COMPUTER-AIDED DESIGN & APPLICATIONS, 2018
VOL. 15, NO. 5, 684–696
https://doi.org/10.1080/16864360.2018.1441236

Graph centrality analysis of feature dependencies to unveil modeling intents

Zhengrong Cheng, Yanan Xie and Yongsheng Ma*

University of Alberta, Canada

ABSTRACT

Modeling intents are what engineers wish the part to be (even under changes) modeled in order to reflect his original design considerations, dependencies and constraints. Such intents in the current Computer-aided Design (CAD) modeling practice are not systematically captured, and often represented in the form of implicit constraints embedded in features. In order to unveil modeling intents it is necessary to analyze the feature relations. Feature dependency graph for a part model is created by extracting historical modeling operations and the dependency information of each feature. It offers a more organized view toward the model construction. A posterior analysis of CAD models is proposed to unveil modeling intents by examining feature dependencies with different graph measures, including degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality. This paper shows critical feature for the construction of the CAD model can be identified with the centrality analysis, which provides engineers a starting point to reexamine the modeling intents behind the model creation.

KEYWORDS

Modeling intents; Feature dependency graph; Network analysis; Feature-based design

1. Introduction

Feature-based CAD models are typically created with a series of operations with predefined features, e.g., block feature, hole feature, blend feature, and extrude feature. In most systems, sketch and *Boolean* operations are also treated as feature operations. CAD model reuse boosts product development process [5]. However, most of the created CAD models are not well reusable. Even with visually the same resulting CAD geometries the modeling procedures and operations applied might be quite diverse. Without a good understanding of the characteristics of model construction, it is hard to modify CAD models to cater to new design requirements. In some cases, even a single alteration of a certain value in the model could render the whole part unusable, which is even worse if they are not visually identifiable. The reason behind it, the authors believe, is the non-optimal and implicit modeling strategy in terms of the applied feature sequences to create the model. In order to reach a more robust CAD modeling strategy, a better understanding of the nature of the CAD model construction is necessary. The authors believe that the key lies in the understanding of modeling intents.

Modeling intents are different from design intents. Wang et al. use design intents in different levels, e.g., to represent design concepts, geometric properties and relations, referred to as features and constraints [24]. This paper distinguishes model intents and design intents. Here the authors treat design intents as what associate functions with product structures, i.e., the reasons why a product has specific structures. Design intents convey the functional design considerations to product structures. Modeling intents are restricted to those that are behind the CAD model construction, i.e., what users wish the model to be. There are two levels of modeling intents, i.e., the reasons why models are constructed in certain ways to, firstly, conform to the physical structures, and secondly, comply with functional design considerations (Fig. 1). One of the generally expected structural modeling intents is that minor (auxiliary) features should be built based on major features that contribute to the general shape of the product.

Understanding modeling intents is critical. If changes are about to be made to the model, it is better to know how and why the model has been constructed in certain ways such that when the changes are carried out the model will at least be able to regenerate. If the intents of model construction are unknown, it would be difficult to change the model properly due to its inner parametric and geometric associations [18,25]. Moreover, if the modeling intents are revealed, engineers can see whether the model has been constructed robustly by judging, for example, if the functional considerations have been conformed to. Unfortunately, modeling

CONTACT Yongsheng Ma yongsheng.ma@ualberta.ca; Zhengrong Cheng zcheng3@ualberta.ca; Yanan Xie yanan8@ualberta.ca

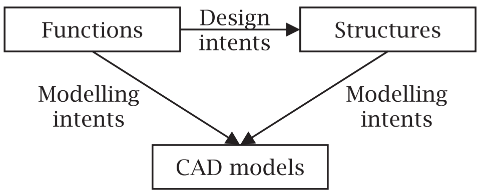


Figure 1. Relation between design intents and modeling intents.

intents are not explicitly expressed in the model. Modeling intents are usually reflected through the way how features are applied in the model construction process. Users, with the same set of feature operations at hand, might construct visually identical product geometry with different modeling procedures, which result in different feature dependencies [1]. So it is not enough to analyze the shape information. In order to unveil modeling intents in CAD models, the analysis of applied features is a way to go, more specifically, the analysis of feature dependencies.

The approach in this research is to retrieve implicit feature dependencies from the CAD model to construct an association graph from which further analysis is carried out. An algorithm is developed to retrieve feature information and to construct the dependency graph automatically. The graph provides a more organized view towards the applied modeling procedure. There are multiple aspects of the ADFDG that can be analyzed to unveil modeling intents. Current research focusses on finding the critical features in ADFDG to provide engineers with a starting point to examine the rightfulness of the modeling intents reflected by the model construction. Centrality related metrics, assessing each node's involvement in the walk structure of a graph [2,4], are applied to characterize the properties of feature dependency graphs.

The rest of this paper is organized as follows. Section 2 presents related works to this research, including feature-based CAD, a preamble to graph theory and its application in product design. Section 3 introduces the observed properties of feature dependency graph that could be made use of. It also presents the framework of system implementation with the algorithm to extract feature information from CAD models, which is designed based on the characteristics of how features are stored in the models. A few examples are studied in Section 4. Section 5 concludes the paper.

2. Related works

2.1. Feature-based CAD

CAD systems have gone through a long way from modeling with lower level geometric elements like points and lines in 2D environment to feature-based parametric 3D modeling, which makes it more convenient for designers to build digital prototype of products that could be further used in downstream design activities, e.g., engineering analysis [16]; hence they reduce time to market. In feature-based CAD, features are the building blocks of the geometric construction. Fig. 2 shows an example of UML diagram for implementing features in CAD, where builder pattern is applied. It could be seen that geometrical and positional data, which are parameterized, need to be set to build features. The richness of available features that represent different engineering semantics makes CAD systems more versatile. Other than system provided features, which are, in the sense of application domain, more general, User Defined Features (UDF) make it possible for end users to enrich the feature library by defining their own features based on their specific application domains [11,12]. The abilities of CAD systems to change feature attributes in the model to fit new design scenarios make part models reusable and further increase productivity, i.e., designer don't have to build CAD models from scratch to fit new design requirements. A working digital prototype might be just a few clicks away.

In the history-mode of CAD system, all feature operations are recorded and kept in the model history. Every feature knows its "children" and "parents". Note that this "children" is different from the parent-children relationships in the Object-oriented Programming. For example, in the software design, all specific features, like block feature and extrude feature, are the children classes of the more generic feature class (See Fig. 2). However, in the CAD modeling procedure, children feature mean those features are built based on the "parent" features because their existences are based on certain geometric aspects of the "parent "features. For example, when a hole feature is built based on an existing block feature, it makes the block feature a parent and the hole child feature. For another example, an extrude feature requires a section upon which to extrude, which makes the extrude feature depending on the section. These feature ownerships, or parent-children relations, are the origin of feature dependencies [1].

