the proposed method and proves the validity and effectiveness of proposed method. The last section concludes the paper.

2. Review of related works

2.1. Feature technology and CAD modeling

According to Shah and Mantyla [49], features represent the engineering meanings or significances of the geometry of a part

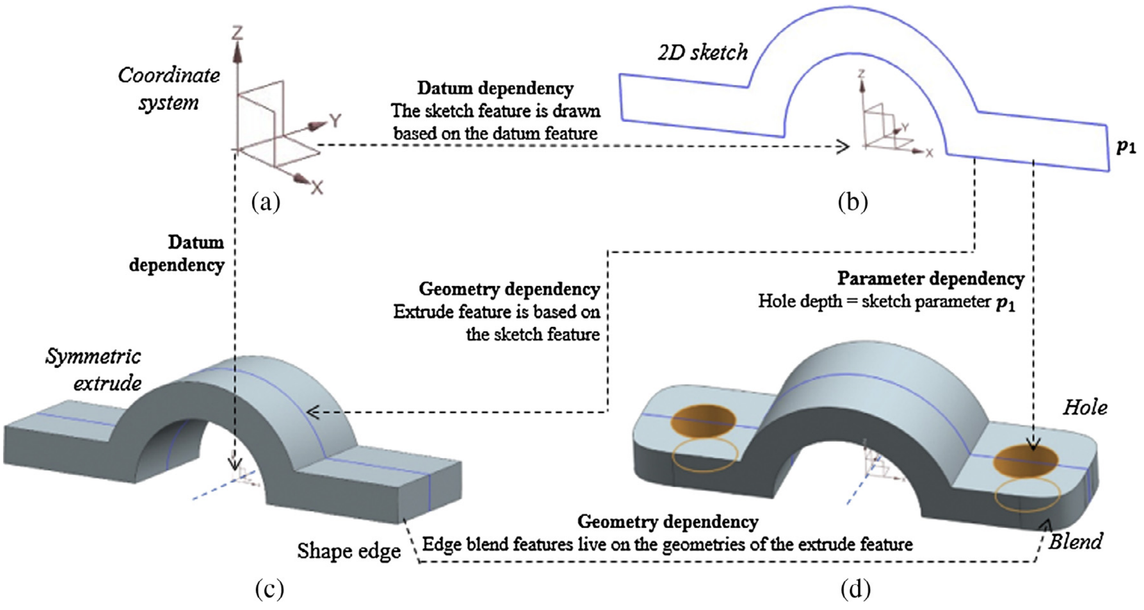
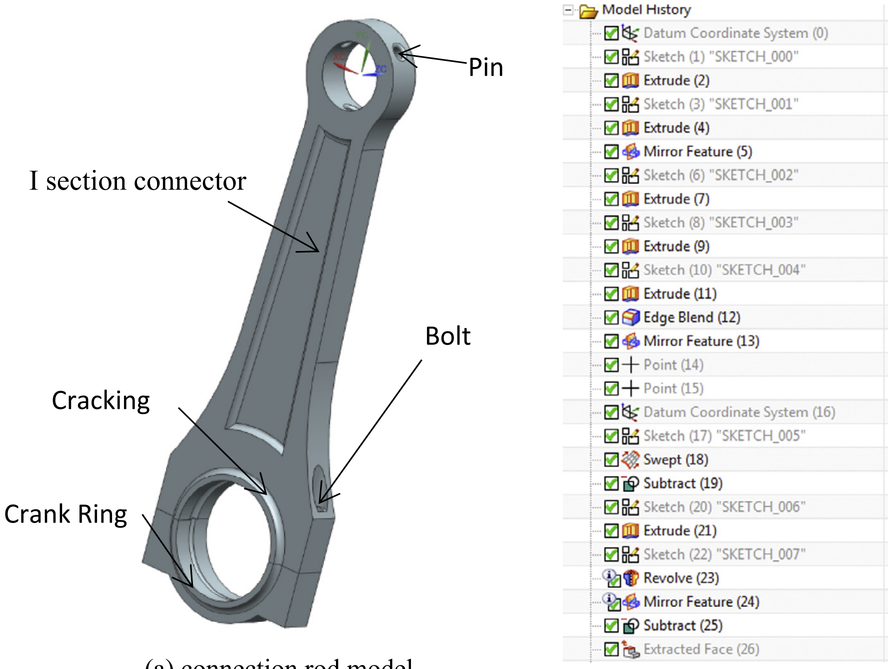


Fig. 2. Some examples of dependencies in CAD.



Model History
☑ Datum Coordinate System (0)
☑ Sketch (1) "SKETCH_000"
☑ Extrude (2)
☑ Sketch (3) "SKETCH_001"
☑ Extrude (4)
☑ Mirror Feature (5)
☑ Sketch (6) "SKETCH_002"
☑ Extrude (7)
☑ Sketch (8) "SKETCH_003"
☑ Extrude (9)
☑ Sketch (10) "SKETCH_004"
☑ Extrude (11)
☑ Edge Blend (12)
☑ Mirror Feature (13)
☑ Point (14)
☑ Point (15)
☑ Datum Coordinate System (16)
☑ Sketch (17) "SKETCH_005"
☑ Swept (18)
☑ Subtract (19)
☑ Sketch (20) "SKETCH_006"
☑ Extrude (21)
☑ Sketch (22) "SKETCH_007"
☑ Revolve (23)
☑ Mirror Feature (24)
☑ Subtract (25)
☑ Extracted Face (26)

(a) connection rod model

(b) modeling operations for connection rod

Fig. 3. A connection rod showing gaps between design intent and procedural modeling operations.

or assembly and could serve as building blocks for product definition and geometric reasoning. Manufacturing planning gives the origin for feature technologies where features correspond to the volumes in the product that could be machined with a single or a sequence of operations [4], for example, hole features instead of cylinders, such that features could have engineering semantics. In the academic world, new features are being created to extend their application domains, for example, user defined features [29], associative features [41,40] kinematics features [2], rib features [35], and user-defined freeform feature [28]. Feature technologies have been applied to CAD in great extent. Current mainstream CAD systems provide modeling operations in terms of features, e.g., block feature, extrude features, revolve features, and fillet features, which are generally categorized as form features. Assembly features, expressing the relationships that exist between different parts within an assembly, are applied to position or orient the parts, restraining the degree of freedoms, usually in the form of mating conditions [42]. A general feature-based CAD modeling procedure is shown in Fig. 4.

Parametric capability of feature-based CAD makes it easier to integrate quantitative design knowledge into the model such that it is possible to change the product model with alterations of values. Dimensions of each feature are controlled by a set of parameters. Since product models are created with features, if applied properly, the whole product shape could be manipulated with a set of parameters. Lin et al. [36] and Lin and Hsu [37] presented an automated design system for drawing dies built on top of a commercial CAD system in a knowledge-based approach. By using the combination capacity of CAD system, design formulas and geometric operations of modeling processes are generated by the system with a minimum set of structure parameters to reduce design time. However, the formulation and determination of the parameters are unclear. Since the construction of features might depend on some previously defined features, feature dependencies are created [5]. The internal tree structure for features in the history-based CAD helps to keep the associated (i.e. parent/children) relations among the features. Powerful as the systems are, the burden of choosing appropriate parameterization and feature operation sequences are still loaded on the designers.

Feature parameter maps are generalized dependency maps among different parameters [62]. In Yin and Ma [62] feature parameter map is a conceptual organization scheme for modeling dependencies among parameters at a lower level of information granularity than features. The procedure follows a top-down design approach. Excel was used for implementation where parameters relations are embedded into formulas. A set of parameters designed to be interfacing with CAD parametric models in part or assembly levels are maintained specifically such that expression synchronization mechanism available in the CAD tool could be used to update the CAD model from Excel. Different levels of feature parameter maps could be constructed, for example, in the conceptual design level, component design level, and assembly design level.

Although parametric design has been applied in the CAD modeling, existing literatures acknowledge the management of structure parameters, i.e., geometrically related parameters, without considering management of non-geometric parameters that have impacts on the product geometry in the CAD system. Camba et al. [9] reviewed three formal parametric modeling methodologies specifically designed to emphasis CAD reusability: Delphi's horizontal modeling, explicit reference modeling, and resilient modeling. They examined the advantages and disadvantages of each approach and comparing their effectiveness during design changes with experiments. Their results reveal the importance of using formal modeling methodology to increase CAD reusability. Bodein et al. [7] presented a framework of actions that can guide designers to improve CAD efficiency by utilizing the advantages of parametric CAD in the automotive industry. Their CAD strategy roadmap consists of standardization, advanced methodology, KBE, and expert rules check, where a lot of trainings are required. Bodein et al. [6] proposed a practical method for complex part modeling in parametric CAD system by explicit management of

