A Method for verification of CAD model errors

Jeongsam Yang
Ajou University

Soonhung Han
Korea Advanced Institute of Science and Technology

Sangho Park
Mokwon University

A Method for Verification of CAD Model Errors

Jeongsam Yang[1*], Soonhung Han[1], Sangho Park[2]

jsyang@icad.kaist.ac.kr, shhan@kaist.ac.kr, spark@cnu.ac.kr

[1] Department of Mechanical Engineering

Korea Advanced Institute of Science & Technology

373-1, Gusong-dong, Yousong-gu,

Daejon 305-701, Korea

Tel: +82-42-862-9225

Fax: +82-42-862-9224

[2] Department of Mechanical Design Engineering

Chungnam National University

220, Gung-dong, Yuseong-gu,

Daejeon 305-764, Korea

Tel: +82-42-821-6643

*Corresponding author. Email: jsyang@icad.kaist.ac.kr

Abstract

Neutral CAD formats such as STEP(Standard for the Exchange of Product Model), IGES(Initial Graphics Exchange Specification), DXF(Drawing Exchange format), and VDAF(Verband der Automobilindustrie Format) are common methods of data sharing among heterogeneous CAD systems. CAD models have errors, which originate from data loss during the data conversion process or the technical weakness of the neutral format. Error recovery requires an increase in lead-time and additional expenses for product development. The serious problem is that errors in a CAD model are not found until after downstream development processes.

In this paper, we propose a method to identify CAD model errors which are contained in a neutral format. Considering different topological and geometrical characteristics in the STEP and IGES formats, our method includes a verification process to check a CAD model error step by step without wasting resources. To test the idea, we have implemented a verification system that checks topological and geometrical errors of 19 items in the STEP format and 12 items in the IGES format.

Keywords

Data exchange; Geometry; IGES; Model error; STEP; Topology; Verification

1. Introduction

During a product development cycle, various CAD systems are being used. They can be classified into a vertical relationship and a horizontal relationship according to the data stream among them. Hyundai Motors uses Pro/Designer and ALIAS for style design, CATIA for designing engineering parts and assemblies, and Pro/Engineer for designing power trains. In a horizontal relationship, the design data should be shared among upperstream applications of CATIA, Pro/Engineer, and ALIAS during the collaborative design stage among various disciplines inside the company. In a vertical relationship, model data from an upperstream application is converted into downstream applications such as finite element analysis, rapid prototyping and numerical control manufacturing.

CAD model data is being exchanged between systems, using a neutral format such as the standard for the exchange of product model data (STEP) or initial graphics exchange specification (IGES). However, both the data loss during the conversion process and the structural weakness of the neutral format cause errors in the CAD models. Because a designer cannot easily find CAD model errors like small gaps, designers waste 20% ~ 50% of their time repairing or recreating CAD models (Tassey, 1999).

We can trace a CAD model error from the designer side. If a designer generates a model based on the wrong method without observing the rules required by the CAD system, bugs or flaws are contained in the CAD model. The error propagates or diffuses into downstream CAD systems during data exchanges. Those errors are frequently found in the final stage such as mold manufacturing or finite element analysis. To solve this problem, designers occasionally depend on their intuition.

There have been researched on the verification of CAD models in restricted areas. Hoffman *et al*. (1998) approached the model errors in the direction of geometric dimensioning and tolerancing, and Gu *et al*. (2001) studied various errors that exist in a CAD model and their theoretical fundamentals. Deshpande *et al*. (2000) used a method called *complementary model object tree* to find and fix errors. Barequet *et al*. (1996, 1997), Steinbrenner *et al*. (2001) and Volpin *et al*. (1998) approximated the exact shape as a polyhedron form to find gaps and overlaps between polygons. Some of previous studies were undertaken as a branch of computational fluid dynamics, computational structural analysis, and mesh generation, where the CAD model errors are examined for specific purposes such as gaps or overlaps.

