Point-Oriented Persistent Identification of Entities for Exchanging Parametric CAD Data

Mutahar Safdar[1], Soonhung Han[2], Soonjo Kwon[3,*] and Ilhwan Song[4]

[1]Korea Advanced Institute of Science and Technology (KAIST), mutaharsafdar@kaist.ac.kr
[2]Korea Advanced Institute of Science and Technology (KAIST), shhan@kaist.ac.kr
[2]Korea Advanced Institute of Science and Technology (KAIST), soonjoy@kaist.ac.kr
[4]Hanwha Aerospace Co., Ltd., ssong82@nate.com

Corresponding author: Mutahar Safdar, mutaharsafdar@kaist.ac.kr

Abstract. TransCAD is a specialized CAD system for data exchange that is based on a set of standard modeling commands through which macro file of a CAD model is exchanged. A problem in CAD data exchange has been the consistency of persistent identifiers among different commercial CAD systems. Identifiers serve the purpose of unique and unambiguous naming for topological entities. The current identification approach experiences problems that are caused due to its topology-based mechanism. An identification approach based on a point that has both geometry and topology information is proposed. The names of topological entities i.e. face, edge and vertex are based on the names of their adjacent points. A point-based name is optimized to have most of the data needed for the mapping of arguments used in entity selection commands. This point-oriented naming method is implemented and verified in TransCAD. It is possible to resolve the current issues through a hybrid reference module. The proposed method may then be used in the development of translators for exchanging parametric CAD data.

Keywords: Parametric & Feature-based Modeling, CAD Data Exchange, Persistent Identification, Point-based Naming, TransCAD
DOI: https://doi.org/10.14733/cadaps.2020.274-287

1 INTRODUCTION

Modern manufacturing industry involves collaboration of different partners who contribute towards the completion of a project. Such collaboration requires sharing of product data and information among different companies. Product data can be exchanged in different ways and a common way is to use conventional formats like STEP or IGES [12], [13]. However, it is not possible to exchange parametric information through these standard formats.

A history-based translator, TransCAD, is being developed to exchange parametric information of CAD models [16]. It is based on a set of standard modeling commands which are composed after a survey of well-known commercial CAD systems [22]. An intelligent feature of macro-parametrics approach (MPA) is its ability to preserve the design intent which permits post-exchange modifications [19]. TransCAD being a specialized CAD system is based on macro-parametrics approach or MPA that was developed to exchange parametric information among heterogeneous CAD systems [8], [11], [18], [21]. It is possible to exchange both part and assembly models. It has different modules as shown in the Figure 1.

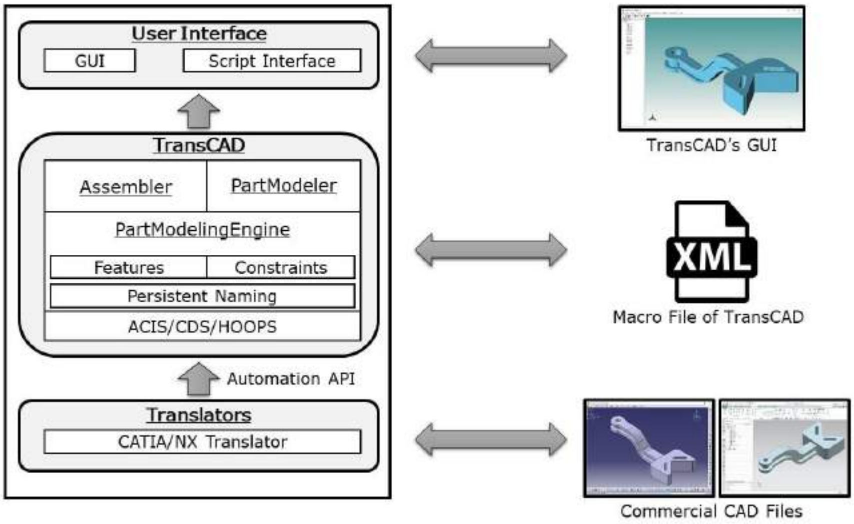

Figure 1: System architecture of TransCAD.

The exchange of CAD data involves heterogeneous CAD systems which makes it difficult to maintain data consistency during the exchange process. Persistence identification is the problem of unique naming for topological entities during the design, modification and exchange of parametric CAD models. Figure 2 covers the context of persistent identification. In commercial CAD systems, persistence identification is achieved by using a geometry or topology-based naming method. A hybrid identification approach is introduced to overcome the limitations of each method.

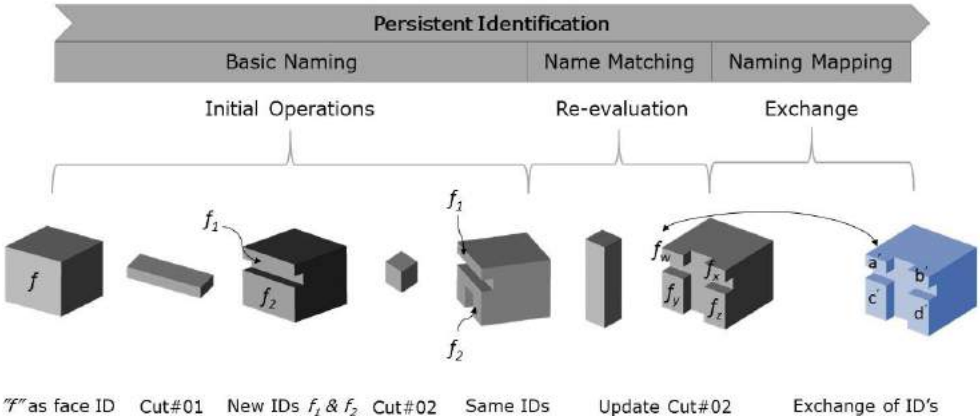

Figure 2: Context of persistent identification.