With the feature-based technology available to designers, many CAD model construction methods are possible. Camba et al. [5] examined three formal modeling methodologies, i.e., Delphi's horizontal modeling, explicit reference modeling, and resilient modeling, by comparing their advantages and disadvantages. Horizontal modeling tries to minimize the need to recreate or repair CAD model by eliminating the parent/child dependencies between features and build features on top of datum planes, instead of features. However, nowadays

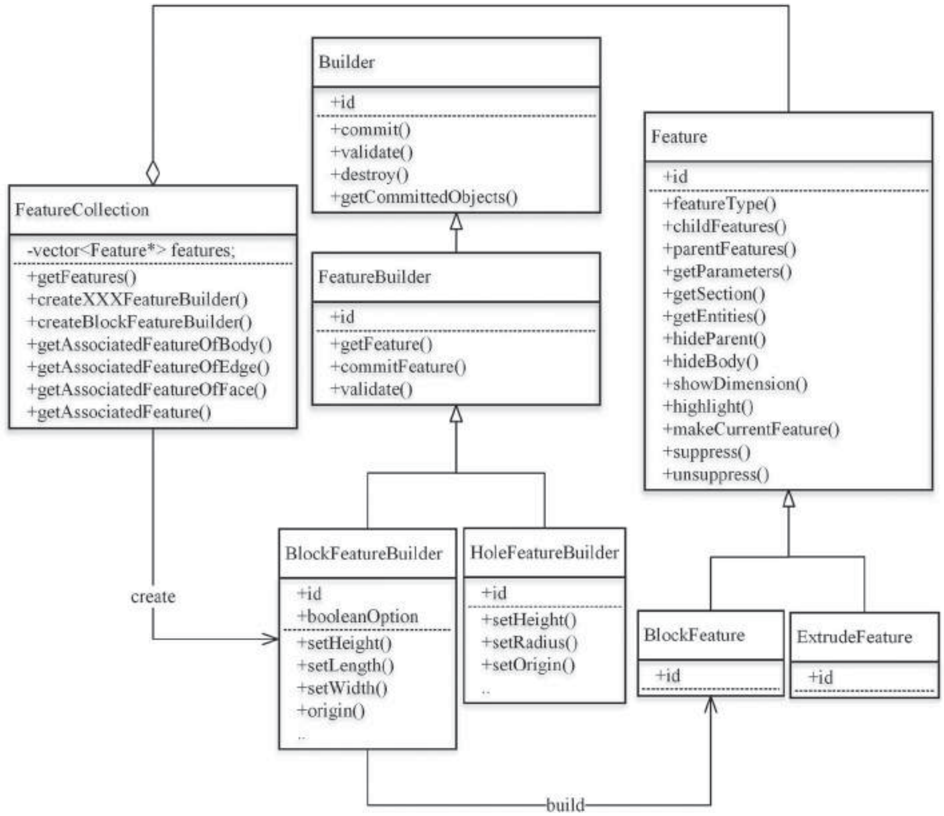


Figure 2. An example of feature implementation in CAD system using UML (extracted from Siemens NX API [21]).

datum planes are also treated as a type of features. Explicit reference modeling focuses on minimizing the number of constraints linked to existing geometry by managing functional references (which are solids). Resilient modeling aims to create a neutral solution to fix the problem of unstable design tree of CAD model by defining a collection of best practice methods. It categorizes features in the design trees by 6 different groups. It is anticipated that those different modeling methodologies would result in different feature dependency trees for a same product. Fig. 3 presents a general CAD modeling procedure [8]. It shows, generally speaking, minor features should be built on top of major features, which is a good rule-of-thumb.

2.2. Graph theory and its applications in product design

Graph, or network, has been used to describe product structures. [19] proposed propagation cost and clustered cost based on Design Structure Matrix (DSM) to compare the structures of different software designs. The propagation cost assumes equal cost among both direct and indirect dependencies, whereas clustered cost assigns different cost across cluster or module. Jaiswal et al. [13] introduced an assembly-based conceptual 3D modeling with unlabeled components, where a probabilistic factor graph was used to encapsulate the relationships between the unlabeled components in a shape database.

Graph theory has been applied in CAD retrieval for model reuse purposes. Graphs are used to represent the CAD models and the similarity between two 3D CAD models could be evaluated with graph matching algorithms. The graphs can be constructed using different elements of the CAD model. Some construct the graph from the shape of the CAD models with B-rep. For example, Tao et al. (2012) [23] uses a representation of face Attributed Relational Graph (ARG), where faces are taken as nodes and edges connecting the faces as arcs in the graph, created from a B-rep CAD model to convert partial retrieval problem into a subgraph matching problem. They also constructed a Face Adjacency Graph (FAG) from B-rep models and assess the model similarity by segmenting the graph into a set of regional graphs for subgraph matching [22]. On the other hand, [17] describes a reuse-oriented retrieval method for CAD models where modeling knowledge are captured in model similarity assessment with feature dependency directed acyclic graph and subgraph decomposition, i.e., their graphs were constructed based on the direct shape of the CAD model but the construction features. The resulting graph was used to simplify the CAD models

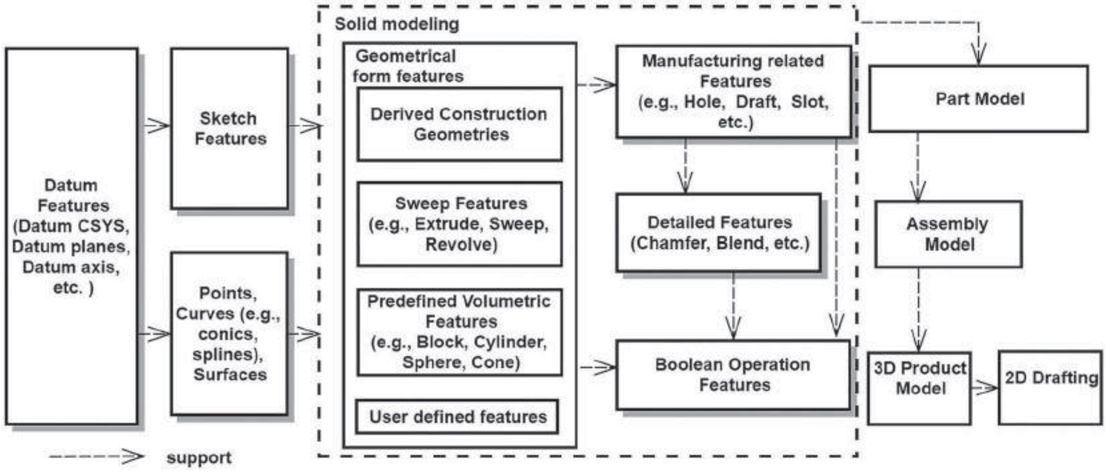


Figure 3. A general CAD modeling procedure [8].

such that shape histogram could be constructed based on the simplified CAD models.

Graph metrics have been used to characterize the key properties of products. [15] applied graph to model the structure and evolution of products. One of the case studies they used is a physical product and they take the different parts as nodes and physical connection as edges of the graph. The evolution of the product is captured by adding and removing of nodes and edges. A few network measures were employed to quantify the evolutionary characteristics of product structures, including average degree, degree distribution, density, clustering coefficient, average shortest path, etc. [6] applied a network approach to assess the impacts of changes on complex product, where a product is considered as a weighted network of parts, subassemblies, or subsystems. Three changeability indices, including degree-changeability, reach-changeability, and between-changeability were presented. However, the dependency relationship between parts is documented with approaches like interviewing experienced engineers, which is subjective and time-consuming.