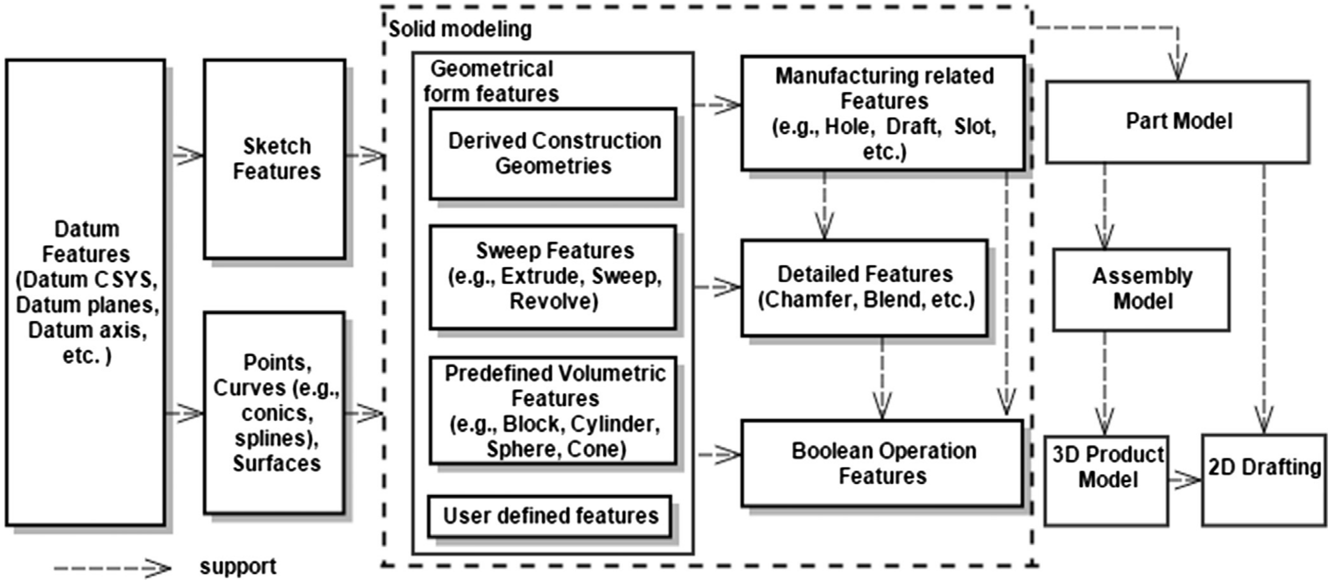


Fig. 4. The traditional feature-based CAD modeling procedure.

references. They decompose a part model into different regions by their "functions" and model those geometries individually at first with their own references. Later, *Boolean* operations are applied to those "functional geometries". Note that in their research their "functional geometries" are solid that could be applied with Boolean operations. It is not clear whether their approach is still applicable when a part region has overlapping functions. Moreover, no details of how to reach the functional geometries of the design are given. It is safe to conclude that it is still in need of a generic CAD modeling methodology.

2.2. Communication of design intents through CAD

Another aspect of building robust CAD model is to foster effective communication of design intents through CAD models. Note that there is no general consensus on the definition of design intent as researchers suggest their own definitions [30]. Design intents need to be documented and managed [16]. Within CAD models design intents are usually expressed implicitly and approach like annotation has been applied to encourage the communication of design intents [10,8]. Annotation elements, such as comments, are commonly used for clarification or explanation [8]. For example, in the programming process of software development comments are used to clarify the functionality and sometimes to provide implementation explanation of the piece of code. In the engineering design domain, annotations are usually seen in 2D drawings, which provide necessary information like dimensions and tolerances for the purpose of manufacturing, as well as complementary verbal explanations that could not be shown in the dimensioning of the product. In the 3D CAD environment, technologies like Product and Manufacturing Information (PMI) provide an approach to attach information to a part or assembly model. PMI objects include dimensions, datum, notes, symbols, and section views, to be used by downstream activities, for example, tooling, manufacturing, inspection and shipping. Moreover, PMI objects support necessary operations, for instance, move, edit, delete, and control of visibility.

In academia, Camba et al. [10] presented an extended annotation method to communicate geometric design intents where design information is represented both internally within the CAD model and externally on a separate repository. Semantic annotations of CAD models are also available in the research community; for example, a system based on segmentations [1] of 3D surface meshes and annotations of the detected part shapes expressed by ontology has been reported. Shapes are decomposed into interesting features within the multi-segmentation framework and annotation pipelines are used to attach semantics to the features and the whole shape. However, as Attene et al. [1] admitted, the inference capability of the system is limited.

Although annotation approach helps to convey design intents in CAD models between designers, the user could not interact with the CAD model directly through annotations. Moreover, when the CAD model is changed, the annotation might not be fully updated automatically and correctly. In addition, even though other designers understand the design intents behind the CAD model with the help of annotations, it is still not clear about: (1) how to effectively alter the CAD model to cope with new design requirements; (2) whether the model will update smoothly with the alterations. That is to say, the annotation is functionally separated from the model construction. Hence the approach of applying annotations on the CAD model to communicate design intents is not enough. It would be superior if the created models themselves reflect design intents and are responsive to the changes of design intents. Design intents are usually aligned with the functionalities of the design artefact, which means that if the CAD models could be constructed in a functional way they can convey design intents systematically. Since the construction approach is associated to the features representing functional intents, they are easier to be altered when functional changes are made.

2.3. Representing functions in design

Functions in the engineering design provide a conceptualization of the purpose of the design artefact, linking different levels of product or system design. Functions have been represented differently in academia, for example, verb-noun pairs [39] such as "transmit force"; "to do" form [32], e.g., "ToMake", "ToPrevent", etc.; and input-output flow transformation, i.e., energy, material, and information [11]. Approaches are available to generate function model. In Function Design Framework (FDM) a four steps methodology is required to generate functional models: identifying the boundary of environment, identifying process boundary, identifying physical boundaries, and decomposing the process and system to the desired level of fidelity [43,44]. Sometimes the generated functions are not primitive enough in the sense that they cannot be fulfilled with low level components. In this situation, functional decomposition could be applied to break down an overly abstract function into several more specific and primitive functions, usually called subfunctions, for example, task decomposition and causal decomposition [56,57]. Usually functional decomposition is done manually. For instance, Deng et al. [15] gave an example of functional decomposition of the assembly of connectors. Recent research tries to at least partially computerize the process, e.g., a hybrid approach to automate functional decomposition in conceptual design by Yuan et al. [63] with qualitative processing reasoning, physical effect decomposition, and backward search decomposition.

Moreover, when people are talking about functions, they tend not to say them in an isolated manner, i.e., functions need to be related with some other aspects in design. A few approaches are available and applicable to correlate and/or integrate functions with other design aspects, especially in the system design level, e.g., Axiomatic Design (AD), Design Structure Matrix (DSM) and Quality Function Deployment (QFD), Requirement Functional Logical Physical (RFLP) [53,55,17,14]. In AD, entities like functional requirements, constraints, design and parameters are defined to facilitate the design reasoning process, where functional requirements are defined as a minimum set of independent requirements that completely characterize the functional needs of the product [53]. For example, in the design of a refrigerator one of the functional requirements could be "to control temperature of the freezer section". QFD has been applied to build the functional relations between customer requirements (CRs) and engineering characteristics (ECs) [19]. RFLP [14] is a system engineering approach to integrate different design activities and processes with Systems Architecture Design & Simulation solution, where the identification of system functionalities is vital.

Other approaches focusing on teleological modeling in engineering design are also available. Models like Function-Behavior-Structure [23,24], Structure, Behavior, and Function [25], Function-Behavior-State [56–58], and functional diagram [54] all treat functions as critical components in the design thinking and provide conceptual approaches to organize the various mappings among functions, behaviors, structures, states, etc. For instance, function diagram in Teoh and Case [54] are used to represent function and structure interactions with a network connected by multiple function units, which is defined as *function operator – function →function operand*, i.e. a verb or verb phrase that defines the action and connects two objects (function operator and function operand).

Admittedly, the conceptual functional design approach is helpful to organize engineers' mental flows and the existing correlation approaches (AD, DSM, QFD, etc.) help to model relations between functions and other design aspects. However, when it comes to the CAD modeling, such functional considerations and correlations are often not sufficiently captured. If functions could be represented within CAD model, then, with the help of existing correlation approaches, the whole product design lifecycle could be dynamically linked and associated. In addition, because CAD modeling involves part modeling and assembly modeling with features, it is not enough to have the correlation with function and structure in the component level but to go deeper into the feature level.

3. Proposed method

3.1. Functional feature framework

This research work proposes a functional CAD modeling approach which can guide the modeling process to achieve robust CAD models by organizing the functional relationships of the design elements such that they are constrained and parameterized properly. Roy and Bharadwaj's [48] Part Function Model (PFM) provides a starting point to connect part function relations with the faces of the part, the concept of which will be extended in the current research. The key idea is that the model should be built with future changes in mind. When building CAD models, designers should have an idea about what kind of changes are foreseeable and the models should be robust enough to cope with the changes. However, given any model there could be a vast amount of possible changes. Therefore it is not cost-effective to build a model that is robust to the extent that it could update automatically given any changes, if not impossible. Since design is to provide certain functionalities and usually the sources of design changes are from alterations of functional considerations, the authors argue that CAD models should be built robust enough to cope with functional changes to increase the model reusability and changeability. Moreover, by embedding functional changes capability into CAD models functional interfaces could be built, which would make it convenient and intuitive for designers and users to make modifications to the design models. As a matter of fact, what the possible functional changes are was one of the main questions brought about in Bodein et al. [6]. In this scenario, engineers should not only have knowledge of how to use CAD tools, but also have certain level of understanding of the functions of the design artefacts. The authors deem it necessary and beneficial. CAD modeling should start with considering product functionality and the results of functional design activities should be carried over and embedded into CAD modeling.

Functional feature is a new type of feature proposed by the authors, trying to capture and represent functional design considerations into CAD models to facilitate the integration of different design activities. The semantic definition of functional feature is provided in Fig. 5 [13]. Functional feature modeling provides a mechanism from which functionally robust CAD models could be created. It is an extension of associative feature [41,40], with new ingredients (e.g., abstract geometry features), serving a more specific purpose, i.e., integrating functional considerations in CAD modeling. The focus of current work is a part of the framework, tackling the modeling issue of applying abstract geometry features in detailed CAD part modeling to build robust models, where the most relevant parts of the functional feature elements will be discussed. The general idea of the framework is to use functional feature to bridge the gap between conceptual functional design and procedural CAD modeling. Abstract geometry feature is one of the key aspects of functional feature. It is a concept carrier and handles the geometric related issue in the modeling process.