The current persistent naming module of TransCAD is based on topology naming which leads to the issues of ambiguity. To resolve these issues, additional implementation is required depending on the cause of ambiguity. An absence of such an implementation leads to the problem explained in Figure 3(a). Additionally, these topology-based identifiers lack geometric reference that may be required where the target system has a geometry-based identification system. This problem, referred to as the exchange issue of persistent naming, is explained in Figure 3(b). These problems inflate the time and effort needed to develop a translator. The aim of this work is to resolve such issues through the integration of geometry and topology into a standard reference module.

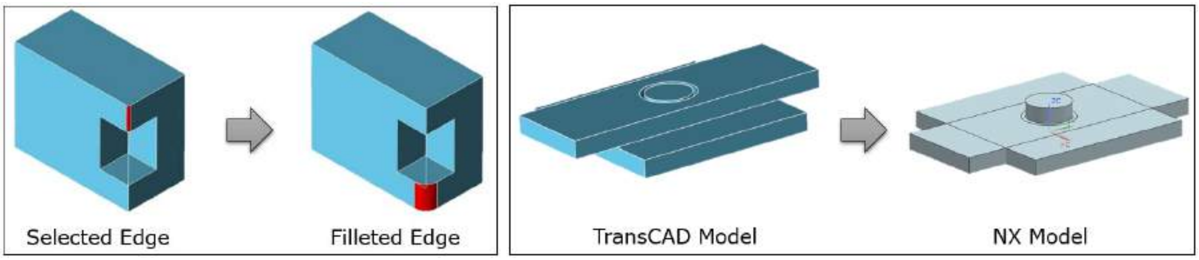

Figure 3: Problems in the current naming module of TransCAD from left to right. (a) Ambiguity Issue, and (b) Exchange Issue

2 BACKGROUND

Persistent identification in CAD models is a well-known research problem and there has been substantial research on this problem. These works can be broadly classified into two types, single CAD system (homogeneous) and CAD data exchange (heterogeneous), based on their application. In all the works of Figure 4, persistent identification was based on either topology or geometry.

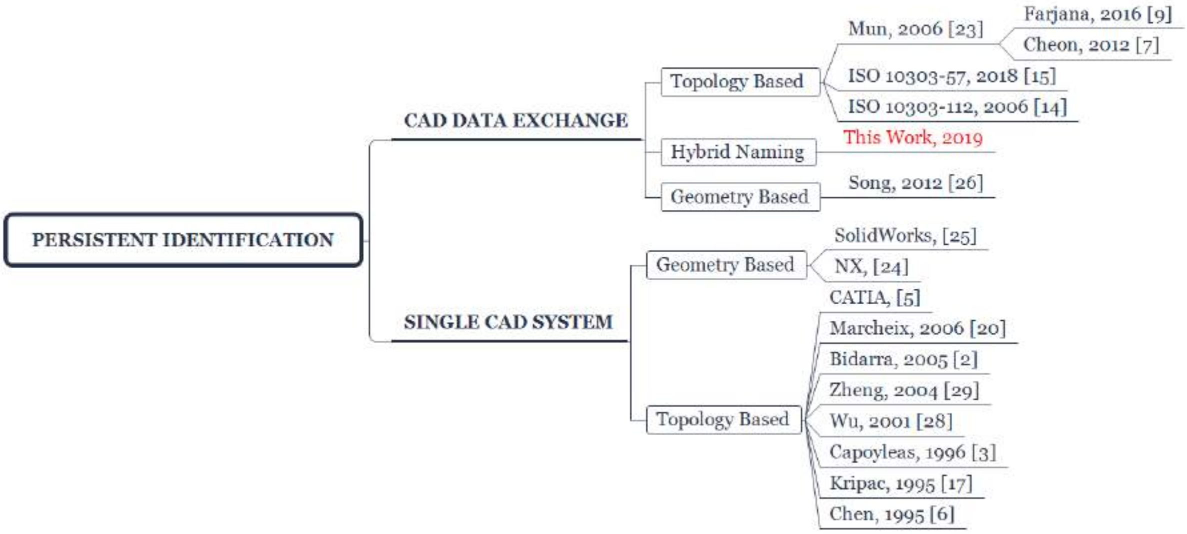

Figure 4: Different naming techniques in CAD models.

The representative works in the case of a single CAD system were those of Capoyleas and Chen who proposed a topology-based identification approach utilizing the base sketch and feature information [3], [6]. Topological context evaluation was proposed as a solution to the issue of ambiguity. In the case of complex models, such evaluation becomes time-consuming. Kripac proposed another topology-based naming technique that was based on IDs [17]. These IDs were assigned to each of the face, edge and vertex inside a CAD model. The ambiguity issue was resolved with face ID graph and intersection codes. The rules to determine intersection codes were not defined. An important contribution in this category was from Wu et al. who proposed a face-based naming mechanism [28]. They defined parametric space information to overcome the issue of ambiguity. Parametric space information was dependent on coordinate system and was altered by the change of parametric space.

Entity ID tables were updated every time a change was made in the model. A persistent naming approach of shells was also proposed by Marcheix [20]. Recently, more contemporary approaches have been proposed to resolve the issue of persistent identification with a special focus on re-evaluation [4].

Similarly, there has been a number of researches on naming methods proposed for the history-based translator. Mun and Han proposed a face-based identification approach similar to Wu with a focus on parametric CAD models [23]. They used geometry-based numbering to differentiate among ambiguous entities. This was later implemented in TransCAD. Farjana et al. expanded this face-based naming to pattern feature by using a similar geometry comparison [9]. However, this method doesn't combine geometry and topology information inside persistent names and hence the hybrid information cannot be used during the exchange process. A geometry-based neutral macro file was introduced by Song and Han to overcome the issue of ambiguity [26]. Lately, ISO 10303-57 was published and procedural identification of entities was based on topology naming [15]. In macro-parametrics approach (MPA), geometry was never integrated for use during the exchange process. This research combines geometric reference with topologic reference to overcome the existing issues of above methods. A comparison of these MPA-based identification approaches is covered in Table 1. The letter P, V, E, and F represent the entities of point, vertex, edge, and face respectively.