Tab. 1 presents the formulas for some of the key graph centralities [2–3,10,14], where A denotes the adjacency matrix of the graph. For local measure, one can use degree centrality. Degree centrality of a node measures how many edges are connected directly to it. In a directed graph, degree centrality could be further categorized into in-degree and out-degree. The larger the number of in- or out-connecting edges is, the bigger the in- or out-degree value is. The value could be normalized by dividing by the maximum possible number of the connections. There are also global measures in the sense that they measure the centrality of a specific node relative to the rest of the network, e.g., closeness, betweenness, and eigenvector centrality [2,4,10]. Betweenness centrality is a measure to

Table 1. Formula to calculate graph centralities [2–3,10,14].

Degree centrality	Non-normalized	$C_D(i) = \deg(i)$
	Normalized	$C'_D(i) = \frac{C_D(i)}{N-1}$
Betweenness centrality	Non-normalized	$C_B(i) = \left[\sum_{j,k} g_{jk}(i)/g_{jk}\right]$
	Normalized	$C'_B(i) = \frac{C_B(i)}{(N-1)(N-2)}$
Closeness centrality	Non-normalized	$C_C(i) = \left[\sum_{j=1}^{N} d(i,j)\right]^{-1}$
	Normalized	$C'_C(i) = \frac{C_C(i)}{(N-1)}$
Eigenvector centrality and its variations		$\lambda e = Ae$
	Katz centrality	$C_{Katz} = ((I - \alpha A^T)^{-1} - I)1$
	Alpha centrality	$C_{Alpha} = (I - \alpha A^T)^{-1} e$
	Bonachich Power Centrality	$C(\alpha, \beta) = \alpha(I - \beta A)^{-1} A1$

quantify the number of times a node acts as a bridge along the shortest path between two other nodes, the value of which can also be normalized. In Tab. 1, g_{jk} is the total number of shortest paths from node j to k. $g_{jk}(i)$ is the number of those paths that pass through node i. Closeness centrality of a node is the reciprocal of the length of the total shortest path between the node and all other nodes in the graph. $d(i,j)$ is the shortest distance between node i and node j. Eigenvector centrality [3] measures the importance of a node by considering its neighbors' connectivity (or influence) as well as their subsequent downstream neighbors. The values of interests are contained in the eigenvector corresponding to the largest eigenvalue of the adjacency matrix of the graph. In Tab. 1., λ is the largest eigenvalue and e is the corresponding eigenvector of the adjacency matrix A. Another variation of the eigenvector centrality, applied in link analysis using hubs and authorities in information networks and World Wide Web [9], and in determining the design domination weights and design subordination weights in dependency analysis of design elements in the product development

Figure 4. From part to features.

[6], is to calculate the dominant eigenvector of, instead of the adjacency matrix of the graph, the multiplication of adjacency matrix with its transpose.

3. Framework and algorithm

It is observed that the feature dependency graph is directed and acyclic, which is called Acyclic and Directed Feature Dependency Graph (ADFDG), where the set of nodes, or vertices V, are the features and the set of edges E, depict the feature dependencies. Moreover, due to the nature of feature modeling, feature dependencies in ADFDG have other characteristics. Feature dependencies are non-reflexive, i.e., a feature cannot depend on itself. Feature dependencies are nonsymmetrical, i.e., two features cannot mutually depend on each other. Feature dependencies are transitive, i.e., if feature a depends on feature b and feature b depends on feature$c$, then feature a also depends on feature c [1,17].

A part model contains a lot more information than just features, for example, geometric information like bodies, faces, edges, etc., and non-geometric information like annotations, colors, and layers. Here what we care about are features in the part. There is a feature manager in the part model called feature collection [21], which is responsible for creating different feature builders to construct features, and keeping track of features that have been created in the part. All features within a part can be obtained from the feature collection, as is shown in Fig. 4. In addition, since each feature has pointers to those features depending on it, we can trace down the feature dependencies to build the graph. The general algorithm to extract feature information and to construct the feature dependency graph is shown in Fig. 6. The general framework of current research is presented in Fig. 5. It starts with a constructed feature-based CAD model with all the model history and feature information. Then feature information is extracted from the model with API programming to construct the ADFDG based on the algorithm introduced in Fig. 6. In general, three representations are available for a graph, adjacency list, edge list, and adjacency matrix. This research uses adjacency list representation for ADFDG by using a data structure called map, which is a type of associative container that stores key-value pairs. Note that some feature operations are created automatically by the CAD system in the background during the model construction. The resulting graph might show more feature nodes than what could be seen in the part navigator, where only explicitly applied feature operations are presented. With the ADFDG at hand, visualization and centrality analysis of the graph could be carried out.

Degree analysis of ADFDG examines how many incoming and outgoing edges each feature node has. It provides the direct dependency measurements for each feature operation. Out-degree of each feature node

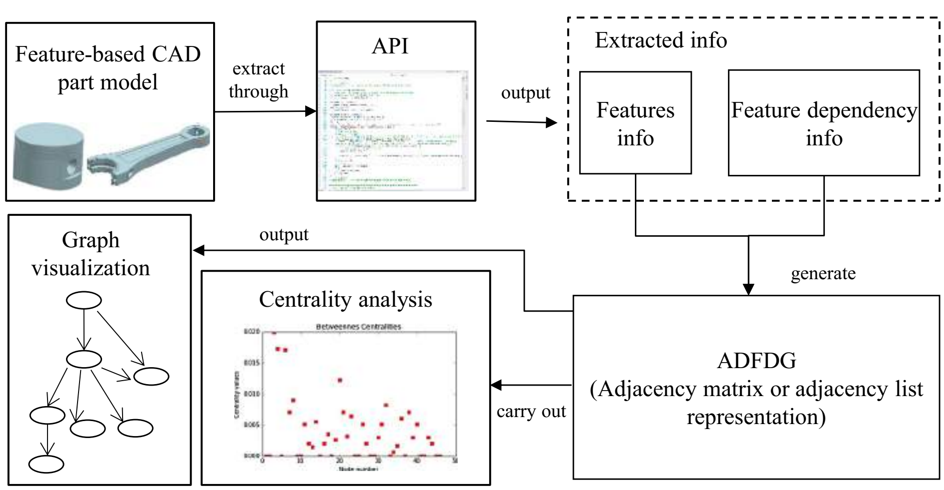


Figure 5. The general framework of current research.

<u>**Algorithm to create adjacency list representation of ADFDG**</u>

Initialization: Given a feature-based CAD part p, an empty set $V = \{\varnothing\}$, and an empty map $A = \{\varnothing\}$

for each feature f in the part p

    add feature f ID f^* to the set V

for each feature f_i^* in the set V, **do**

    create an empty list $L = [\varnothing]$

    for each immediate child feature cf belonging to the feature f, **do**

        insert the child feature ID, cf^*, into list L

    insert the pair $\{f^*, L\}$ into the map A

return the map A

Figure 6. Algorithm to construct ADFDG from feature-based CAD model.

indicates its influence on its direct neighboring feature nodes. In-degree of feature nodes shows their direct "parent nodes" they base upon. Out-degree centrality is more interesting here because we want to find the influencing features. Betweenness centrality helps to quantify the number of times a feature acts as a bridge along the shortest path between two other features. Features with higher betweenness values are those located in the "center" of the ADFDG. If we imagine the path with flows flowing through them, those features with high betweenness values mean that more flows are flowing through them, i.e., they are the busiest ones. Eigenvector analysis is calculated based on the matrix multiplication of adjacency matrix and its transpose, instead of adjacency matrix directly. As is shown in [7,9] that the meaning of dominant eigenvector of AA^T and A^TA are different. Here since the interest is in finding the features with dominant influences over other features, AA^T is chosen as the matrix based on which eigenvector analysis was carried out.