3.1.1. Functional considerations of design

Functional considerations of design have multiple sources, most well-known of which include customer requirements and engineering considerations (see Fig. 6). Approaches like Quality Function Deployment (QFD) and Axiomatic Design (AD) [53] could be used to transfer customer requirements to functional requirements [61], which might not be extensive as customers focus more on their "consumer needs" so that engineering judgements also need to be incorporated into the model to enrich the semantics of functional requirements. In addition to bringing up new functional requirements, engineering considerations are critical as most times customer requirements tend to be qualitative and engineering thinking need to be instilled to quantify them. Moreover, system level functions need to be broken down into lower level granularities, the process of which is called functional divergence in current research. Later processes involve encapsulating and materializing the functions into different parts, which is called parts convergence.

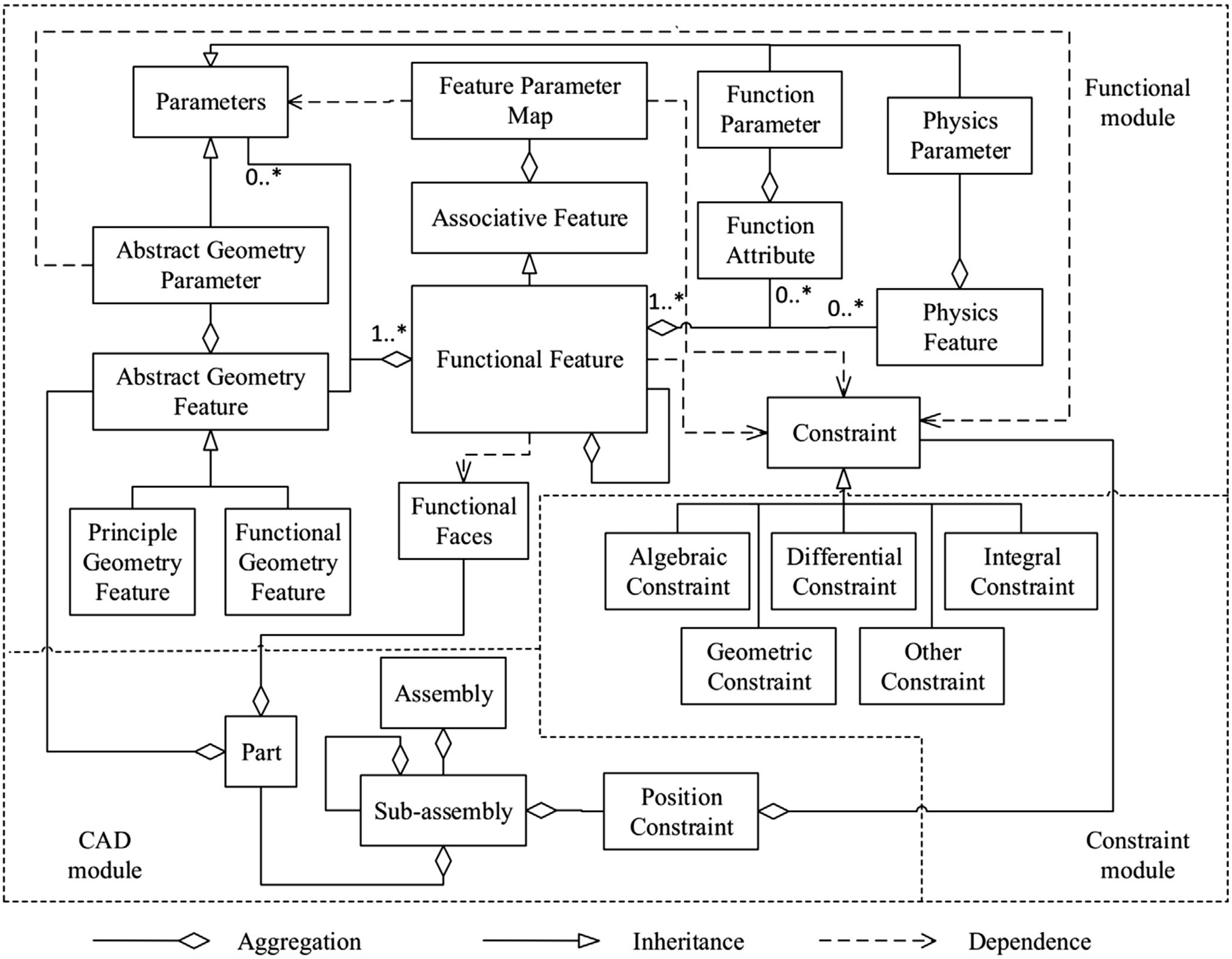


Fig. 5. UML diagram representing functional feature [13].

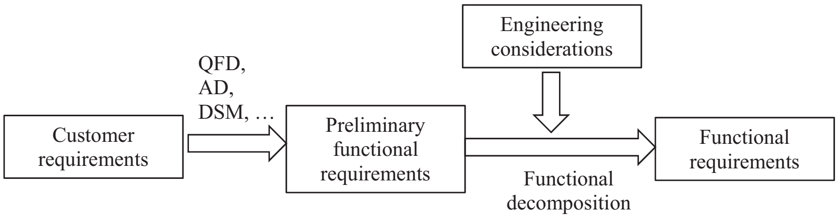


Fig. 6. Information mapping from customer requirements to functional requirements.

With the identified functional requirements, possible functional changes of design could be foreseen.

3.1.2. Abstract geometry features as functional concepts carriers

According to Roy and Bharadwaj's [48] Part Function Model (PFM), faces of the part could be connected to part function relations. In current research, abstract geometry feature is proposed to handle representation of geometric elements of functions in CAD. Abstract geometry feature is a functional design concept carrier with an abstraction of the key characteristic shapes of the design artefacts. It provides, on the one hand, a suitable form of geometry for conceptual design and, on the other hand, guidance for modeling of detailed design geometry. Abstract geometry feature includes principle geometry feature and functional geometry feature. It could be represented with both manifold and non-manifold geometries. The key difference between form features and abstract geometry features is that form features are usually directly used in the construction of a solid model, i.e. as building blocks, whereas abstract geometry features are used as concept abstraction carriers, which might not contribute directly to the detailed CAD geometry of the product. This being said, abstract geometry features do share similarities with conventional form features, i.e., they both need to have references, relevant geometric entities, parameters, constraints, and, possibly, feature ownerships. It could be made into template [52] where other knowledge elements could be attached.

An abstraction process needs to be carried out to reach at abstract geometry features. The abstraction rules are flexible and application dependent. Some general guidelines are: (1) abstract geometry feature captures the kinematic relation (physics) of the design (e.g., gear, slider and crank mechanism), (2) abstract geometry feature captures the general shape of the design (e.g., pressure vessel without thickness), (3) abstract geometry feature captures functionally important shapes. With the characteristic geometry available, detailed and fully-fledged CAD geometry can be refined from the abstract geometry. The refinement could be done either manually or automatically with some programming. Fig. 7 gives two examples of abstraction and embodiment of geometries of different levels of details in the design process. Attributes are attached to the abstract geometry features to enrich their semantics such that downstream activities could derive the necessary information from the object of abstract geometry features. For example, the tube used in the slotted liner abstract geometry feature is constructed without thickness. The thickness value could be attached to the abstract geometry feature such that in generating the detailed part geometry operation could be performed by reading the thickness value of the abstract geometry feature.

There are more ways than one to construct the geometry elements of each abstract geometry feature in CAD. Constructed abstract geometry features could be stored in the feature library for future usages. Some abstract geometry features cannot be converted to detailed CAD product geometry straightforwardly. However, it does control some key dimensional constraints of the product geometry. Implementation wise, abstract geometry features are implemented as User Defined Features (UDFs) or User Defined Objects (UDOs). During the construction of detailed part model, users can use them as regular features. The geometric and non-geometric elements can be referred to freely, or use them as separate parts models from which other parts can refer to by using technology like, for example, WAVE from Siemens NX.

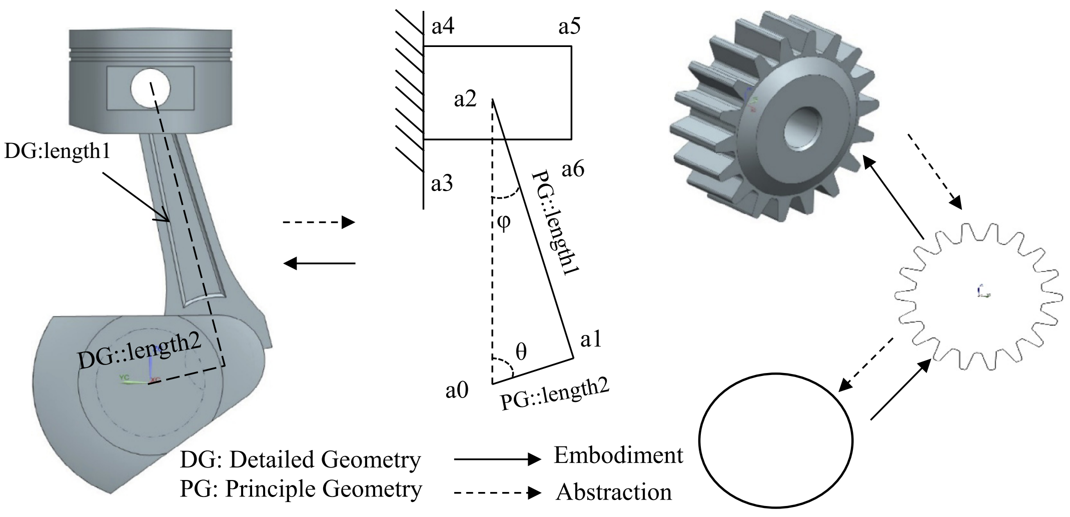
3.1.3. Constraints and parameterization

Constraints in the engineering design state the conditions that need be satisfied for the design to be viable and help to reduce the feasible solution space. Constraints are imposed on the design from different sources. For example, for the consideration of structural integrity under certain working condition, design artefact should not have the engineering stress value that is beyond the yield stress of the material. For another example, due to the restrictions of the manufacturing capability, design artefact might be required to have certain slot patterns instead of random slot patterns, even when the slot patterns are, for example, “optimized” from topology optimization. Constraints in the design process in general manifest themselves in the structure of design artefacts and naturally in their CAD models. During the CAD modeling procedure, they need to be determined by engineers. Unfortunately, the constraint mapping usually lacks a systematic approach.