In this paper, we classify the CAD model errors from the viewpoint of topology and geometry, and we propose a procedural method to find errors. The procedural method avoids duplicating the evaluation process and executes a systematic error verification based on the quantitative input of the designer. The CAD model formats that we have used are STEP AP214 (1997), which is for automotive mechanical design processes, and IGES v5.3 (1990). Using the implemented *CAD model error verification system* we can verify 19 errors in the STEP format and 12 errors in the IGES format. We have tested this system on automotive parts with the courtesy of Hyundai Motors.

2. Procedural Approach of Error Verification

2.1 Error Classification

The error items to be verified are based on the JAMA(Japan Automobile Manufacturers Association) PDQ(Product Data Quality) Verification Model Data Specifications V2.0 (JAMA, 2002). Among the 45 error items of the JAMA/JAPIA PDQ, the 19 error items that the Hyundai Motors' designers put high priority are handled.

The international standard STEP AP214 is an application protocol for the automotive industry. It contains topological and geometrical information used for constructing a base geometry. From a STEP physical file, we can check the following 19 topological and geometrical errors, among which are illustrated in Figures 1 and 2:

1) Edge-Vertex Gap

2) Edge Length

3) Edge Curvature

4) Edge-Face Gap

5) Edge-Loop Consistency

6) Edge-Edge Gap/Overlap

7) Edge-Edge Angle

8) Edge-Edge Proximity

9) Face Degree

10) Face-Face Gap/Overlap

11) Face-Face Angle

12) Face-Face Consistency

13) Edge-Use-Count

14) Void-Face/Surface

15) Curve Length

16) Curve Curvature

17) Surface Degree

18) Surface-Surface Gap/Overlap

19) Surface-Surface Angle

The first fourteen items check the solid data and surface data with topological information, and the last five items check the surface without topological information.

Most IGES models contain only geometrical information. To handle CAD models without a topology, we can extract topological data such as vertices, edges, and loops from the geometrical data. For example, edges from the surface boundary curves (IGES entity types 102, 126, 141, and 142) and the faces from the surface data (types 143, 144, and 510). As a result, we can identify the following 12 error items contained in the IGES format:

1) Edge (Curve) Length

2) Edge (Curve) Curvature

3) Edge–Edge Gap

4) Edge–Edge Angle

5) Edge–Face Gap

6) Edge–Edge Proximity

7) Edge–Loop Consistency

8) Surface–Surface Consistency

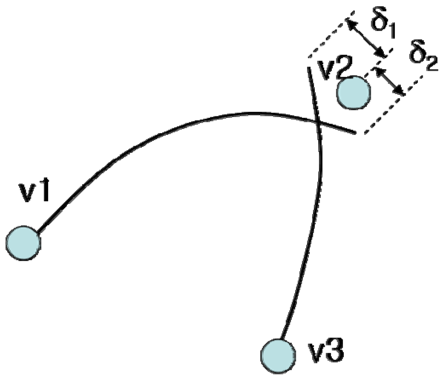
9) Surface Degree

10) Surface–Surface Gap

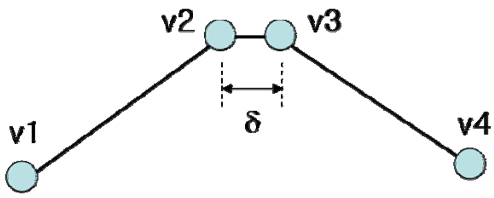
11) Surface–Surface Angle

12) Void–Surface

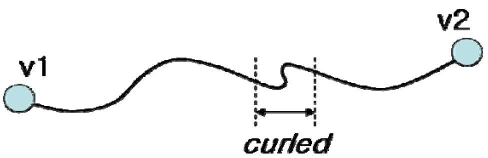
Nomenclature- v_i: the i^{th} vertex; e_i: the i^{th} edge