DETAILS	MUN [2006]	CHEON [2012]	SONG [2012]	FARJANA [2016]	ISO 10303-57 [2018]	This Research [2019]
Basic Naming	o	×	o	×	o	o
Ambiguity Solving	Δ	×	o	o	o	o
Name Matching	o	o	o	×	o	o
Naming Mapping	o	×	o	×	×	o
Information Type	Topology	Topology	Geometry	Geometry	Topology	Hybrid
Naming Hierarchy	F-E & F-V	F-E & F-V	V-E & V-F	×	F-E & F-V	P-V-E & P-V-F
O = Covered **Δ** = Incomplete **×** = Out of Scope/ Not Covered						

Table 1: MPA-based naming techniques.

3 POINT-ORIENTED NAMING

A naming method is an approach through which ID's are assigned to each and every entity of a CAD model. The purpose of these ID's is to provide unique and unambiguous identification for future references. As a model is updated inside a CAD system so are its ID's. It is clear that only a well sorted method can be a base for a robust algorithm. Most of the CAD vendors have their own identification approaches and cannot be used directly because of the copyright issues. In order to develop a history-based translator with a neutral format as an intermediate scheme, we need to define a separate naming method. In the current version of TransCAD, a topology-based naming method is implemented that has its limitations as described in previous section. A hybrid naming method is therefore defined and implemented for TransCAD.

In order to come up with a robust naming mechanism that has a hybrid structure representing both topology and geometry information, a point-based naming mechanism is explained in this section. It is based on the concept of defining a point inside a CAD model by using both geometry and topology information. The method can be described as a hybrid because it uses both topology and geometry to overcome the limitation of each method as shown in Figure 5. A point is the most basic entity in a CAD model. A vertex in topology is analogous to a point in geometry. This provides the basis to develop a point-based naming mechanism. The remaining entities namely edge and face are defined in terms of already defined vertices.

As the method is based on a point definition, it is necessary to survey the different types of user operations through which a point can be inserted. Once all these operations are defined, a general syntax for naming a point in CAD models can be finalized. These named points can then be used to name topological entities i.e. vertices, edges and faces.

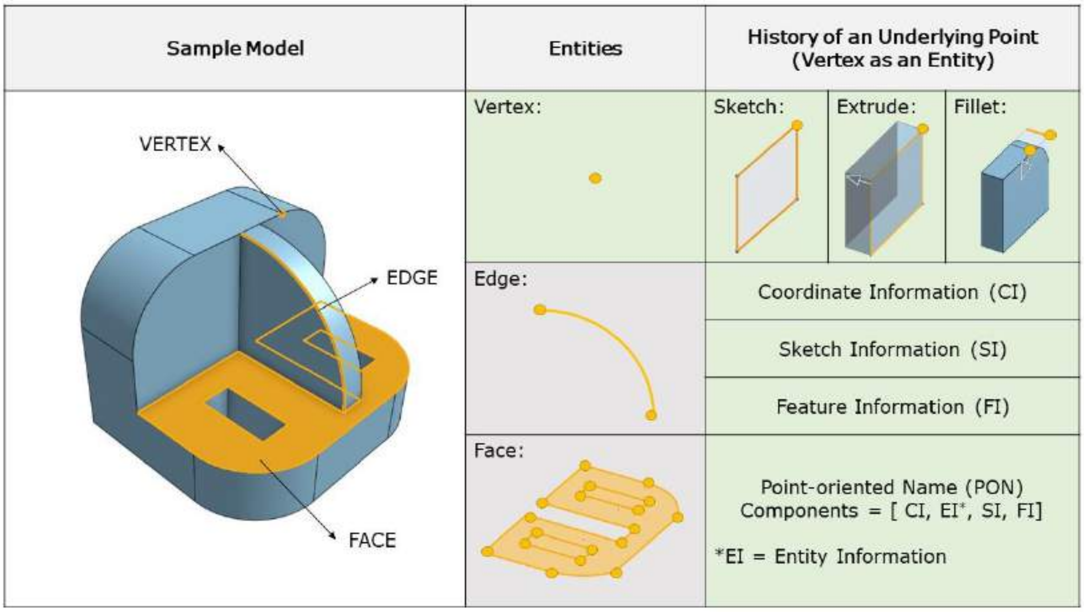

Figure 5: Definition of point-oriented naming.

3.1 Information Components of Point-Oriented Name

A CAD model consists of different points depending on their creation method. Modern CAD systems provide a variety of features and operations. A point can have the following information components depending on how it was created:

3.1.1 Coordinate information (CI)

Coordinates like anywhere else are used to define a specific position in a CAD model. This represents the location (local/global) of a point in design space of a CAD system. Since each location is unique, the coordinates provide the much-needed geometric information. In point-oriented names, coordinate information is used to resolve the issues related with ambiguity.

3.1.2 Entity information (EI)

Entity information refers to the basic entity types present inside a CAD system. A model can be decomposed into three basic entity types namely face, edge, and vertex. In some commercial CAD systems, entity information is at the start of an ID. It has the advantage of providing instant information about the type of entity that is to be referenced.

3.1.3 Sketch information (SI)

Some of the features in CAD systems are based on sketch definitions. These definitions can be exploited to uniquely name an entity in a CAD system. This includes the sketch name and its elements in the case a point is generated through a sketch-based feature. The sketch information can also be used during the exchange process.

3.1.4 Feature information (FI)

This represents the features with which a point is associated. A point can have more than one feature associations. In this category, we have generating feature information (GFI), transition feature information (TFI), and reproducing feature information (RFI) depending on which type of feature was used to create a point.

3.2 Naming a Point

Most of the commercial CAD systems are feature-based parametric modeling systems. A feature-based point-oriented syntax is introduced. A point in a CAD model can be created through the following methods: feature creation, feature transition, or feature imitation.

3.2.1 Sketch-based or generating features

Points can be generated inside a CAD model by inserting new features. A feature can be inserted by referencing a base sketch. In this case, the name will have feature information, sketch information, and coordinate information. Such features include Extrude, Sweep, and Revolve etc.