Constraints in CAD, on the other hand, are primarily spatial constraints with different types, i.e., geometric constraints, dimensional constraints, and assembly constraints. Examples of geometric constraints include vertical, horizontal, and parallel, which restrict the relative position of sketch objects with respect to a reference. The reference could also be a sketch object. Dimensional constraints determine the size of a sketch object or form feature, for example, specifying the width and length of a rectangular. Assembly constraints are used to position different parts with respect to each other in an assembly structure to reduce their degrees of freedoms, which include touch, concentric, distance, fix, etc.

Constraints in different forms, i.e., continuous, discrete, and mixed constraints [47], are embedded in the functional feature. On the one hand, in the geometric representations of abstract geometry features certain kinds of geometric or dimensional constraints are applied. On the other hand, engineering design considerations impose some forms of constraints in the parameters of functional feature from, for example, physics feature modeling. i.e., constraints from different aspects are handled separately but integrated into functional feature modeling.

Parameterization is a key to link the constraints in the design domain and during the CAD modeling process, as depicted in Fig. 8. Constraint management should work hand in hand with parameterization. The capability of parameterization of features in the CAD system makes it easier to manage the parameters involved in the design process. Besides defining parameters to control the behaviors of form features, other groups of parameters could also be defined. For example, functional parameters that are used to describe the function requirements of the design, physics parameters, which define parameters relating to physics phenomena or entities, and abstract geometry parameters, which reflect key characteristic shapes. Parameters could be named, assigned with values, constrained (e.g., apply interval constraints), and even combined with logic expressions (e.g., *if* . . . *then* . . .) [52]. Given a product, there are multiple approaches to apply constraints and parameterization schemes. The current research takes a functional approach intentionally to choose those constraints and



(a) (b)

Fig. 7. Design abstraction and embodiment of geometries of different levels.

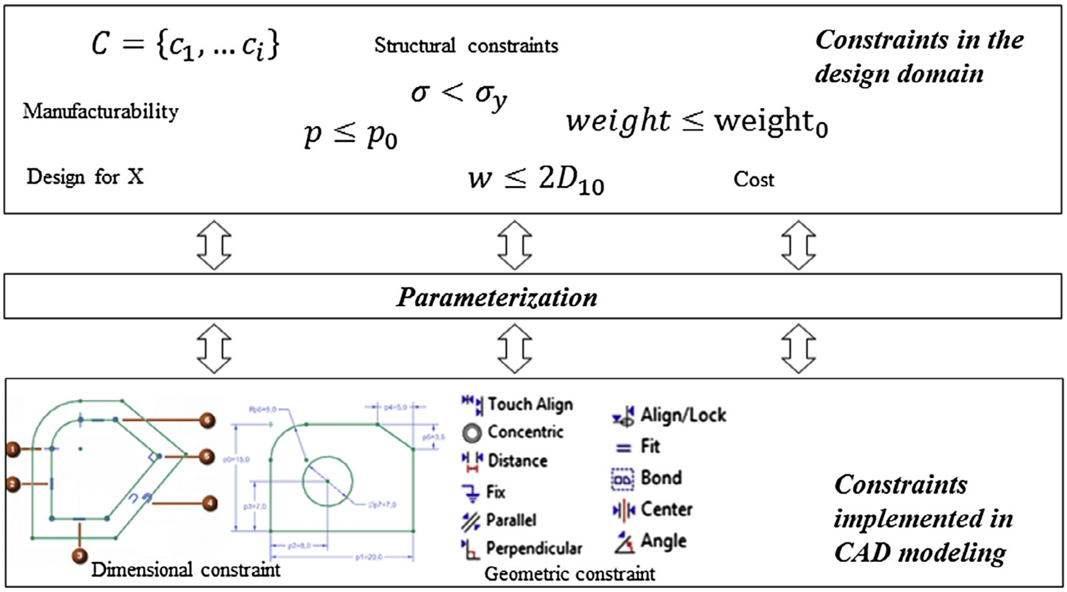


Fig. 8. Parameterization of design constraints in CAD modeling.

parameterization schemes of the CAD modeling that reflect the functional considerations of the design.

Parameterization and constraints defined in abstract geometry features could be referred to during the detailed part model construction. There are two possible scenarios. First, when abstract geometry features are imported directly into the detailed part model, the constraints and parameterizations are accessible by the detailed part model since abstract geometry features are within the same part file. Second, when abstract geometry features are put into separate part files, detailed part can refer to them through technologies like WAVE and inter-part expressions [50]. WAVE can link, both associatively and non-associatively, geometry elements like bodies, curves, datum, faces, and points, between two part files and get information about the linked geometry and parts, including parameters and applied constraints. In this approach the relationships between the detailed part file containing linked geometry and the original part file of the abstract geometry features are managed in an assembly-like structure, which might or might not be desirable depending on the application. For example, it is desirable if a part file containing abstract geometry features is to be used as a template part that defines necessary skeleton of the assembly structure. Inter-parts expressions allow links among expressions between different part files of abstract geometry features and detailed part model. Hence, the associativity and traceability of parameterization and constraints are maintained such that auto-update of detailed part model could be achieved.

3.2. Proposed modeling procedure

With the theoretical foundation laid down in the above section, the modeling procedure would be described briefly in this section. In Section 4 it would be discussed more extensively with a case study. A brief schematic overview of the general modeling procedure is shown in Fig. 9. It mainly consists of three major steps, i.e., functional analysis, abstract geometry features modeling, and CAD part geometry synthesis.

3.2.1. Functional analysis of the design artefact

Apply approaches like QFD, DSM and AD [53,55,17] to identify the required functionalities for the design artefact. Domain knowledge could be elicited from experts or from existing documents/patents. Functional decompositions need to be carried out to decompose more general functions into smaller granularities. The resulting functions could be in the form of a tree structure. Next step is to identify the key parameters, both geometrically and non-geometrically. Then build the relations among geometric and non-geometrically related parameters with feature parameter maps [62]. Finally, identify the functional faces ([48,18]) or other key characteristic geometries required to perform the functions.

3.2.2. Abstract geometry features modeling

Model the abstract geometry features that enable the design functionalities. References, constraints and parameterizations within the abstract geometry feature should be well organized accordingly. Next, identify the relations among those abstract geometries spatially. For example, a few abstract geometry features might need to be formed within a single solid part. In this case, the spatial relations among the abstract geometry features need to be considered, which is done preferably through their references to make them well constrained and parameterized. In some scenarios the spatial relations need to be determined with other design considerations. Parameters within abstract geometry features should be named meaningfully. The constructed abstract geometry features can be made into UDFs. If more flexibility is required, they can be programmed as UDOs or saved as separated files. They could be placed into a feature library and are reusable for future design activities.

3.2.3. Detailed CAD part modeling

This is the stage that performs the modeling activities to construct the CAD model for detailed design. After identifying the abstract geometry features, it is often not easy to synthesize abstract geometry features into fully fledged CAD model with current CAD tools at hand. Ideally the synthesis process is straight forward by simply combining the abstract geometry features together with proper positioning and additional feature operations. Instead, designers must make use of available modeling operations to construct the geometry where the abstract geometry features and their spatial relations are embedded. One might not use the exactly same modeling operations used in modeling the abstract geometry features, or make direct use of abstract geometry

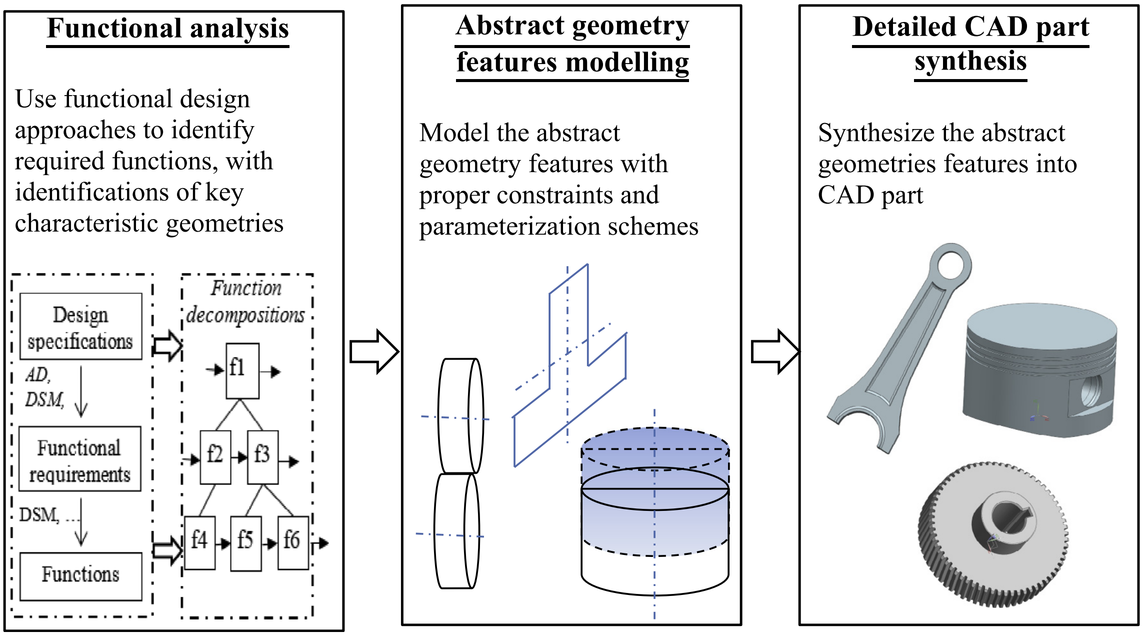


Fig. 9. Schematic overview of the proposed modeling procedure.

features, to model the corresponding geometry during the detailed CAD model construction. Nevertheless, the approach of parameterizations and constraints identified above provide guidance for this stage.