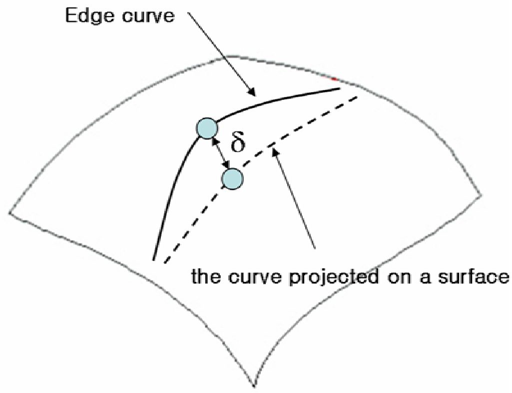
(a) Edge–Vertex(or Edge) Gap



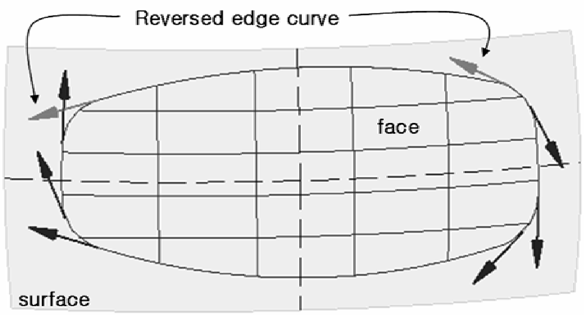
(b) Edge Length



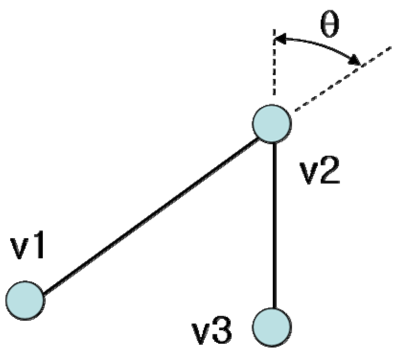
(c) Edge(or Curve) Curvature



(d) Edge–Face Gap

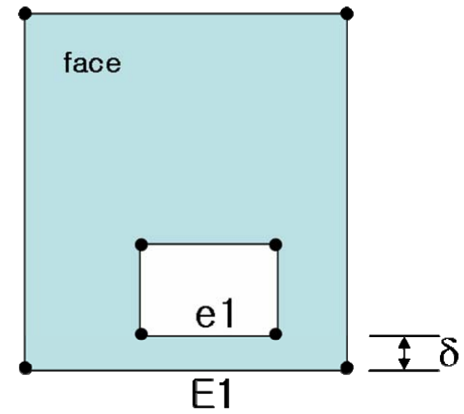


(e) Edge–Loop Consistency

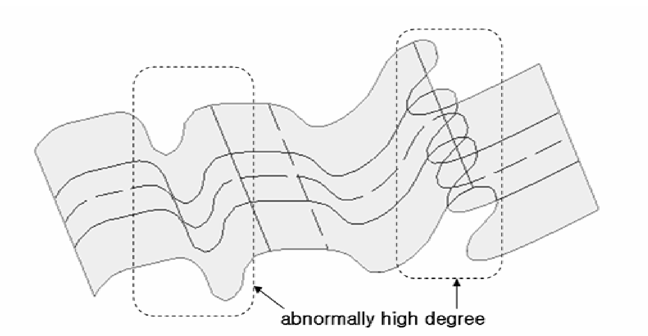


(f) Edge–Edge Angle

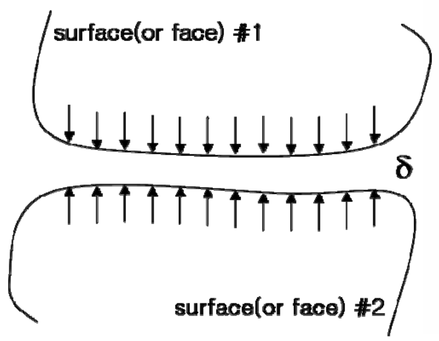
Figure 1. Representation of error items (a to f)