The point-oriented name (PON) will have the following information components:

PON = [GFI, SI, CI]

3.2.2 Entity-based or transition features

Transition features are an example of entity-based features. The representative types include fillet and chamfer features. A topological entity is referenced for the creation of these features. In this case, the name will have feature information and the coordinates of newly generated points.

The PON will have the following information components:

PON = [TFI, CI]

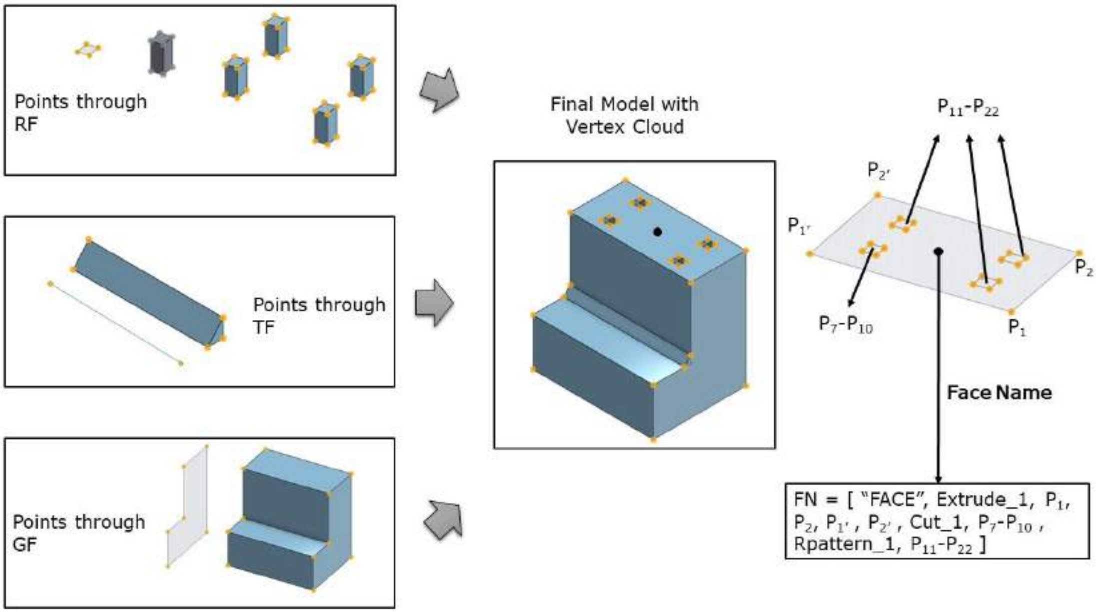

Figure 6: Vertex cloud of different features.

3.2.3 Feature-based or reproducing features

A feature-based parametric modeling system provides the functionality to copy or mirror the existing features in user specified patterns. These patterns take the selected features of a CAD model and reproduce them inside a part or an assembly model. As a result, new points are generated inside the CAD model. In this case, the name will have feature information and coordinates of newly generated points.

The PON will have the following information components:

PON = [RFI, CI]

A combination of all these point types as a vertex cloud in the final model is shown in Figure 6.

3.3 Naming a Vertex

Vertex is one of the entities which is named using the previously defined point names. As explained earlier, a topologic vertex is semantically similar to a geometric point. This allows the use of a point name in place of a vertex with the addition of entity information. The entity information (EI) in the case of a vertex is a capitalized "VERTEX" at the start of its name.

The point-oriented vertex name (POVN) will have the following information components, where the information in PON will depend on the type of underlying point.

POVN = [VERTEX, Underlying PON]

3.4 Naming an Edge

Edge is the second entity to which a point-based ID is attached. A topologic edge is semantically similar to a geometric curve. It is defined by its bounding vertices. Principally, an edge name should consist of two surrounding vertex names. The entity information (EI) in the case of an edge is a capitalized "EDGE" at the start of its name.

The point-oriented edge name (POEN) will have the following information components, where the information in POVN will depend on the type of underlying point.

POEN = [EDGE, Start POVN, End POVN]

3.5 Naming a Face

Face is the third and final topological entity to be named. A topologic face is semantically similar to a geometric surface. It is defined by its bounding vertices. The entity information (EI) in the case of a face is a capitalized "FACE" at the start of its name. A face can have three or more boundary points depending on its configuration.

The point-oriented face name (POFN) will have the following information components, where the information in POVN will depend on the type of underlying point.

POFN = [FACE, Boundary POVNs]

3.6 General Names

Summarizing the above point creation approaches and the information that is related to points in a 3D model, a general point and the corresponding topological entities can be named as follows:

Point Name (PN) = [NFI, FI]

Vertex Name (VN) = [EI, PN]

Edge Name (EN) = [EI, {VN}]

Face Name (FN) = [EI, {VN}]

The NFI or non-feature information varies from feature to feature and includes everything except feature itself. The curly brackets in EN & FN indicate the possibility of having more than one VN. The reason to separate a point name from a vertex name is to adopt a more conventional outlook for naming topological entities i.e. naming of vertices, edges, and faces.

4 IMPLEMENTATION

The point-based naming mechanism is implemented for the history-based macro-parametrics translator TransCAD. The target CAD systems at this point are CATIA and NX. It will be used for exchanging parametric CAD models between CATIA and NX [5], [24]. This process is usually referred to as the naming mapping and is logically the next step of this ongoing development.

The implementation specifications are summarized in Table 2.

COMPONENT	TYPE
OS	Windows 10
Language	C++
Geometric Modeling Kernel	ACIS R25
Visualization Tool	HOOPS 1919
IDE	Visual Studio 2010
MPA Translator	TransCAD 6.2

Table 2: Implementation specifications.