4. Results and discussion

Three examples will be studied in this section to prove the feasibility and effectiveness of the proposed approach. In section 4.1 a connection rod example will be discussed in detail. Section 4.2 will present the results for other two examples briefly.

4.1. Connection rod case study

A connection rod model of an inner combustion engine design is taken as an example in this subsection. This model was constructed by one of the authors. Fig. 7 gives the geometry of the connection rod in (b), its modeling history in (a), and the corresponding visualization of its ADFDG in (c). The adjacency list representation of the ADFDG for the connection rod case study is given in Fig. 8. Tab. 2 shows parts of the feature information for the model. Note that the tags for features are likely to change in different sessions. They are unique for different features within a session.

Table 2. Part of the extracted feature information for the connection rod model.

Index	Tag	Feature type	Feature name
1	38439	DATUM_CSYS	Datum Coordinate System(0)
2	38400	DATUM_CSYS	Datum Coordinate System(1)
3	38406	SKETCH	SKETCH_000:Sketch(1)
4	38407	EXTRUDE	Extrude(2)
5	38408	DATUM_CSYS	Datum Coordinate System(3)
6	38403	SKETCH	SKETCH_001:Sketch(3)
7	38404	EXTRACT_STRING	Linked Curve Object(4)
8	38405	EXTRUDE	Extrude(4)
9	46523	Mirror Feature	Mirror Feature(5)
10	46522	Instance Feature	Mirror Feature (5) / Instance[1][0]
11	38451	EXTRACT_STRING	Mirror Feature(5)
12	38450	EXTRUDE	Mirror Feature(5)
13	38452	DATUM_CSYS	Datum Coordinate System(6)
14	38453	SKETCH	SKETCH_002:Sketch(6)
15	38454	EXTRUDE	Extrude(7)

The results of different centrality analyses are provided in Fig. 10, where the horizontal axis gives node numbers and the vertical axis centrality values, which bring to our attentions of two major features shown in Fig. 9. Fig. 11 provides the correlations of those four centralities, where some key features are numbered and their correlation values are given in *c*. It is found that the connection rod model has a few dominant features. Out degree, eigenvector, and closeness centralities agree that the most critical features are feature 1, a datum feature, and feature 4, an extrusion feature based on the sketch feature 3. It might indicate that betweenness centrality might not be a good indicator for feature dependency graph in this case since it does not correlate well with other centralities. Those most critical features for

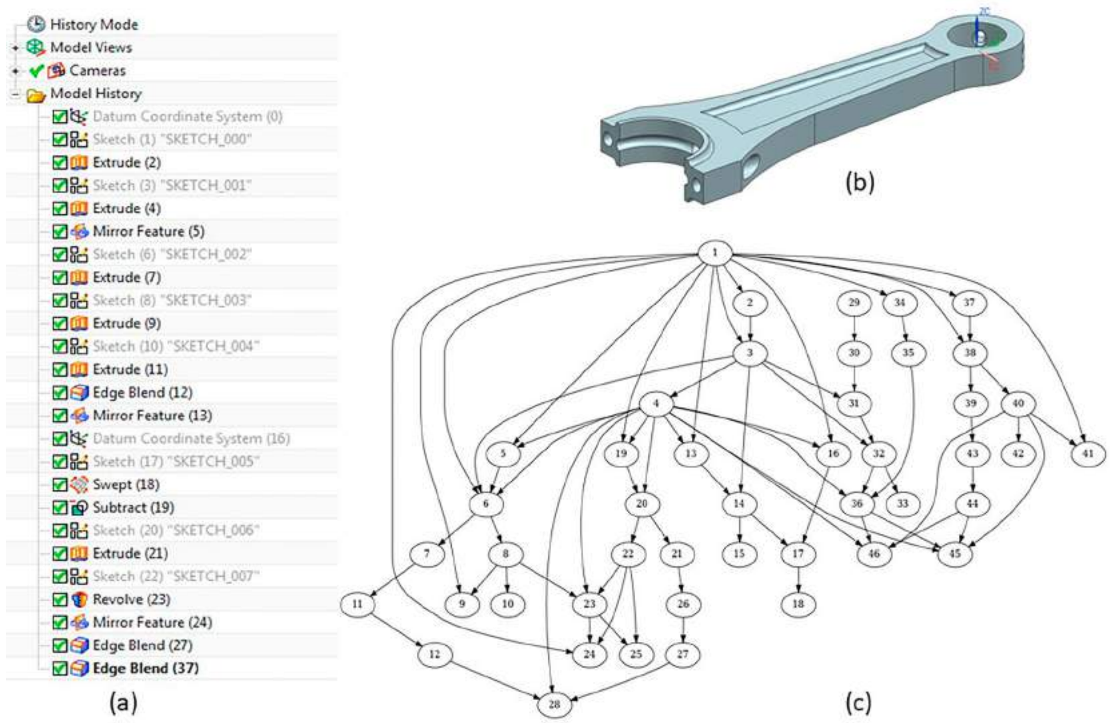
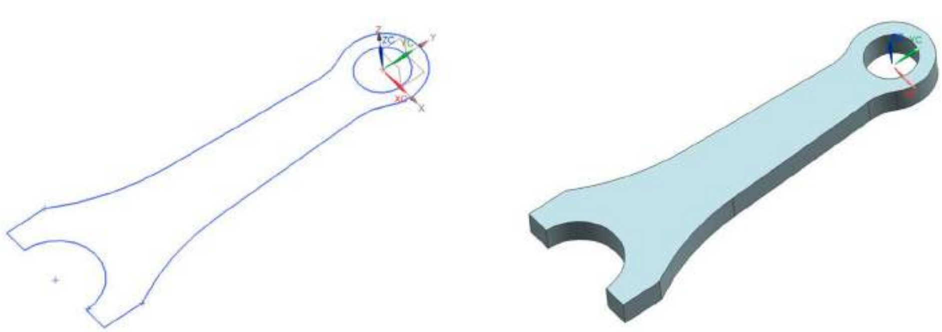


Figure 7. A connection rod case study with (a) model history, (b) CAD model, (c) visualization of its ADFDG.

Adjacency list representation: *{1: [2, 3, 5, 6, 9, 13, 16, 19, 24, 34, 37, 38, 41], 2: [3], 3: [4, 6, 14, 31, 32], 4: [5, 6, 13, 16, 19, 20, 23, 28, 36, 45, 46], 5: [6], 6: [7, 8], 7: [11], 8: [9, 10, 23], 9: [], 10: [], 11: [12], 12: [28], 13: [14], 14: [15, 17], 15: [], 16: [17], 17: [18], 18: [], 19: [20], 20: [21, 22], 21: [26], 22: [23, 24, 25], 23: [24, 25], 24: [], 25: [], 26: [27], 27: [28], 28: [], 29: [30], 30: [31], 31: [32], 32: [33, 36], 33: [], 34: [35], 35: [36], 36: [45, 46], 37: [38], 38: [39, 40], 39: [43], 40: [41, 42, 45, 46], 41: [], 42: [], 43: [44], 44: [45, 46], 45: [], 46: []}*

Figure 8. Adjacency list representation for the connection rod ADFDG.

connection rod are the ones constructed in the beginning of the modeling process, i.e., feature 1, 3, and 4, that influence the following feature operations a lot. It could be seen that it is reasonable because for connection rod many features are built on top of the features that generate the overall shape, which could be seen as a characteristic of the connection rod. It is predictable that for some other mechanical parts one might found more numbers of dominant shapes upon which smaller features are built. Hence, the resulting ADFDG and centrality analyses would be totally different. It could be said that on the one hand centrality analysis helps to



(a) At index 3 (b) At index 4

Figure 9. Two key non-datum features of the connection rod.