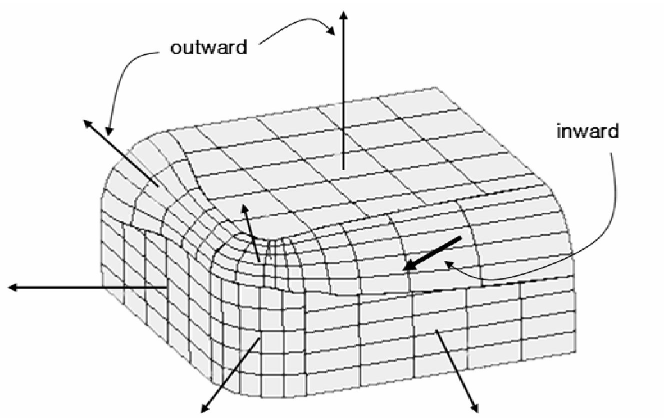
(g) Edge–Edge Proximity



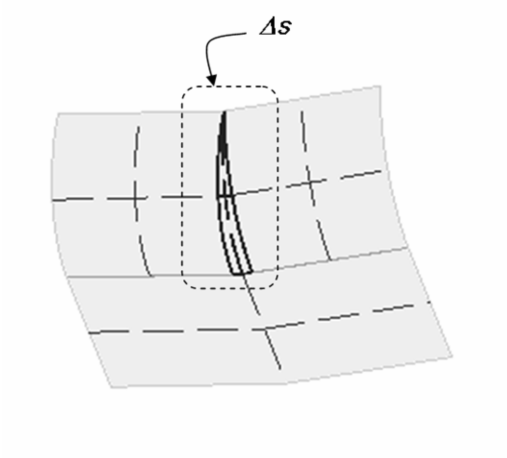
(h) Face(or Surface) Degree



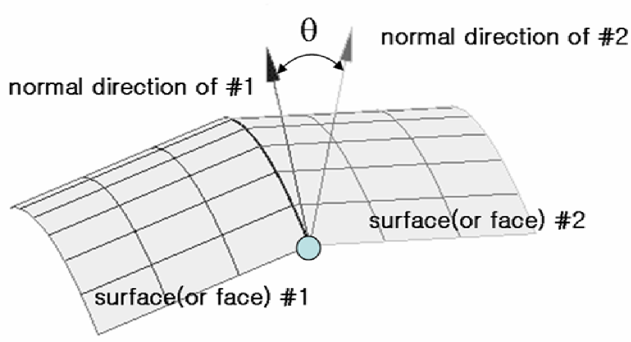
(i) Face(or Surface) –Face(or Surface) Gap



(j) Face–Face Consistency



(k) Void–Face(or Surface)



(l) Face(or Surface) –Face(or Surface) Angle

Figure 2. Representation of error items (g to l)

2.2 Approaches for STEP Models

STEP Tools' ST-Developer V8.0 is used to read and manage STEP models with topological structures. Browsing and searching all the data stored in the internal data structure is possible using ST-Developer APIs and our own evaluation functions. The verification result is managed as an independent data structure. We have used OpenCASCADE v3.0 modules in which low-level functions are available for mathematical analysis and the verification modules which are implemented for geometric analysis. The mapping module allows communication of the internal data structure with the ST-Developer and OpenCASCADE. We have used the following seven *shape_representation forms* of STEP when creating a part:

1. *advanced_brep_shape_representation*
2. *manifold_surface_shape_representation*
3. *geometrically_bounded_surface_shape_representation*
4. *faceted_brep_shape_representation*
5. *geometrically_bounded_wireframe_shape_representation*
6. *hybrid_3d_shape_representation*
7. *csg_shape_representation*

Only *advanced_brep_shape_representation*, *manifold_surface_shape_representation*, and *geometrically_bounded_surface_shape_representation* are allocated for the full error verification process. The others ended with a warning sign. In particular, the shape *faceted_brep_shape_representation* only sends the message '*faceted_solid is used*' to the designer, which means a faceted solid is used in this STEP model, and the message is written at the end of the report file.

As shown in Figure 3, ① *advanced_brep_shape_representation* is used for normal solid data, and ② *manifold_surface_shape_representation* has a topological set of faces that consist of a shell unit. As both cases consist of a number of faces with a list of edges, we can find topological errors based on these entities. However, since *geometrically_bounded_surface_shape_representation* consists of a number of surfaces without a topological relationship, we can find those errors after creating boundary curves from each surface. The bigger the STEP data file becomes, the more time and memory is spent on searching the internal data and confirming its connectivity. Consequently, to improve the verification performance, we have grouped the verification processes and their functions based on similarity and organized their processing procedurally, as shown in Figure 3.