The first step of the implementation includes adding the method's algorithm in existing source code of MPA translator. Like most of the existing CAD systems, the implementation is performed via explicit feature definitions. Based on each feature definition, information components required to write the name of a topological entity are derived from each feature. The functions to get such information are in accordance with the geometric modeling kernel and are usually provided by the respective vendors. The geometric modeling kernel used in TransCAD is ACIS from Spatial Corporation which is a part of Dassault Systèmes [1]. This is important as it greatly influence the overall structure of the implementation algorithm. The visualization in TransCAD is based on HOOPS toolkit [10]. The implementation of MPA approach as a whole involves a number of complexities and is not a cornerstone in this research. However, the source code can be accessed at GitHub [27].

The completion of previous process provides an implementation of the proposed naming method inside the translator. The next step of implementation revolves around the utility of this method in exchanging parametric information. The implementation in this step is usually dependent on the target CAD systems and target test models. The result of initial completion, which is a string of attributes, is used in this step. The file is parsed to read the name of each referenced entity.

The reference or persistent naming module is updated as in Figure 7.

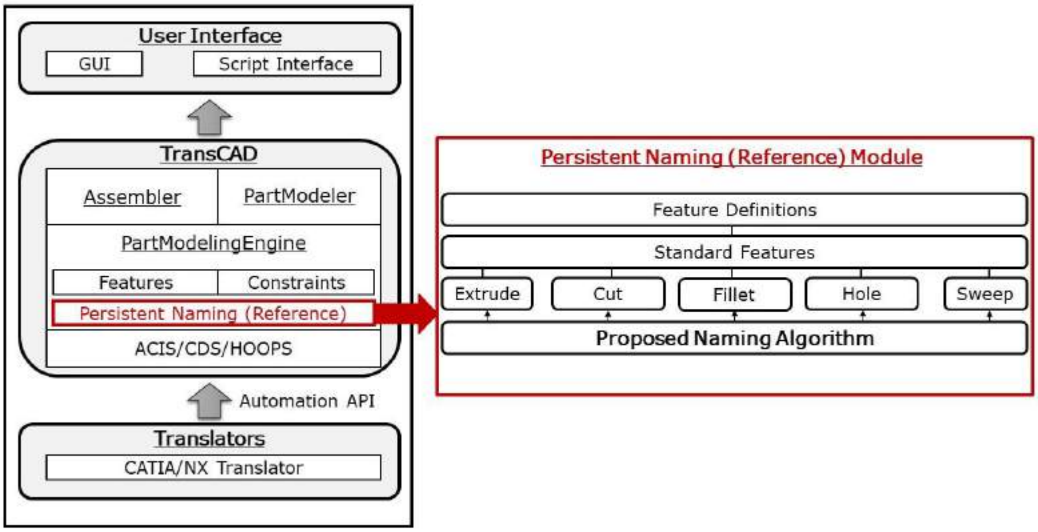

Figure 7: Implementation of proposed reference module in TransCAD

5 DATA STRUCTURE AND SOLUTION ALGORITHM

The MPA approach has a feature-based data structure and the solution algorithm of naming method is implemented through individual feature classes. The data structure for the point-based naming method is shown in Figure 8. The algorithm can be same for all the features with minor variations depending on feature types.

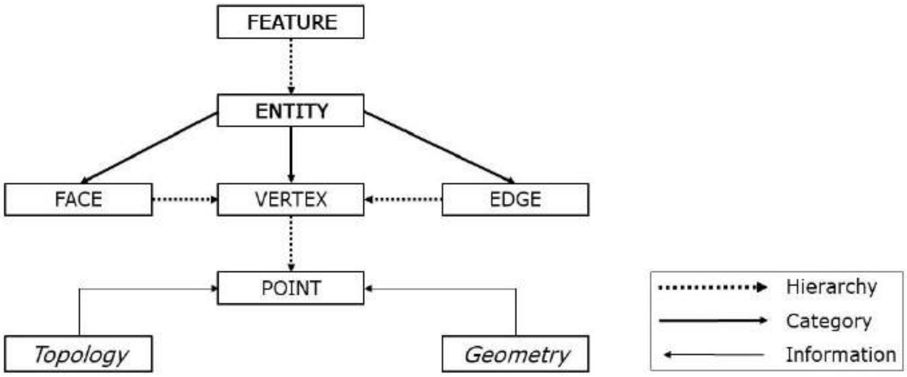

Figure 8: Data structure of point-based naming method.

The current algorithm works by extracting the points associated with each feature. The feature information along with the point's coordinate is then attached to the underlying vertex. These IDs are used in the naming of faces, edges and vertices inside a CAD model. In TransCAD interface, users can select the names of these entities. The point names are attached to all the vertices separately.

A pseudo code for the point-based naming method is as follows:

Part-I: ***Naming points in final model***

```
For each POINT
        Get associated feature, coordinate & sketch information
        Arrange the information in a name
        Attach the point-based name
End for POINT
```

Part-II: ***Naming topological entities***

```
If (entity == FACE)
        For each entity FACE
                Attach entity information as FACE
                Traverse COEDGE loop of FACE
                Get associated EDGE list
                        For each entity EDGE
                                Get associated VERTEX list
                                        For Each VERTEX
                                                Get underlying POINT name
                                        End for VERTEX
                        End for EDGE
        End for FACE
Else If (entity == EDGE)
        For each entity EDGE
                Attach entity information as EDGE
                Get associated VERTEX list
                        For Each VERTEX
                                Get underlying POINT name
                        End for VERTEX
        End for EDGE
Else
```

```
Attach entity information as VERTEX
          For Selected VERTEX
                    Get underlying POINT name
          End for VERTEX
```

6 RESULTS

In this section, results of point-based persistent identification in history-based macro-parametrics translator are presented. These persistent ID's of TransCAD are amplified to improve readability. The current results of implementation include basic naming and name matching. The basic naming is shown for sketch-based and entity-based features. While name matching in the creation stage of CAD models is depicted through different test models.