4. Case study

Design of a piston part is taken as an example in the current research. Note that the case study is not meant to be inclusive but to offer a demonstration of functional modeling of CAD with a few representative functional considerations. It can be observed from the case study that part faces are commonly used to reflect abstract geometry features. However, this does not mean it is restricted to faces only. Other geometric entities can be applicable, for example, solids, edges, vertices as well as constructive elements, such as sketch elements, datum planes, center point positions, and feature dimensions. Non-geometric entities like attributes, derived parameters and constraints can also be used.

4.1. Functional analysis of a piston

As discussed above, engineers should embed the functional considerations of the part being modeled in the CAD modeling process such that the resulting model is functionally robust. The main function of piston used in engine is *to transfer force* from expanding fuel in the cylinder to the crankshaft with piston rod, the function of which could be further decomposed. Moreover, to prevent the combustion gases from bypassing the piston, *sealing* need to be considered as well, which will be handled with the help of metal rings, or piston rings, around the piston. From the above discussion, it could be seen that in order to function well for the piston, it needs to have following functional faces

(a) A functional face to interact with fuel, i.e., compressing and expanding. Henceforth denoted as f_1.
(b) Grooves on which piston rings could be placed. These serve multiple functions. For example, those piston rings seal the combustion chamber to prevent gases from leaking to the crank, and support heat transfer from the piston to the cylinder wall. Henceforth denoted as f_2.
(c) A functional face to connect with the connecting rod, i.e., through piston pin. Henceforth denoted as f_3.

Note that the functional analysis examples listed above are not meant to be complete. In this scenario, a piston is not seen as a standalone object but is put into a context that it could interact with other parts of the whole system to perform certain functions. The key of the interactions lies in the geometry of the product, the functional faces, to be more specific.

4.2. Abstract geometry features modeling

Based on the identified functions, abstract geometry features could be modeled, as is shown in Fig. 10. For example, for f_1 the geometrical representation of the abstract geometry feature is a circular surface, and for f_3 a cylindrical surface, with corresponding references, parameters, and constraints. For example, it is clear that a circular surface could be parameterized by its diameter or radius and referenced by a coordinate system, and a cylindrical surface parameterized by its diameter or radius and its length and referenced by its own axial. There might be more than one ways to reference or parameterize abstract geometry features, depending on the requirements on the restriction of the corresponding degree of freedoms. Constraints could be applied to build up the relations among the parameters. The fundamental is that each abstract geometry feature is self-contained in the sense it is properly parameterized and well constrained with appropriate references. Since parameters are named they are easy to be identified and changed if needed. The model should adhere to the functions such that once upstream functional requirement is changed it should also be updated accordingly.

The spatial relationships among abstract geometry features also need to be considered. For example, top land, the distance between the edge of the piston crown and the top side of the first piston ring groove, p_1, is a critical parameter in the design of the piston. The first piston ring is a compression ring. It requires a temperature range that needs to be compatible with its function. It is known that the value of p_1 is a compromise between different factors. For example, the piston is preferred to have low mass, which means to have a small p_1. However, p_1 also pertains to the function

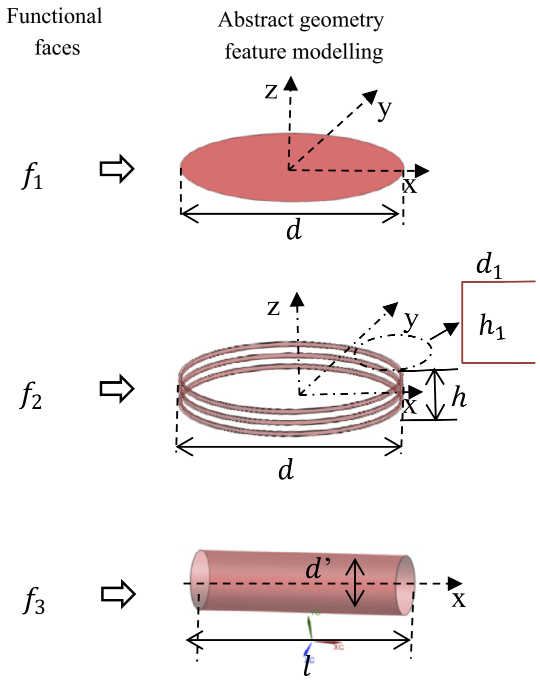


Fig. 10. A schematic of from functional faces to abstract geometry feature modeling.

of the first piston ring, which is further related to the compression process, material, etc. [46]. For another example, the compression height, the distance between the center of the piston pin and the upper edge of the top land p_2 is also critical in the piston design. It needs to be as small as possible to have a low mass. However, if the value is too small, it will result in higher temperatures in the pin bore and high stress on the piston crown, which is likely to give rise to cracks in the pin bore or the piston crown [46]. Such design considerations need to be transferred into the constraints among abstract geometry features and well parameterized, which in turn manifest into the detailed CAD model.

Since abstract geometry features are well constrained and parameterized, they are adaptable to new use cases. Different function attributes could be attached to abstract geometry features since same abstract geometry feature could carry different functional concepts.

4.3. Detailed CAD part modeling

With the identified abstract geometry features, the next step is to synthesize them into the detailed model. It is vital to understand that when materializing abstract geometry features in the detailed model designers might not be using the same modeling operations applied in constructing abstract geometry feature to model the corresponding parts with detailed geometry. In addition, there are also more than one modeling strategies to construct the detailed CAD geometry even with seemingly the same abstract geometry features. Certain entities of the abstract geometry features could be imported or linked into detailed CAD model to facilitate the model construction, e.g., geometry elements, parameters.

For example, Fig. 11 shows two example of embedding two of the abstract geometry features into the detailed CAD model. In Fig. 11(a) it is done separately. The edge of face f_1, is used directly to extrude into a solid. f_2 is manifested by revolving three rectangular shapes. Proper positioning and *Boolean* operations are applied to combine the resulting solids together. Not all details of referencing are shown in the figure in order to save space. It is doable but not optimal in the sense that the process could be synthesized more organically. Fig. 11(b) gives a better example of synthesizing abstract geometry features into detailed CAD model. At least two different abstract geometry features are combined into one sketch such that a single revolve feature could build up the required intermediate model whereas the previous example demands many more feature operations. Moreover, the synthesis of abstract geometry feature into the detailed model might not seem to be straight forward. For example, functional face f_1 is not materialized by extrusion of a circular with same diameter, as is shown in Fig. 11(a), but a revolution, as is shown in Fig. 11 (b). Parameterizations are applied properly such that the resulting faces have the same dimensions. The synthesis for f_3 could be carried out in a similar manner (see Fig. 12). It is desirable to point out that other than synthesizing abstract geometry features in the part model, it could also be shown that abstract geometry features are often the key to associate different parts together. As is shown in Fig. 13, face f_3 serves as an interface among different parts and it indicates certain kinds of assembly constraints required to position the parts. The associativity requires that when f_3 is changed the relevant parts should also be updated accordingly.

As mentioned in the stage of abstract geometry feature modeling, the spatial relations among abstract geometry features need to be considered beforehand and they need to be manifested into the detailed CAD model. The manifestation is achieved through either geometry association or proper constraints and parameterizations, or the combination of the both approaches. For example, the relationship predefined by p_1 is applied in Fig. 14 to define the top land and p_2 to define the compression height. In sum, Fig. 14 shows the schematic of the modeling process with abstract geometry features.

5. Discussion

The resulting CAD models built with the proposed method are functionally robust because functional considerations of design, manifested by function concepts carrier – abstract geometry features – are taken as modeling guidance with geometry associations, proper parameterizations and constraints management. The modeling of detailed CAD geometry is based on the synthesis of abstract geometry features, which in turn reflects design functionalities. Functional changes could be traced to abstract geometry features, or the relations among them, and then to the detailed CAD models. The traceability of functional changes into detailed CAD model makes it easier to carry out the functional design changes.

Table 1 provides some approaches toward engineering design. RFLP is a model-based system engineering approach that takes CAD models as one of the "Physical" (the "P" in RFLP) representations, i.e., the virtual solution. CAD models are components of system engineering process as a whole, assuming CAD models provide valid virtual representation without looking into the details of how to construct CAD models. The interest of this paper, in contrast, lies in the nitty-gritty details of how to construct CAD models that are robust enough to capture functional design knowledge. Of course, the results of our approach could be integrated into model-based system engineering process to construct the valid and robust CAD models. Knowledgeware from Dassault Systems supports parameter control with formulas. It is also capable of rule-based reasoning (*if-else-then*) and checking for constraint validation. Users can create intelligent and automated templates from a model. Knowledge Fusion from Siemens NX is a generative modeling language based on the principle of KBE. Geometric objects and pertaining operations can be carried out by Knowledge Fusion program with system and user classes. Both approaches can be used as