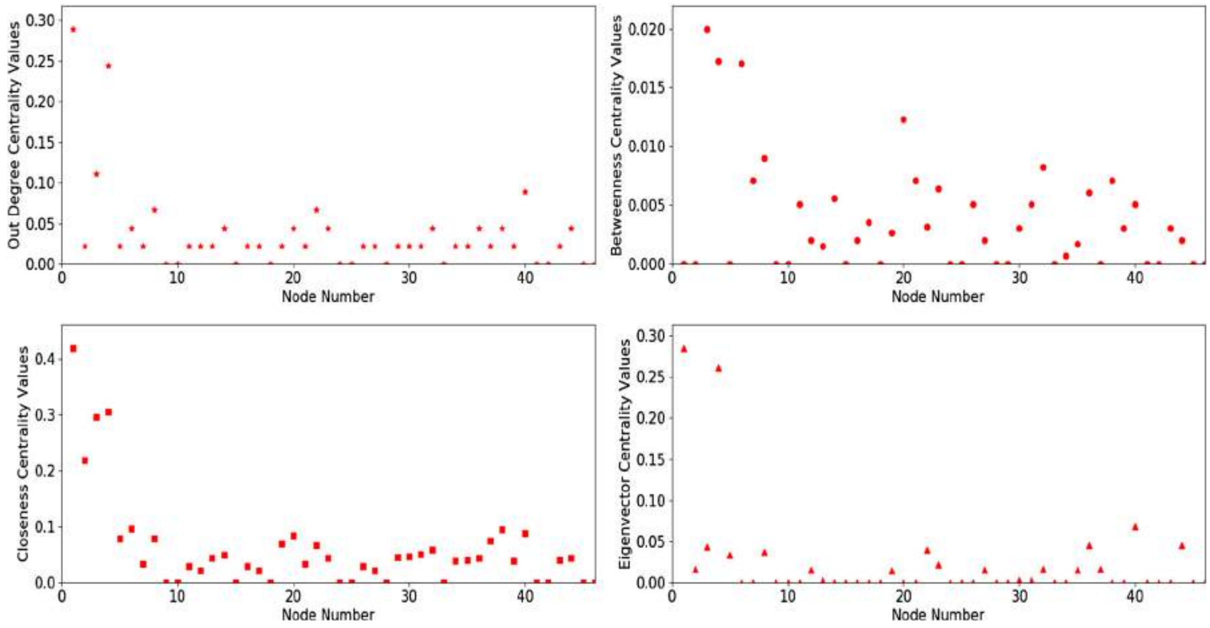


Figure 10. centralities of the connection rod case study.

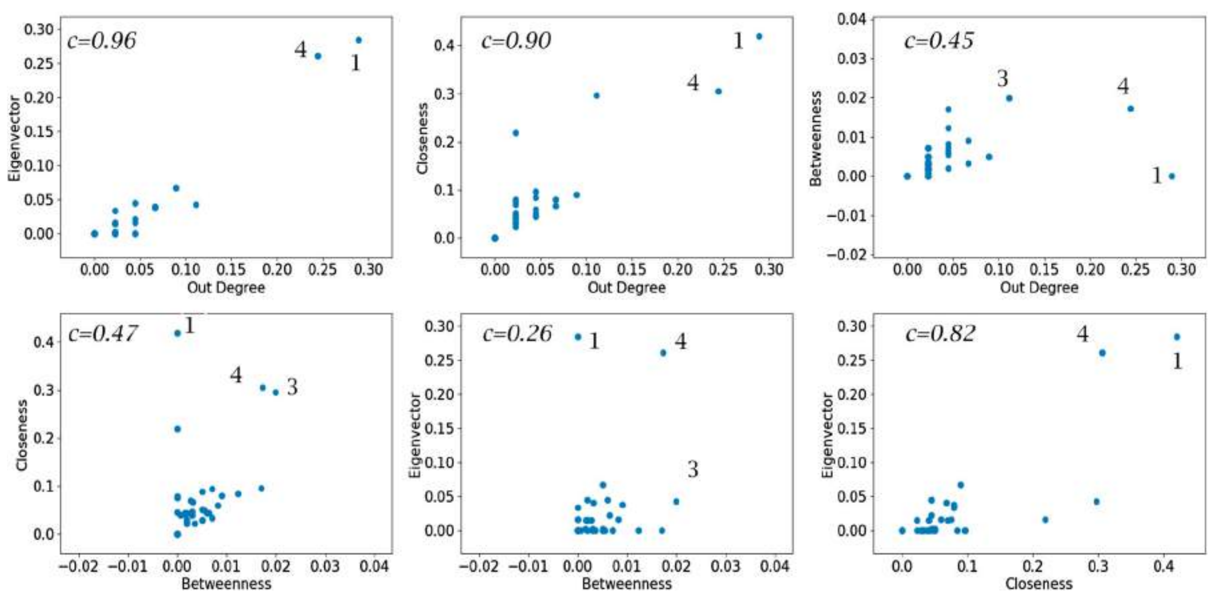


Figure 11. correlations of centralities of the connection rod example.

reveal critical features in the model construction, on the other hand, helps to identify the characteristics of the model. In could be seen that similar results could be obtained with the other two case studies in the next subsection.

4.2. Sport car seat and trigger switch cases

In order to demonstrate the generality of the proposed approach, two more cases are demonstrated in this subsection. Fig. 12 shows a sport car seat model in (b) and the virtualization of its ADFDG in (a). The sport car seat model is taken from GrabCAD [20], Tab. 3 presents part of the extracted feature information for the sport's car seat model. Fig. 13 shows some of the key features of this case study, identified by the centrality analysis results from

Fig. 14. These key features lay down the foundations for the auxiliary features to work on.