6.1 Point-Oriented Naming in TransCAD

The basic naming is done first as shown through the GUI output of Sec. 6.1. The persistent ID's are magnified in order to show the components of point-based name. These ID's include entity, feature, sketch, and coordinate information. The basic naming of different features in TransCAD is shown in this section. Figure 9, Figure 10, Figure 11, and Figure 12 include naming for Extrude, Cut-Extrude, Fillet, and Chamfer feature respectively.

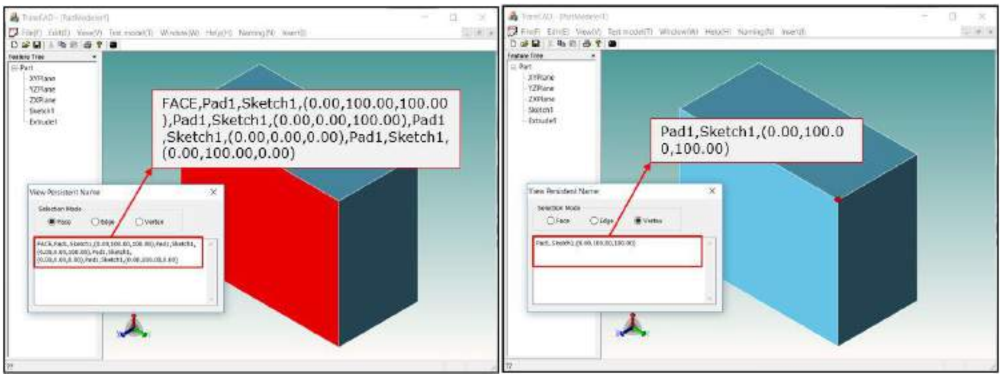

Figure 9: Extrude feature with point-based naming from left to right. Entities: (a) FACE name, and (b) VERTEX name.

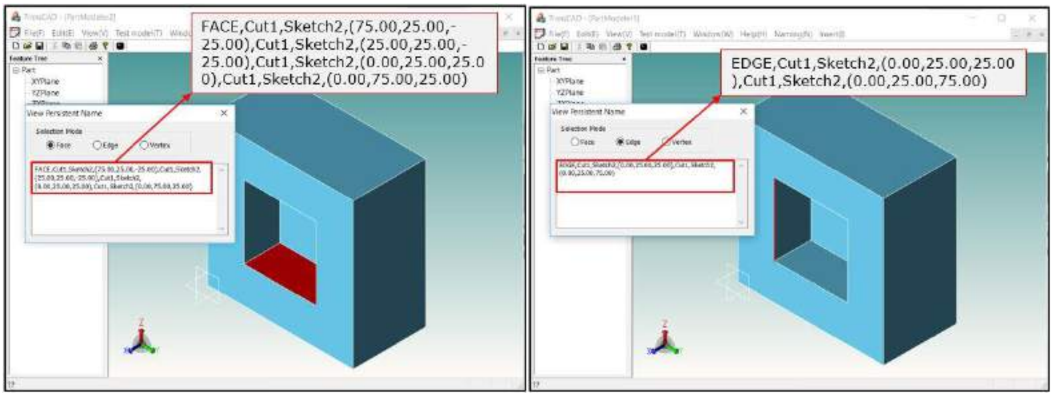

Figure 10: Cut-Extrude feature with point-based naming from left to right. Entities: (a) FACE name, and (b) EDGE name.

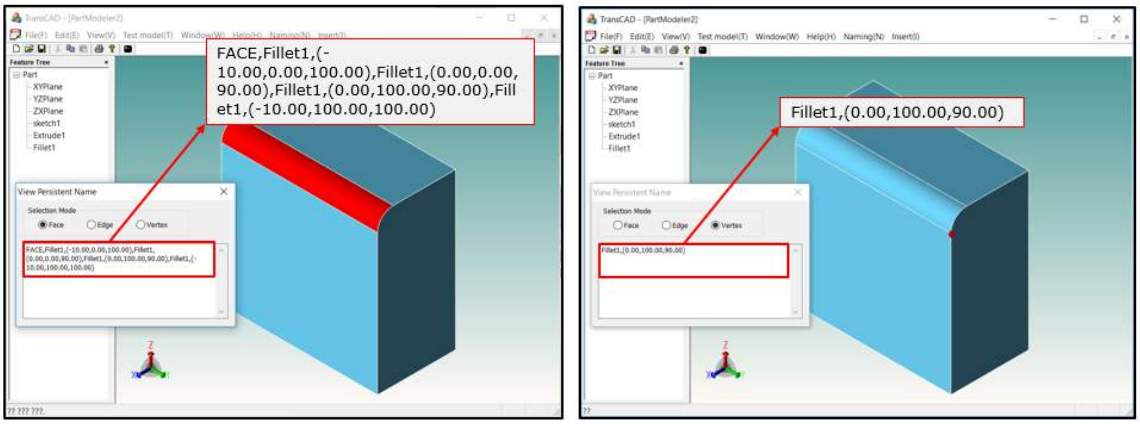

Figure 11: Fillet feature with point-based naming from left to right. Entities: (a) FACE name, and (b) VERTEX name.

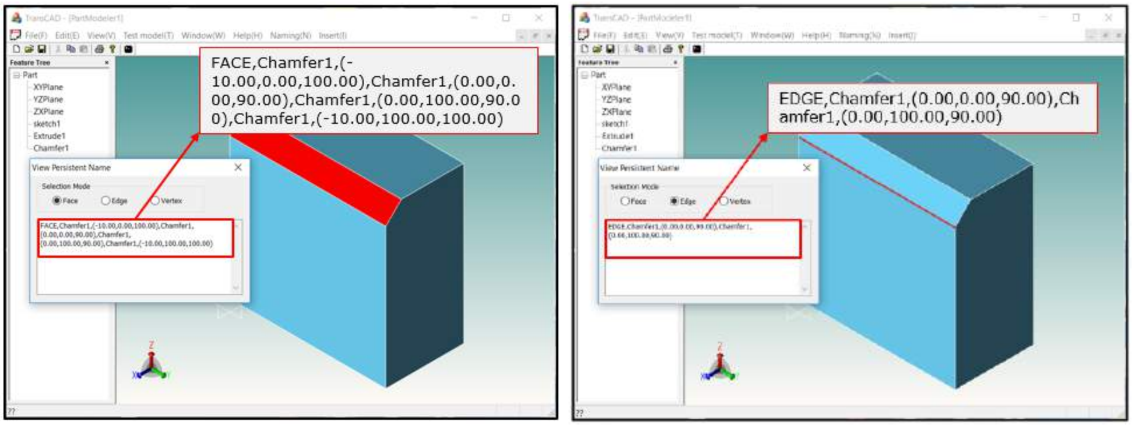

Figure 12: Chamfer feature with point-based naming from left to right. Entities: (a) FACE name, and (b) EDGE name.