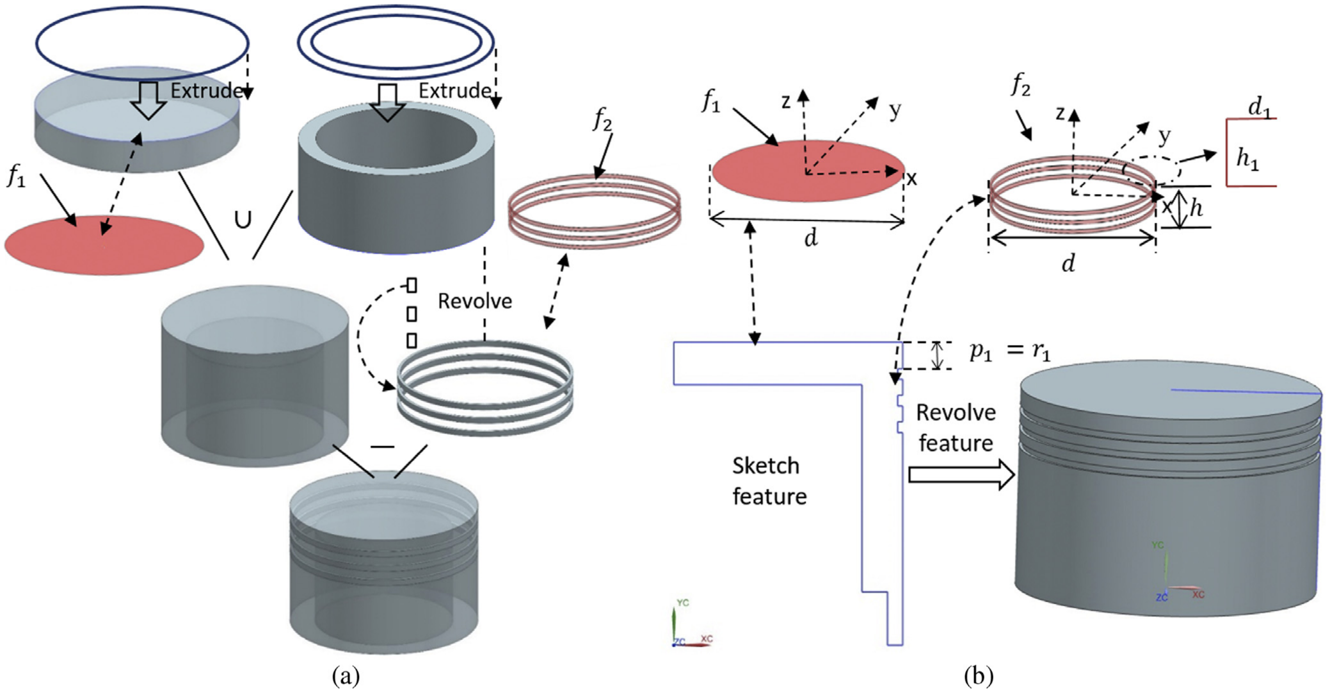


Fig. 11. Two different approaches to synthesize abstract geometry features into detailed CAD modeling.

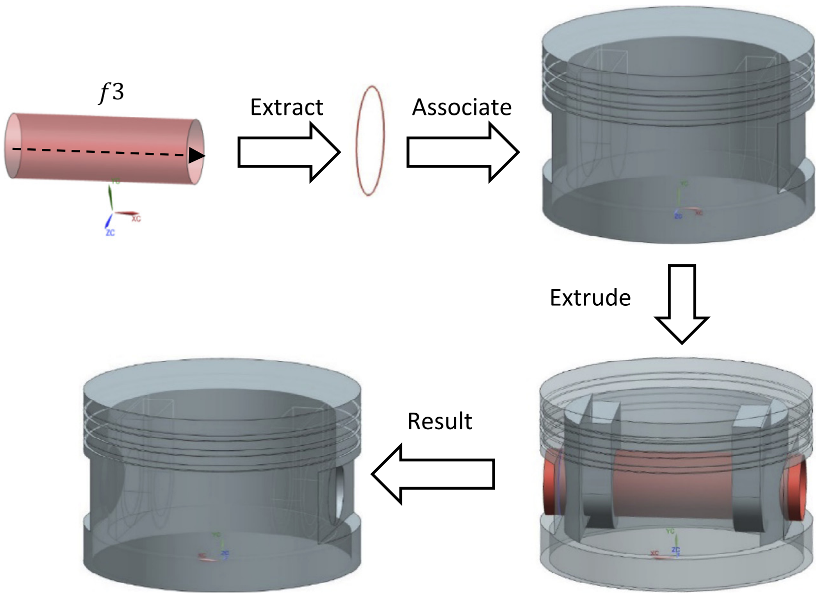


Fig. 12. Manifestation of $f3$ into the detailed CAD model.

means to implement our proposed approach. For example, abstract geometry features can be defined as templates with predefined slots for parameters or expressions. The proposed approach in this paper can actually be seen as a top-down modeling. The "top" in here means the modeling of function-driven abstract concept carriers, i.e., abstract geometry features; and the "down" is the detailed CAD modeling, including part modeling and assembly modeling.

CAD modeling strategies, for example, explicit reference modeling, resilient modeling, and horizontal modeling, are available to improve the efficiency of model construction. Horizontal modeling gets its name by trying to achieve a horizontally structured feature tree without long-chained feature dependencies by creating a bunch of datum planes after a base feature to eliminate the parent/child dependencies. However, as has been pointed out by Camba et al. [9], it is hard to express design intents in their feature tree. Realizing the problem of unstable CAD models, resilient modeling offers a solution to manage the sequence and structure of the feature tree by defining a collection of best practices. Features are organized in different groups according to their importance, function, and volatility, namely, reference features, construction features, core features, detail features, modify features, and quarantine features. However, their approach is still shape-oriented without functional considerations of the design itself.

Since both explicit reference modeling and our functional feature modeling approaches appreciate the importance of functions in the CAD modeling, detailed comparisons between those two will

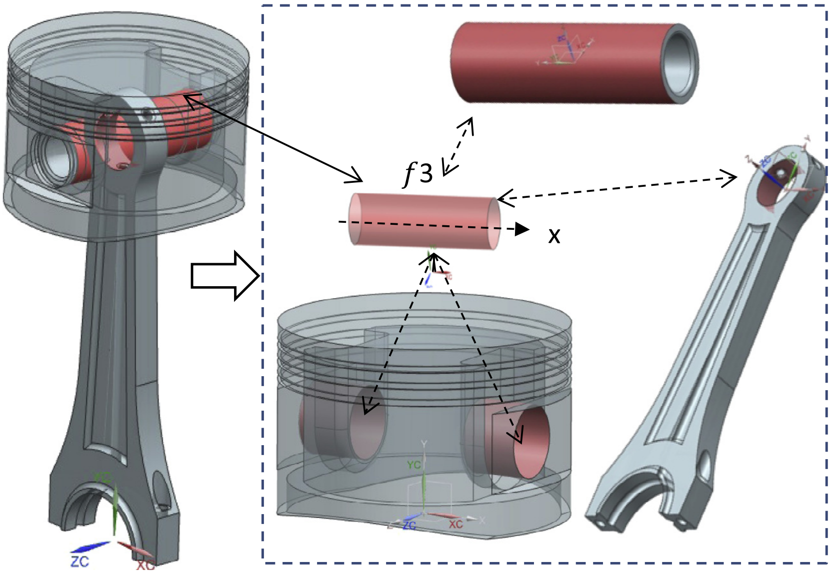


Fig. 13. An example abstract geometry feature is used as an interface to associate different parts.

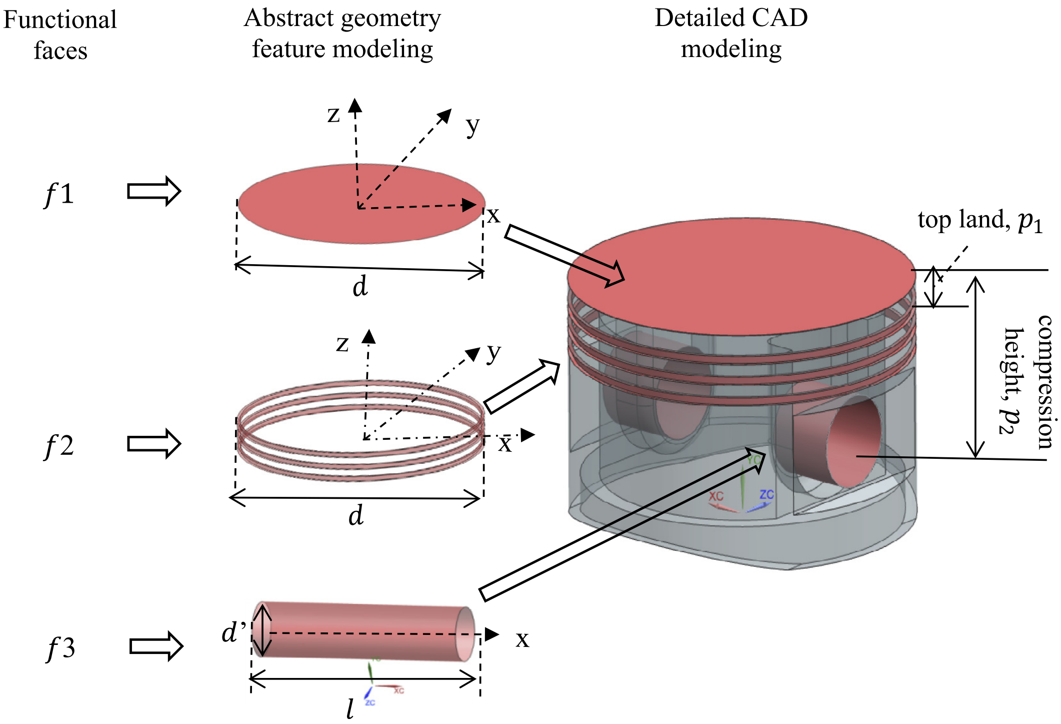


Fig. 14. A schematic of the modeling process with abstract geometry feature.

Table 1
Selected approaches toward engineering design related to CAD.

Approaches	Comments	Source
RFLP	Industrial implementation of system engineering approach	Dassault Systèmes [14]
Explicit reference, horizontal modeling, resilient modeling	CAD modeling methodologies focusing on reusability	Bodein et al. [6], Gebhard [22], Landers and Khurana [33]
Knowledgeware, knowledge fusion	Industrial implementation of knowledge-based engineering	Dassault Systèmes [14]; Siemens NX [50]; Amadori et al. [3]
Top down modeling	Assembly design, KBE	Gao et al. [20], Amadori et al. [3]

be given. Table 2 briefly organizes the key differences between explicit reference modeling and the proposed functional feature modeling with abstract geometry features.