Fig. 15 shows a trigger switch CAD model in (a), and its corresponding ADFDG in (b). This is an example model from Siemens NX [21]. Tab. 4 shows part of

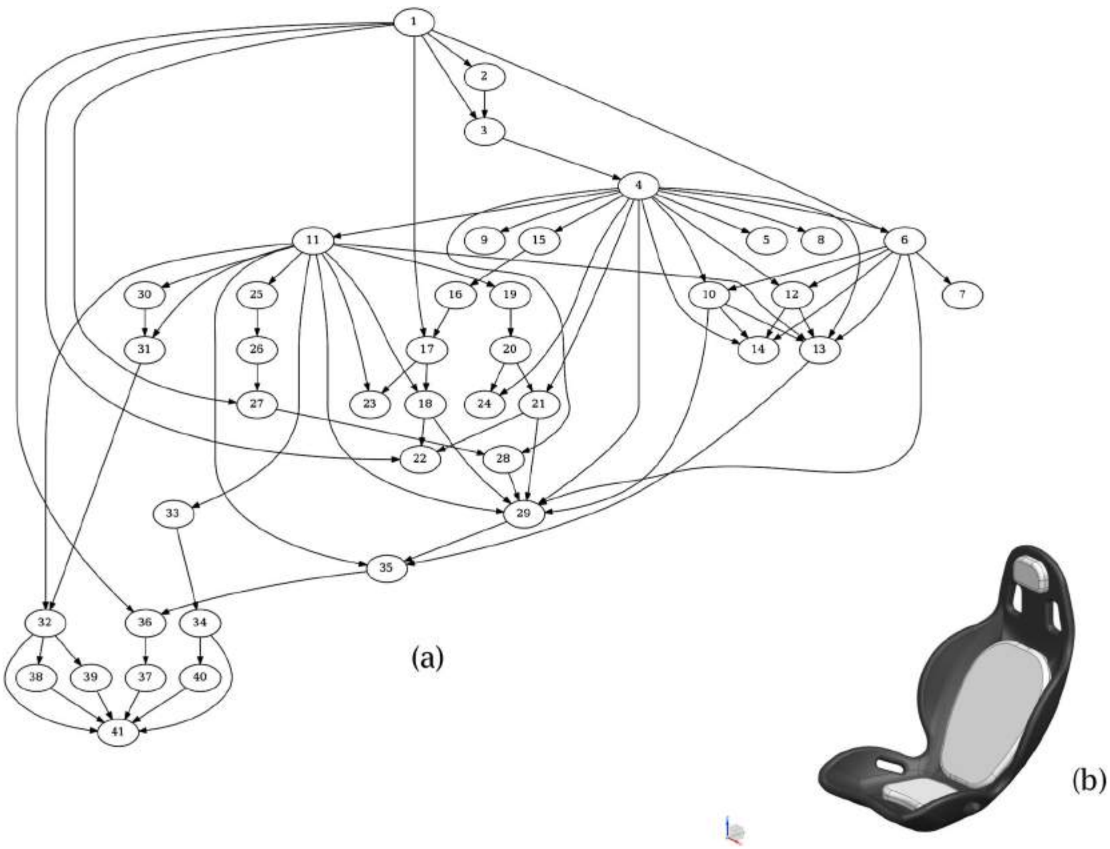


Figure 12. Sport's car seat model and its ADFDG.

Table 3. Part of the extracted feature information for the sport's car seat model.

Index	Tag	Feature type	Feature name
1	123910	DATUM_CSYS	Datum Coordinate System(0)
2	123938	DATUM_CSYS	Datum Coordinate System(1)
3	123930	SKETCH	SKETCH_000:Sketch(1)
4	123931	EXTRUDE	Extrude(2)
5	113377	BSURF_XFORM	X-Form(3)
6	113374	MIRROR	Mirror Body(4)
7	113376	UNITE	Unite(5)
8	113394	BSURF_XFORM	X-Form(6)
9	113395	BSURF_XFORM	X-Form(7)
10	107179	BLEND	Edge Blend(8)
11	123917	SHELL	Shell(9)
12	112800	BLEND	Edge Blend(10)
13	123924	BLEND	Edge Blend(11)
14	112694	CHAMFER	Chamfer(12)
15	123936	DATUM_PLANE	Datum Plane(13)

the extracted feature information for the trigger switch model. Fig. 16 gives the centrality values for the model, which helps to identify some key features shown in Fig. 17. It can be observed that the trigger switch model has two relatively separate major volumetric features (highlighted in Fig. 17 (b) and (c)), which is different from the above two case studies.

4.3. Discussion

The graphical representation of the feature dependencies provided by ADFDG offers engineers a more organized view toward the model construction, where interactions among feature operations are easily seen. Many graph properties are exploitable to give insights in understanding modeling intents for an individual model. Current research studies centrality properties of ADFDG, which reveals the information about which set of features are critical from the perspective of network topology.

Different metrics interpret the graph differently. They might not always agree on which features are important. Users need to choose the centrality metrics according to their application. For example, if one wants to find out which is the feature that other features directly depend on, he/she need to use the degree centrality. If one need also consider indirect dependencies, he/she could use one of the eigenvector centralities.

By looking at the identified critical features, engineers can start to ponder whether it is reasonable based on their engineering judgements, i.e., whether or not the applied modeling procedure is suitable or not. Ideally,

(a) At index 3 (b) At index 4 (c) At index 11

Figure 13. Some key figures of the sport's car seat model.

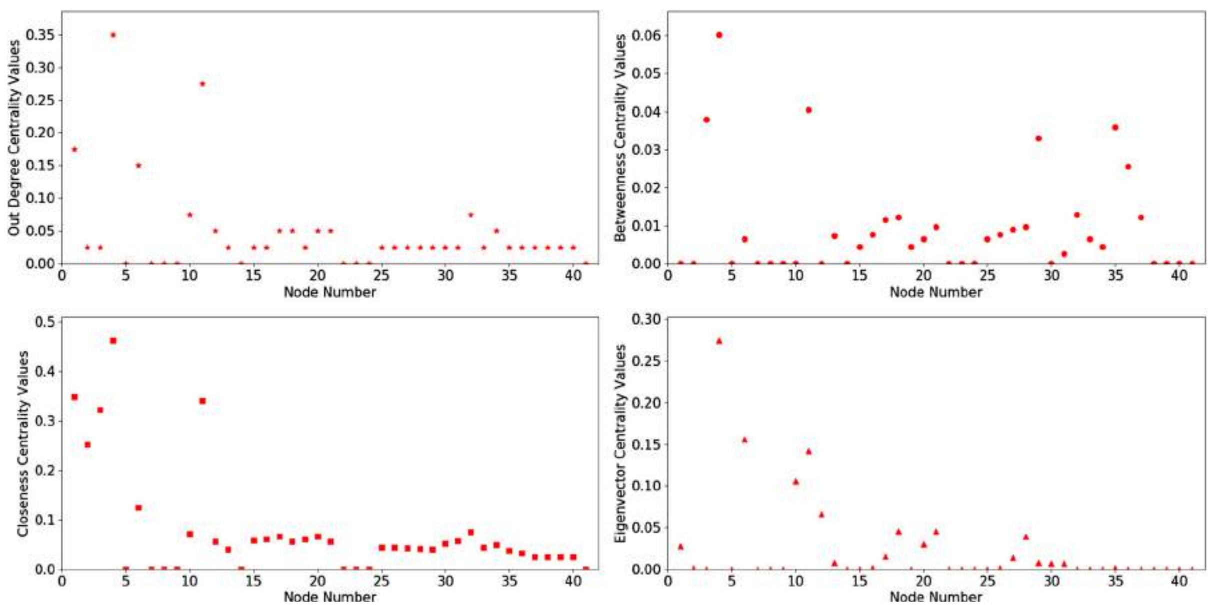


Figure 14. Centrality values for sport car seat model.

the modeling intents should not only conform to the structural requirements but also comply with functional considerations of the design. So engineers need to check whether or not the modeling intents reflected from the ADFDG comply with the structural and functional considerations of the design. They can start with the critical features. It also provides a means to review the quality of the constructed model.