6.2 Point-Oriented Test Models

In the second category, test models are constructed based on point-oriented persistent names as in Figure 13. These models were selected as they result in different errors while testing newly implemented naming methods. These feature interactions require name matching to include complete and updated point names (those that are achieved after two or more feature interactions).

In the first model, three features namely Extrude, Cut-Extrude and Fillet are used, and are based on the names shown in previous section. In these models, face names are highlighted to show the combination of point names in the case of multiple features. P-Models are critical in judging a naming method because they present issues as in P1 model. The resolution of these issues both inside TransCAD and during exchange with commercial CAD systems has been the primary motive of this research. It is expected that both type of errors can be resolved by using a hybrid reference module.

Following models of Figure 13 were used to check the point-based names in the case of multiple feature interactions:

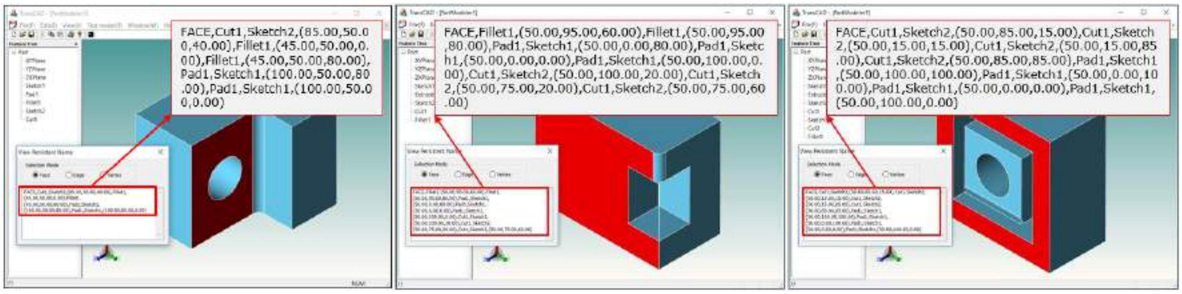

Figure 13: Point-based feature interaction from left to right. Models: (a) K1, (b) P1, and (c) P2.

6.3 Split Edge Issue Resolution

One of the issues discussed at the start was ambiguity arising from the absence of geometric reference in CAD models. It is tested in P1 model that geometric information can help resolve this issue as each endpoint in EDGE entity has different coordinates. Ambiguity resolution, with point-based names, is shown in Figure 14. As a result, the fillet feature is applied correctly.

It can be seen from these ID's that each EDGE is traversed in same direction. This leads to different arrangement of feature information in each ID. According to ACIS topology, these edges are part of two different COEDGES (each of the adjacent faces). A LOOP of COEDGES is always traversed in counter-clockwise direction, which clearly means that these EDGES are part of the COEDGE LOOP that belongs to bigger/larger face.

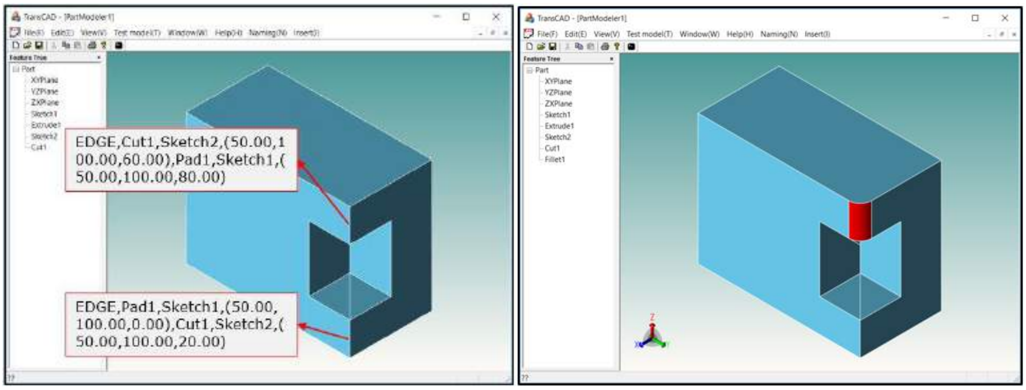

Figure 14: Ambiguity resolution in TransCAD from left to right, (a) Unique names, and (b) Fillet on distinct EDGE.

7 CONCLUSIONS AND FUTURE WORK

It has been a long-standing desire to resolve the problem of persistent identification in CAD models. The same problem is responsible for inaccurate results in the macro-parametrics translator TransCAD which is a specialized CAD system for exchange. It can be seen from the case of split edge in Figure 14 that geometry can distinguish among topologically similar entities and hence resolve this problem. From the viewpoint of CAD data exchange, the newly attached geometric reference can be a substitute of missing geometry information if the target system is based on geometry. With this method, it is possible to see much of the associated modelling information as in Figure 13(b) where the selected face has the information of three different features. In the case of a pure topology or geometry-based naming method, only single feature information is available. Although the method assigns an information rich string to each entity inside the CAD model, it becomes difficult to manage this information in the case of a face having large number of boundary points. In addition, some models have round edges where underlying boundary points need to be searched in order to attach point-based names.