First of all, both approaches emphasize on the uses of references. In explicit reference approach, the references are mainly referred to as those elementary parametric elements, which could

Table 2
Comparison of explicit reference and functional feature modeling approaches.

	Explicit reference modeling approach [6]	Suggested functional feature modeling approach
References	• Parametric elements, e.g. points (instead of vertex), plane, or surface (instead of face) • Creating references for functional areas	• Datum coordinate systems, datum point, and datum plane • Referential parameters, as well as references converted from sketch elements
Parameters	• Embedding in the parameterized references • Otherwise not clear	• Constraining reference elements • Representing dimensions of the abstract geometry features, functional parameters, and principle parameters, etc. • Feature parameter maps
Constraints	• Mandatory and non-mandatory constraints • Geometric related constraints • Creating features close to their primitive for mandatory constraints with implicit references	• Geometrically and non-geometrically related constraints • Containing constraints during design process as well as constraints involved in geometry creation in CAD • Parameterization
Geometries related to functions	• Functional areas, which are solids, resulting from functional analysis • Not clear what to do when multiple functions coexist in part of the same solid region	• Abstract geometry features • Both manifold and non-manifold geometry • Not necessarily solid, could be point, surface, volume, etc.
Solid product geometry	• Applying Boolean operations on functional areas (solid only)	• Synthesizing abstract geometry features in the detailed modeling process

be points, planes, surfaces, etc. On the other hand, the concept of references in functional feature modeling is broader in the sense that it includes datum coordinate systems, datum points, datum axis, and datum planes. It seems that parameterization is not the focus in explicit reference modeling other than being embedded in the references. Parameterizations of the references, as well as other geometric entities, in abstract geometry features and functions are encouraged in the functional feature modeling, which are organized with feature parameter maps [62].

In explicit reference modeling, constraints are categorized into mandatory and non-mandatory ones and they suggested using references instead of current shapes to apply constraints in the non-mandatory cases whereas building features close to their primitives in the mandatory cases. It can be seen that constraints in their approach mainly refer to geometric related constraints. Functional feature modeling includes both geometrically and non-geometrically related constraints; for example, geometric constraints, dimensional constraints, and assembly constraints are geometrically related constraints, whereas constraints applied to functional related parameters are non-geometrically related constraints. That is to say, constraints in functional feature modeling have richer engineering semantics.

In explicit reference modeling, solids for each function are constructed independently and then combined together by using *Boolean* operations, which makes it unclear what they do when overlapping functions exist in a given solid region. In functional feature modeling, since abstract geometry features, as abstract concept carriers, are not necessarily solid, *Boolean* operations alone are not enough. The current method is to choose the best approach to integrate or synthesize abstract geometry features into the detailed solid model construction with proper constraints and parameterization, instead of depending on *Boolean* operations alone. Thus, functions can not only be traced down to solid region in CAD model but also in abstract geometry forms.

As indicated in Bodein et al. [6], it is difficult for a designer to apply a generic modeling concept during the design phase. Additional trainings are required before designers can apply their proposed CAD modeling methodology. It applies in current case as well. Meanwhile, since simply building the shape of product is good but not good enough, design thinking must also be instilled into the CAD practitioners and reflected in the model building process. CAD education in universities should not only focus on teaching the CAD software, but also address the functional modeling methodology, which helps to build functionally robust CAD models.

The proposed method might sacrifice some easiness during the model creation but boosts functional knowledge capture and manifestation, and facilitates the design changes implementation. It might not be the easiest method to create the shape of the model but it strives to construct the models that are robust and ready for functional changes. On the other hand, CAD systems need to be enhanced to streamline the synthesis of abstract geometry features into fully fledged CAD model for detailed design based on the proposed modeling method.

6. Conclusion

This paper presents an in-depth and detailed description of a functional feature modeling method that entails how to construct robust CAD part model with abstract geometry features within functional feature modeling framework. A case study is presented to demonstrate the proposed method in an extensive manner. With multiple possible ways to construct the CAD model for a given product, a functional approach is believed to be effective to convey design intents. It could serve as a guideline for CAD practitioners to build functionally robust CAD model with smooth functional design change capabilities.

The main innovation of our approach lies in its function-oriented nature with a systematic modeling method. In particular, the proposed approach incorporates functional semantics into CAD models and narrows the gap between function-oriented design idealizations and procedurally-constructed CAD geometries. A systematic modeling procedure is presented with detailed description of modeling with abstract geometry features. Ideally, designers could start from the functional consideration, trace to its abstract geometry feature representation, which is further linked to the materialization in the fully fledged CAD model with associated geometries, parameterization, and constraints management.

One of the possible future research directions is to improve current CAD systems based on the proposed method, such that they could synthesize the convergence of design functions into solid part by maintaining the associativity of the abstract geometry features with part modeling operations. Although the proposed method could be carried out manually by users with existing CAD systems, a systematic implementation method and its detailed guidance to the end users could streamline the cyclic knowledge-rich engineering process and further improve the design efficiency and model reusability.

Acknowledgments

The authors would like to acknowledge Mitacs Canada and Canada Pump and Power (CPP) Pte Ltd for providing the industrial internship opportunity and financial support. We would also like to thank China Scholarship Council (CSC) and RGL Reservoir Management Inc. (Canada) for financial support. All the research works were carried out at the University of Alberta.

References

[1] M. Attene, F. Robbiano, F. Spagnuolo, B. Falcidieno, Characterization of 3D shape parts for semantic annotation, CAD Comput. Aided Des. 41 (10) (2009) 756–763, http://dx.doi.org/10.1016/j.cad.2009.01.003.

[2] N. Aifaoui, D. Deneux, R. Soenen, Feature-based interoperability between design and analysis processes, J. Intell. Manuf. 17 (1) (2006) 13–27, http://dx.doi.org/10.1007/s10845-005-5510-4.

[3] K. Amadori, M. Tarkian, J. Olvander, P. Krus, Flexible and robust CAD models for design automation, Adv. Eng. Inform. 26 (2) (2012) 180–195, http://dx.doi.org/10.1016/j.aei.2012.01.004.

[4] E. Van Den Berg, W.F. Bronsvoort, J.S.M. Vergeest, Freeform feature modelling concepts and prospects, Comput. Ind. 49 (2002) 217–233.

[5] R. Bidarra, W.F. Bronsvoort, Semantic feature modelling, Comput. Aided Des. 32 (3) (2000) 201–225, http://dx.doi.org/10.1016/S0010-4485(99)00090-1.

[6] Y. Bodein, B. Rose, E. Caillaud, Explicit reference modeling methodology in parametric CAD system, Comput. Ind. 65 (1) (2014) 136–147, http://dx.doi.org/10.1016/j.compind.2013.08.004.

[7] Y. Bodein, B. Rose, E. Caillaud, A roadmap for parametric CAD efficiency in the automotive industry, Comput. Aided Des. 45 (10) (2013) 1198–1214, http://dx.doi.org/10.1016/j.cad.2013.05.006.

[8] J.D. Camba, M. Contero, Assessing the impact of geometric design intent annotations on parametric model alteration activities, Comput. Ind. 71 (2015) 35–45, http://dx.doi.org/10.1016/j.compind.2015.03.006.

[9] J.D. Camba, M. Contero, P. Company, Parametric CAD modeling: an analysis of strategies for design reusability, Comput. Aided Des. 74 (2016) 18–31, http://dx.doi.org/10.1016/j.cad.2016.01.003.

[10] J. Camba, M. Contero, M. Johnson, P. Company, 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.

[11] A. Chakrabarti, T.P. Bligh, A scheme for functional reasoning in conceptual design, Des. Stud. 22 (6) (2001) 493–517.

[12] Xiang Chen, Shuming Gao, Youdong Yang, Shuting Zhang, Multi-level assembly model for top-down design of mechanical products, Comput. Aided Des. 44 (10) (2012) 1033–1048, http://dx.doi.org/10.1016/j.cad.2010.12.008.

[13] Zhengrong Cheng, Yongsheng Ma, Explicit function-based design modelling methodology with features, J. Eng. Des. 28 (3) (2017) 205–231, http://dx.doi.org/10.1080/09544828.2017.1291920.

[14] Dassault Systèmes, System Architecture Design & Simulation. Retrieved from <https://www.3ds.com/industries/high-tech/smarter-faster-lighter/systems-architecture-design-simulation/>, 2017.

[15] Y. Deng, Y. Ma, L. Shi, A behavioural process design model for development of assembly devices, Int. J. Prod. Dev. 18 (5) (2013) 445–460, http://dx.doi.org/10.1504/IJPD.2013.058451.

[16] F. Elgh, Automated engineer-to-order systems – a task-oriented approach to enable traceability of design rationale, Int. J. Agile Syst. Manage. 7 (3/4) (2014) 324–346, http://dx.doi.org/10.1504/IJASM.2014.065358.

[17] S. Eppinger, T.R. Browning, Design Structure Matrix Methods and Applications, The MIT Press, 2012.

[18] N. Frei, Der Konstruktionsprozess: Wirkmodelle basierend auf Constraints, PhD thesis, ETH Zurich, 2002.

[19] R.Y.K. Fung, Y. Chen, J. Tang, Estimating the functional relationships for quality function deployment under uncertainties, Fuzzy Sets Syst. 157 (1) (2006) 98–120, http://dx.doi.org/10.1016/j.fss.2005.05.032.

[20] S. Gao, Shuting Zhang, X. Chen, Y. Yang, A framework for collaborative top-down assembly design, Comput. Ind. 64 (8) (2013) 967–983, http://dx.doi.org/10.1016/j.compind.2013.05.007.

[21] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, et al., The status, challenges, and future of additive manufacturing in engineering, Comput. Aided Des. 69 (2015) 65–89, http://dx.doi.org/10.1016/j.cad.2015.04.001.

[22] R. Gebhard, A Resilient Modeling Strategy. Technical Presentation, Solid Edge University, 2013.