Not every designer adopts a modeling methodology. Some just follow their own habits or construct models based on their experiences. That is to say, there is no standard way or consensus on how to build CAD models. Visualization and analysis of the ADFDG would reveal how a model constructed by an experienced user is different from the one constructed by a new user, which provides guidance for CAD training. It is observed that experienced CAD users build models in a way that minor features depend on major features, which could be confirmed by their resulting ADFDG. What junior CAD users can learn is how to manage the feature dependencies.

Table 4. Part of the extracted feature information for the trigger switch model.

Index	Tag	Feature type	Feature name
1	294028	DATUM_CSYS	Datum Coordinate System(0)
2	294031	DATUM_CSYS	Datum Coordinate System(1)
3	294038	SKETCH	SKETCH_000:Sketch(1)
4	294032	EXTRUDE	Extrude(2)
5	294848	DATUM_CSYS	Datum Coordinate System(3)
6	294047	SKETCH	SKETCH_001:Sketch(3)
7	294853	SWP104	Sweep(4)
8	294847	TRIM BODY	Trim Body(5)
9	294033	SHELL	Shell(6)
10	294029	BLEND	Edge Blend(7)
11	294039	DATUM_CSYS	Datum Coordinate System(8)
12	294040	SKETCH	SKETCH_002:Sketch(8)
13	294845	EXTRUDE	Extrude(9)
14	294030	EXTRUDE	Extrude(10)
15	294035	DATUM_CSYS	Datum Coordinate System(11)

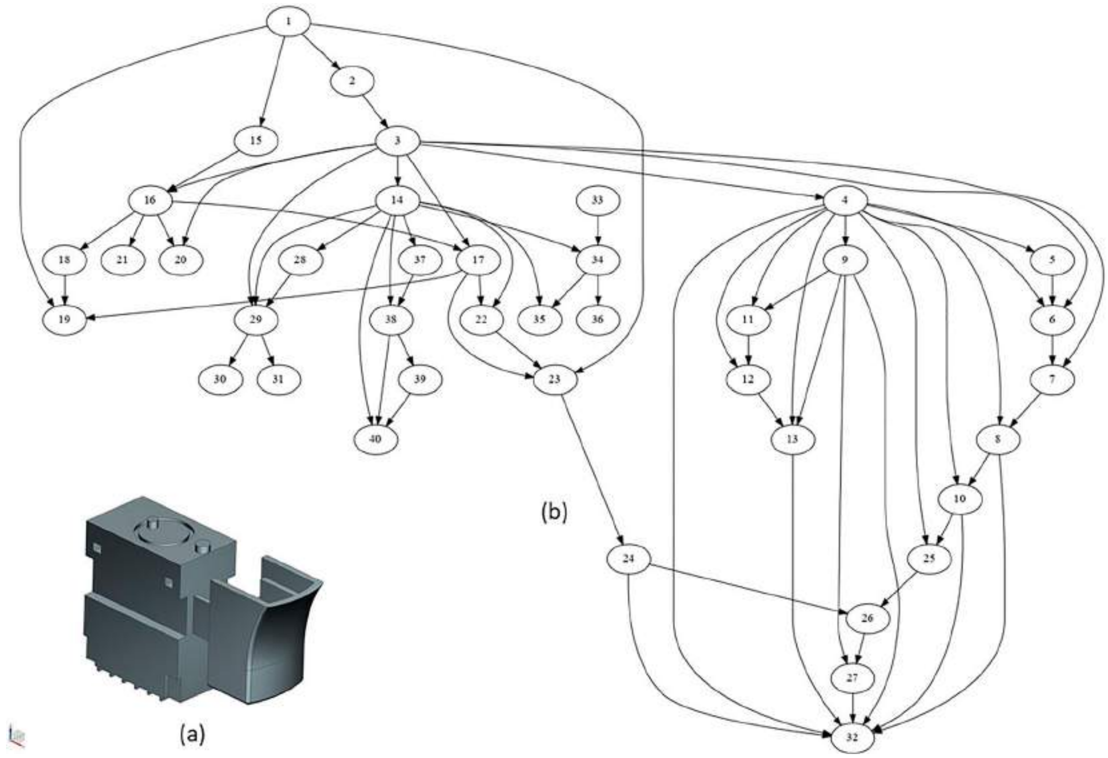


Figure 15. A trigger switch model and its ADFDG.

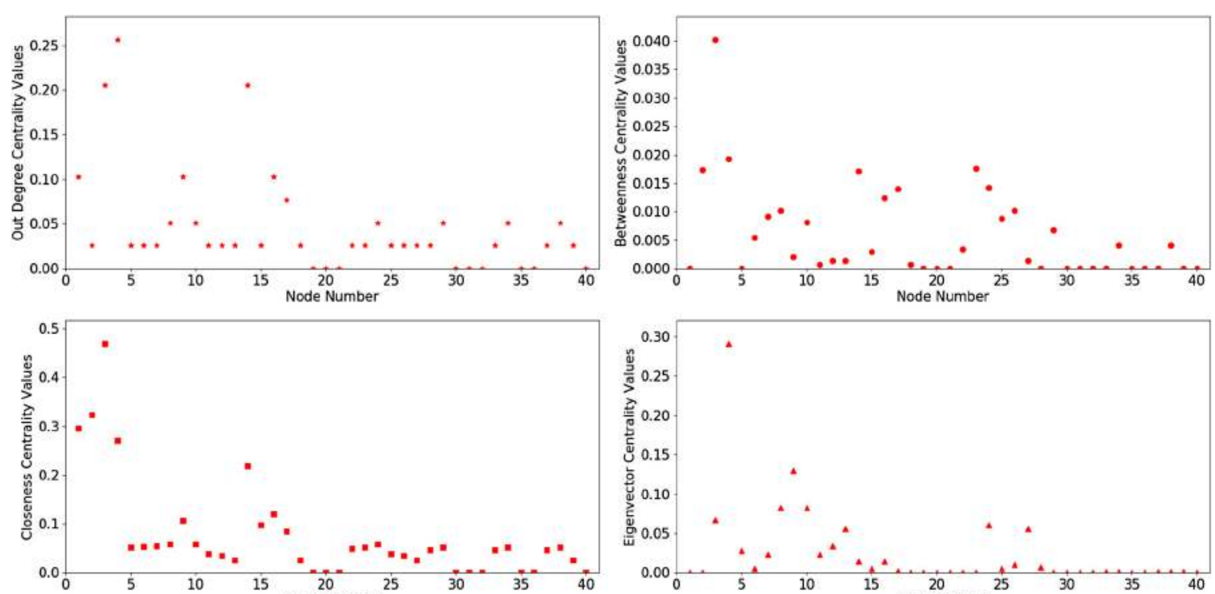


Figure 16. Centrality values for the trigger switch model.

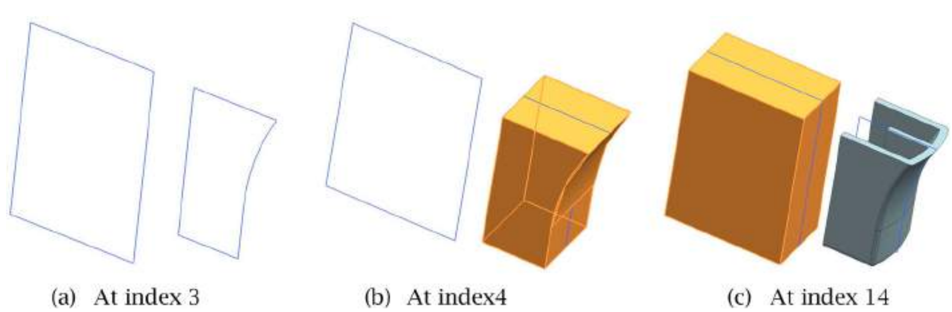


Figure 17. Some key features of the trigger switch model.