In future, these newly attached persistent names are to be used during the exchange involving CAD data. This is illustrated by naming mapping in Figure 2. It is also possible to attach additional attributes (including sketch element information) in the existing point-based names. The naming mapping will be done by further implementing a geometry based neutral macro file. This will allow to read/write the hybrid information from/to the file of each CAD model. Such an implementation can be a separate topic of research and will be the next step of this ongoing development.

8 ACKNOWLEDGEMENTS

This research is supported by the Plant Research Program [Project ID: 18IFIP-B091004-05] funded by the Ministry of Land, Infrastructure and Transport, and by the Industrial Core Technology Development Program [Project ID: 10063452] funded by the Ministry of Trade, Industry and Energy of the Korean Government. The authors gratefully acknowledge their support.

Mutahar Safdar, https://orcid.org/0000-0003-0961-3404
Soonhung Han, https://orcid.org/0000-0001-5676-8121
Soonjo Kwon, https://orcid.org/0000-0003-1578-9262

REFERENCES

[1] ACIS, https://www.spatial.com
[2] Bidarra, R.; Nyirenda, P.J.; Bronsvoort, W.F.: A feature-based solution to the persistent naming problem, Computer-Aided Design & Applications, 2(1-4), 2005, 517-526. https://doi.org/10.1080/16864360.2005.10738401
[3] Capoyleas, V.; Chen, X.; Hoffmann, C.: Generic naming in generative, constraint-based design, Computer-Aided Design, 28(1), 1996, 17-26.
[4] Cardot, A.; Marcheix, D.; Skapin, X.; Arnould, A.; Belhaouari, H.: Persistent naming based on graph transformation rules to reevaluate parametric specification, Computer-Aided Design & Applications, 16(5), 2019, 985-1002. https://doi.org/10.14733/cadaps.2019.985-1002
[5] CATIA, https://www.3ds.com/products-services/catia/
[6] Chen, X.; Hoffmann, C.: On editability of feature-based design, Computer-Aided Design, 27(12), 1995, 905-914.
[7] Cheon, S.U.; Mun, D.; Han, S.; Kim, B.C.: Name matching method using topology merging and splitting history for exchange of feature-based CAD models, Journal of Mechanical Science and Technology, 26(10), 2012, 3201-3212. https://doi.org/10.1007/s12206-012-0827-3
[8] Choi, G.; Mun, D.; Han, S.: Exchange of CAD part models based on the macro-parametric approach, International Journal of CAD/CAM, 2(1), 2002, 13-21.
[9] Farjana, S.H.; Mun, D.; Han, S.: Implementation of persistent identification of topological entities based on macro-parametrics approach, Journal of Computational Design and Engineering, 3(2), 2016, 161-177. https://doi.org/10.1016/j.jcde.2016.01.001
[10] HOOPS, https://www.techsoft3d.com/products/hoops/visualize
[11] iCAD Laboratory, http://icad.kaist.ac.kr
[12] IGES, https://en.wikipedia.org/wiki/IGES
[13] ISO 10303, https://en.wikipedia.org/wiki/ISO_10303
[14] ISO TC184/SC4, ISO 10303 - Industrial automation systems and integration -- Product data representation and exchange -- Part 112: Integrated application resource: Modelling commands for the exchange of procedurally represented 2D CAD models, 2006.
[15] ISO TC184/SC4, ISO 10303 - Industrial automation systems and integration -- Product data representation and exchange -- Part 57: Integrated generic resource: Persistent identification of elements in procedural shape modeling, 2018.
[16] Kim, B.C.; Han, S.: Integration of history-based parametric translators using the automation APIs, International Journal of Product Lifecycle Management, 2(1), 2007, 18-29. https://doi.org/10.1504/IJPLM.2007.012872
[17] Kripac, J.: A mechanism for persistently naming topological entities in history based parametric solid models. Proceedings of the 3rd ACM symposium on Solid Modeling and Applications, 1995, 21-30. http://doi.org/10.1145/218013.218024

[18] Lee, H.; Jauhar, T.A.; Kim, I.; Kwon, S.; Han, S.: A web-based solution for collaborative design supporting multiple CAD systems, Proc. 23rd International ACM Conference on 3D Web Technology, Poznan, POLAND, 2018. https://doi.org/10.1145/3208806.3208822
[19] Macro-Parametrics Approach (MPA) Homepage, http://www.macro-parametrics.org/
[20] Marcheix, D.: A persistent naming of shells, International Journal of CAD/CAM, 6(1), 2006, 125-137.
[21] MPA Digital Handbook (Video), https://www.youtube.com/watch?v=3sfGt8JBmjU
[22] Mun, D.; Han, S.; Kim, J.; Oh, Y.: A set of standard modeling commands for the history-based parametric approach, Computer-Aided Design, 35(13), 2003, 1171-1179. https://doi.org/10.1016/S0010-4485(03)00022-8
[23] Mun, D.; Han, S.: Identification of topological entities and naming mapping for parametric CAD model exchanges, International Journal of CAD/CAM, 5(1), 2005, 69-81.
[24] NX, https://www.plm.automation.siemens.com/global/en/products/nx/
[25] SolidWorks, https://www.solidworks.com/
[26] Song, I.; Han, S.: Parametric CAD data exchange using geometry-based neutral macro file, Proc. 7th International Conference on Cooperative Design, Visualization and Engineering, Calvia, Mallorca, SPAIN, 2010, 145-152. https://doi.org/10.1007/978-3-642-16066-0_23
[27] TransCAD source code on GitHub: https://github.com/Macro-parametrics/TransCAD
[28] Wu, J.; Zhang, T.; Zhang, X.; Zhou, J.: A face based mechanism for naming, recording and retrieving topological entities, Computer-Aided Design, 33(10), 2001, 687-698. https://doi.org/10.1016/S0010-4485(00)00099-3
[29] Zheng, J.; Fan, M.; Tong, R.; Dong, J.: A mechanism for persistently naming topological entities, Proc. 8th International Conference on Computer Supported Cooperative Work in Design, Xiamen, CHINA, 2004, 84-89. https://doi.org/10.1109/CACWD.2004.1349162