[23] J.S. Gero, Design prototypes : a knowledge representation schema for design, AI Mag. 11 (4) (1990) 26–36.

[24] J.S. Gero, U. Kannengiesser, The situated function-behaviour-structure framework, Des. Stud. 25 (4) (2004) 373–391, http://dx.doi.org/10.1016/j.destud.2003.10.010.

[25] A.K. Goel, S. Rugaber, S. Vattam, Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language, Artif. Intell. Eng. Des. Anal. Manuf. 23 (01) (2009) 23, http://dx.doi.org/10.1017/S0890060409000080.

[26] G. Gouaty, L. Fang, D. Michelucci, M. Daniel, J.P. Pernot, R. Raffin, M. Neveu, Variational geometric modeling with black box constraints and DAGs, CAD Comput. Aided Des. 75–76 (2016) 1–12, http://dx.doi.org/10.1016/j.cad.2016.02.002.

[27] G.P. Gujarathi, Y.-S. Ma, Parametric CAD/CAE integration using a common data model, J. Manuf. Syst. 30 (3) (2011) 118–132, http://dx.doi.org/10.1016/j.jmsy.2011.01.002.

[28] K. He, Z. Chen, J. Jiang, L. Wang, Creation of user-defined freeform feature from surface models based on characteristic curves, Comput. Ind. 65 (4) (2014) 598–609, http://dx.doi.org/10.1016/j.compind.2014.01.011.

[29] C.M. Hoffmann, R. Joan-Arinyo, On user-defined features, Comput. Aided Des. 30 (5) (1998) 321–332, http://dx.doi.org/10.1016/S0010-4485(97)00048-1.

[30] G.R. Iyer, J.J. Mills, Design intent in 2D CAD: definition and survey, Comput.-Aided Des. Appl. 3 (1–4) (2006) 259–267.

[31] J. Jiang, Z. Chen, K. He, A feature-based method of rapidly detecting global exact symmetries in CAD models, CAD Comput. Aided Des. 45 (8–9) (2013) 1081–1094, http://dx.doi.org/10.1016/j.cad.2013.04.005.

[32] A.M. Keuneke, Device representation the significance of functional knowledge, IEEE Expert 6 (2) (1991) 22–25.

[33] D.M. Landers, P. Khurana, Horizontally-structured CAD/CAM Modeling for Virtual Concurrent Product and Process Design, US Patent, 6, 775, 581, 2004.

[34] S.H. Lee, A CAD–CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques, Comput. Aided Des. 37 (9) (2005) 941–955, http://dx.doi.org/10.1016/j.cad.2004.09.021.

[35] Y. Li, W. Wang, X. Liu, Y. Ma, Definition and recognition of rib features in aircraft structural part, Int. J. Comput. Integr. Manuf. 27 (1) (2014) 1–19, http://dx.doi.org/10.1080/0951192X.2013.799784.

[36] B.T. Lin, C.K. Chan, J.C. Wang, A knowledge-based parametric design system for drawing dies, Int. J. Adv. Manuf. Technol. 36 (7–8) (2008) 671–680, http://dx.doi.org/10.1007/s00170-006-0882-y.

[37] B.T. Lin, S.H. Hsu, Automated design system for drawing dies, Expert Syst. Appl. 34 (3) (2008) 1586–1598, http://dx.doi.org/10.1016/j.eswa.2007.01.024.

[38] R. Lipman, J. Lubell, Conformance checking of PMI representation in CAD model STEP data exchange files, CAD Comput. Aided Des. 66 (2015) 14–23, http://dx.doi.org/10.1016/j.cad.2015.04.002.

[39] L.D. Miles, Techniques of Value Analysis and Engineering, McGraw-Hill, New York, 1972.

[40] Y.-S. Ma, G.A. Britton, S.B. Tor, L.Y. Jin, Associative assembly design features: concept, implementation and application, Int. J. Adv. Manuf. Technol. 32 (5–6) (2007) 434–444, http://dx.doi.org/10.1007/s00170-005-0371-8.

[41] Y.-S. Ma, T. Tong, Associative feature modeling for concurrent engineering integration, Comput. Ind. 51 (1) (2003) 51–71, http://dx.doi.org/10.1016/S0166-3615(03)00025-3.

[42] S.M. Manbub Murshed, J.J. Shah, V. Jagasivamani, A. Wasfy, D.W. Hislop, OAM+: an assembly data model for legacy systems engineering, in: Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2007, 2007, pp. 869–881, http://dx.doi.org/10.1115/DETC2007-35723.

[43] R.L. Nagel, R.B. Stone, R. Hutcheson, D.A. McAdams, Function design framework (FDF): integrated process and function modeling for complex systems, in: Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 2008, pp. 1–14.

[44] R.L. Nagel, K.L. Perry, R.B. Stone, D.A. Mcadams, Functioncad: a functional modeling application based on the function design framework, in: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2009, 2009, pp. 1–10.

[45] Z. Pan, X. Wang, R. Teng, X. Cao, Computer-aided design-while-engineering technology in top-down modeling of mechanical product, Comput. Ind. 75 (2016) 151–161, http://dx.doi.org/10.1016/j.compind.2015.05.004.

[46] Piston design guidelines, in: MAHLE Gmb (Ed.), Pistons and Engine Testing, vol. 1, Vieweg+Teubner Verlag, Wiesbaden, 2012. http://dx.doi.org/10.1007/978-3-8348-8662-0.

[47] A.J. Qureshi, J.-Y. Dantan, J. Bruyere, R. Bigot, Set based robust design of mechanical systems using the quantifier constraint satisfaction algorithm, Eng. Appl. Artif. Intel. 23 (2010) 1173–1186, http://dx.doi.org/10.1016/j.engappai.2010.02.003.

[48] U. Roy, B. Bharadwaj, Design with part behaviors: behavior model, representation and applications, Comput. Aided Des. 34 (9) (2002) 613–636, http://dx.doi.org/10.1016/S0010-4485(01)00129-4.

[49] J.J. Shah, M. Mantyla, Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications, John Wiley & Sons, 1995.

[50] Siemens NX, NX Knowledge Fusion for Designers. https://www.plm.automation.siemens.com/nl_nl/support/trainingen/overzicht/nx/nx-knowledge-fusion-designers.shtml, 2017.

[51] M. Sortino, S. Belfio, B. Motyl, G. Totis, Compensation of geometrical errors of CAM/CNC machined parts by means of 3D workpiece model adaptation, CAD Comput. Aided Des. 48 (2014) 28–38, http://dx.doi.org/10.1016/j.cad.2013.10.010.

[52] J. Stjepandić, W.J.C. Verhagen, H. Liese, P. Bermell-Garcia, Knowledge-based engineering, in: Josip Stjepandić, Nel Wognum, Wim J.C. Verhagen (Eds.), Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges, Springer International Publishing, Switzerland, 2015, pp. 1–839, http://dx.doi.org/10.1007/978-3-319-13776-6.

[53] N.P. Suh, Axiomatic Design: Advances and Applications, Oxford University Press, New York, 2001.

[54] P.C. Teoh, K. Case, Modelling and reasoning for failure modes and effects analysis generation, Proc. Inst. Mech. Eng. Part B: J. Eng. Manuf. 218 (B3) (2004) 289–300.

[55] V.H. Torres, J. Rios, A. Vizan, J.M. Perez, Integration of design tools and knowledge capture into a CAD system: a case study, Concurr. Eng. (2010), http://dx.doi.org/10.1177/1063293X10389788.

[56] Y. Umeda, M. Ishii, M. Yoshioka, Y. Shimomura, T. Tomiyama, Supporting conceptual design based on the function-behavior-state modeler, Artif. Intell. Eng. Des. Anal. Manuf. 10 (04) (1996) 275, http://dx.doi.org/10.1017/S0890060400001621.

[57] Y. Umeda, M. Ishii, M. Yoshioka, Y. Shimomura, T. Tomiyama, Supporting conceptual design based on the function-behavior-state modeler supporting conceptual design based on the function-behavior-state modeler, Artif. Intell. Eng. Des. Anal. Manuf. 10 (1996) (1996) 275–288, http://dx.doi.org/10.1017/S0890060400001621.

[58] Y. Umeda, S. Kondoh, Y. Shimomura, T. Tomiyama, Development of design methodology for upgradable products based on function–behavior–state modeling, Ai Edam 19 (03) (2005) 161–182, http://dx.doi.org/10.1017/S0890060405050122.

[59] H. Wang, Y. Zeng, E. Li, G. Huang, G. Gao, G. Li, ''Seen is solution" a CAD/CAE integrated parallel reanalysis design system, Comput. Methods Appl. Mech. Eng. 299 (2016) 187–214, http://dx.doi.org/10.1016/j.cma.2015.10.022.

[60] B. Welle, J. Haymaker, M. Fischer, V. Bazjanac, CAD-centric attribution methodology for multidisciplinary optimization environments: enabling parametric attribution for efficient design space formulation and evaluation, J. Comput. Civ. Eng. 28 (April) (2014) 284–296, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000322.

[61] S. Wiesner, M. Peruzzini, J.B. Hauge, K.-D. Thoben, Requirements engineering, in: Josip Stjepandić, Nel Wognum, Wim J.C. Verhagen (Eds.), Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges, Springer International Publishing, Switzerland, 2015, pp. 103–132, http://dx.doi.org/10.1007/978-3-319-13776-6.

[62] C.-G. Yin, Y.-S. Ma, Parametric feature constraint modeling and mapping in product development, Adv. Eng. Infor. 26 (3) (2012) 539–552, http://dx.doi.org/10.1016/j.aei.2012.02.010.

[63] L. Yuan, Y. Liu, Z. Sun, Y. Cao, A. Qamar, A hybrid approach for the automation of functional decomposition in conceptual design, J. Eng. Des. 4828 (June) (2016) 1–28, http://dx.doi.org/10.1080/09544828.2016.1146237.