5. Conclusions

This work proposes an intelligent knowledge discovery scheme to unveil engineering modeling intents in CAD models via centrality analysis with a type of automatically-generated feature dependency graph. An algorithm has been developed to retrieve feature dependency information from CAD models, and, instead of consulting designers or engineers to build up the network for products, to generate ADFDG for both visualization and analysis purposes. Posterior examination of the modeling intents could reveal engineering constraints applied in those CAD models. Current research focuses on one important aspect of the graph properties, i.e., centrality analysis. Users with different levels of experiences tend to construct the model in different fashion, which is reflected by the resulting feature tree. In terms of the interpretation of the charts when the number of features increases, there is not much difference because there are always a few dominant features. The computational approach, given that the rules are correct, is more objective than visual observation. Potentially many exploitable engineering knowledge aspects can be revealed through this approach; that prospect warrants more future research. For example, more discoveries can be expected in the direction of merit comparison of different feature embodiment solutions.

Acknowledgements

The authors would like to acknowledge Canada MITACS and NSERC and China Scholarship Council (CSC) for their financial support. All the research works were carried out at the University of Alberta.

ORCID

Zhengrong Cheng http://orcid.org/0000-0003-2707-4491
Yanan Xie http://orcid.org/0000-0002-6615-6468
Yongsheng Ma http://orcid.org/0000-0002-6155-0167

References

[1] Bidarra, R.; Bronsvoort, W. F.: Semantic Feature Modeling, Computer-Aided Design, 32(3), 2000, 201–25. http://doi.org/10.1016/S0010-4485(99)00090-1

[2] Bonacich, P.: Power and Centrality: A Family of Measures, American Journal of Sociology, 92(5), 1987, 1170–1182. http://doi.org/10.1086/228631

[3] Bonacich, P.; Lloyd, P.: Eigenvector-like measures of centrality for asymmetric relations, Social Networks, 23 (3), 2001, 191–201. http://doi.org/10.1016/S0378-8733(01)00038-7

[4] Borgatti, S. P.; Everett, M. G.: A Graph-Theoretic Perspective on Centrality, Social Networks, 28(4), 2006, 466–84. http://doi.org/10.1016/j.socnet.2005.11.005

[5] Camba, J. D.; Contero, M.; Company, P.: Parametric CAD modeling: An analysis of strategies for design reusability, Computer-Aided Design, 74, 2016, 18–31. http://doi.org/10.1016/j.cad.2016.01.003

[6] Cheng, H.; Chu, X.: A network-based assessment approach for change impacts on complex product, Journal of Intelligent Manufacturing, 23, 2012, 1419–1431. http://doi.org/10.1007/s10845-010-0454-8

[7] Cheng, Z.-R.; Ma, Y.-S.: A network-based assessment of design dependency in product development, In Proc. 2014 Int. Conf. Innov. Des. Manuf., 2014, 137–142. Montreal, Quebec, Canada. http://doi.org/10.1109/IDAM.2014.6912684

[8] Cheng, Z.-R.; Ma, Y.-S.: A Functional Feature Modeling Method, Advanced Engineering Informatics, 33, 2017, 1–15. https://doi.org/10.1016/j.aei.2017.04.003

[9] Easley, D.; Kleinberg, J.: Networks, Crowds, and Markets: Reasoning about a Highly Connected World, Cambridge University Press, NY, USA, 2010

[10] Freeman, L. C.: A set of measures of centrality based on betweenness, Sociometry, 40(1), 1977, 35–41. http://doi.org/10.2307/3033543

[11] He, K.-J.; Chen, Z.-M.; Jiang, J.-F.; Wang, L.: Creation of User-Defined Freeform Feature from Surface Models Based on Characteristic Curves, Computers in Industry 65(4), 2014, 598–609. https://doi.org/10.1016/j.compind.2014.01.011

[12] Hoffmann, C.- M.; Robert, J.-A.: On User-Defined Features, Computer-Aided Design, 30(5), 1998, 321–32. doi:doi:10.1016/S0010-4485(97)00048-1.

[13] Jaiswal, P.; Huang, J.; Rai, R.: Assembly-based conceptual 3D modeling with unlabeled components using probabilistic factor graph, Computer Aided Design, 74, 2014, 45–54. http://doi.org/10.1016/j.cad.2015.10.002

[14] Katz, L.: A New Status Index Derived from Sociometric Index, Psychometrika, 18(1), 1953, 39-43. http://doi.org/10.1007/BF02289026

[15] Le, Q.; Sha, Z.; Panchal, J.-H: A Generative Network Model for Product Evolution, J. Comput. Inf. Sci. Eng., 14(1), 2014, 11003. http://doi.org/10.1115/1.4025856

[16] Li, L.; Lange, C.-F.; Ma, Y.-M.: Association of Design and Computational Fluid Dynamics Simulation Intent in Flow Control Product Optimization, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, available online, 2017, doi:doi:10.1177/0954405417697352.

[17] Li, M.; Zhang, Y.-F.; Fuh, J. Y. H.; Qiu, Z. M.: Toward Effective Mechanical Design Reuse: CAD Model Retrieval Based on General and Partial Shapes, Journal of Mechanical Design, 131(12), 2009, 124501. http://doi.org/10.1115/1.4000253

[18] Ma, Y.-S.; Tong, T.: Associative Feature Modeling for Concurrent Engineering Integration, Computers in Industry, 51(1), 2003, 51–71. http://doi.org/10.1016/S0166-3615(03)00025-3

[19] MacCormack, A.; Rusnak, J.; Baldwin, C.-Y.: Exploring the structure of complex software designs: An empirical study of open source and proprietary code, Management Science, 52(7), 2006, 1015–1030. Retrieved from http://www.jstor.org/stable/20110577.

[20] Sport's Car Seat, GrabCAD, https://grabcad.com/library/sports-car-seat, accessed January 2017.

[21] Siemens NX, https://www.plm.automation.siemens.com/en_us/products/nx/, accessed January 2017.
[22] Tao, S.: CAD Model Retrieval Based on Graduated Assignment Algorithm, 3D Research, 6(2), 2015, 21. http://doi.org/10.1007/s13319-015-0054-7
[23] Tao, S.; Huang, Z.; Zuo, B.; Peng, Y.; Kang, W.: Partial retrieval of CAD models based on the gradient flows in Lie group, Pattern Recognition, 45(4), 2012, 1721–1738. http://doi.org/10.1016/j.patcog.2011.09.017
[24] Wang, J.; Gu, D.; Gao, Z.; Yu, Z.; Tan, C.; Zhou, L.: 2013. Feature-Based Solid Model Reconstruction, Journal of Computing and Information Science in Engineering 13(1), 11004. http://doi:10.1115/1.4023129.
[25] Yin, C.-G.; Ma, Y.-S.: Parametric Feature Constraint Modeling and Mapping in Product Development, Advanced Engineering Informatics, 26(3), 2012, 539–552. http://doi.org/10.1016/j.aei.2012.02.010