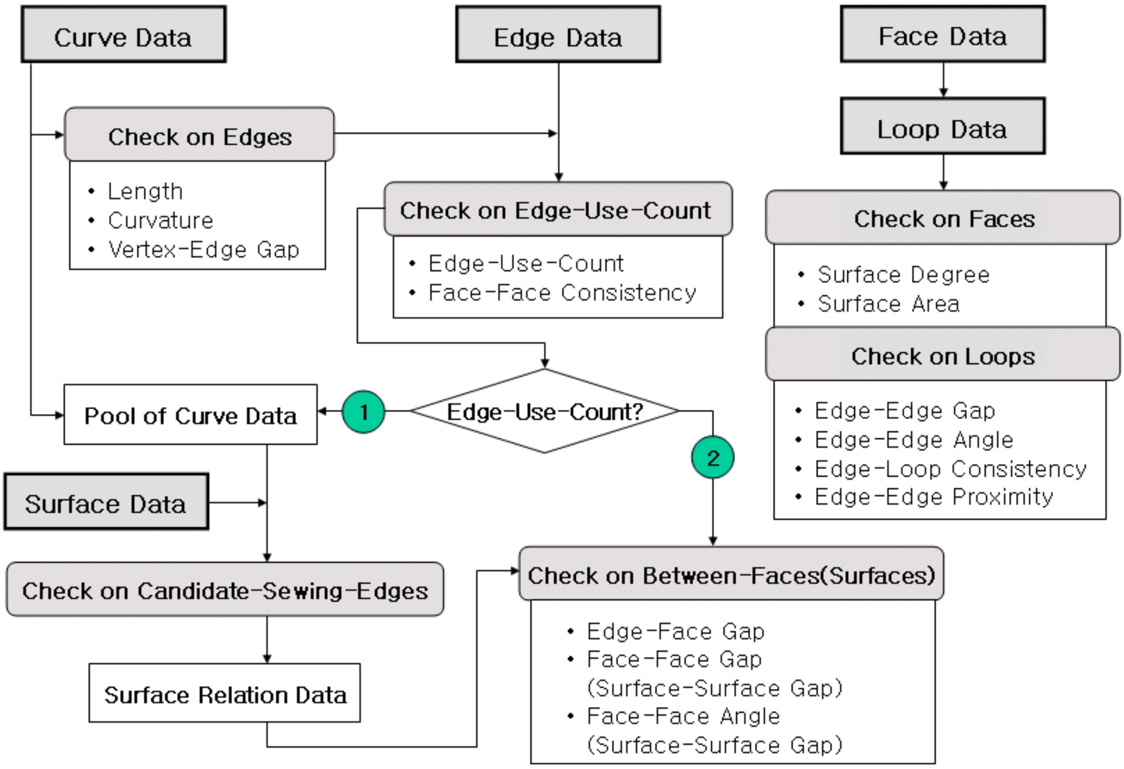


Figure 3. The procedural verification process of the STEP model

2.3 Approaches for IGES Models

An IGES model consists of geometrical data, not topological data—that is, IGES consists of point entities, line entities, curve entities, surface entities, and solid entities. To implement verification algorithms for geometrical entities without topology, we extract topological data such as vertices, edges, and loops from the boundary curves (IGES entity types 102, 126, 141, and 142) for a surface that has a geometrical datum. Similarly, faces are extracted from a surface (types 143, 144, and 510) with boundary curves. For the surface that does not have boundary curves (types 108, 114, 118, 120, 122, 128, and 140), we find out boundary curves using U-parameters and V-parameters.

There are two verification workflows; Process I for the boundary curve data and Process II for the surface data. The whole process finishes in three steps. In the first step, the evaluation processes of edge (curve), loop, and surface are executed in sequence. The verification process of the candidate sewing edge is achieved in the second step. In the last step, based on the preceding two processes, the verification of between surfaces is executed. To improve the performance of the verification process and to avoid redundancy in the verification modules, we divided the two processes based on the verification targets and the three layers according to the verification tasks, as illustrated in Figure 4. In addition, independent modules allow individual verifications. The verification system for IGES models consists of 13 modules including the verification of the candidate sewing edge.

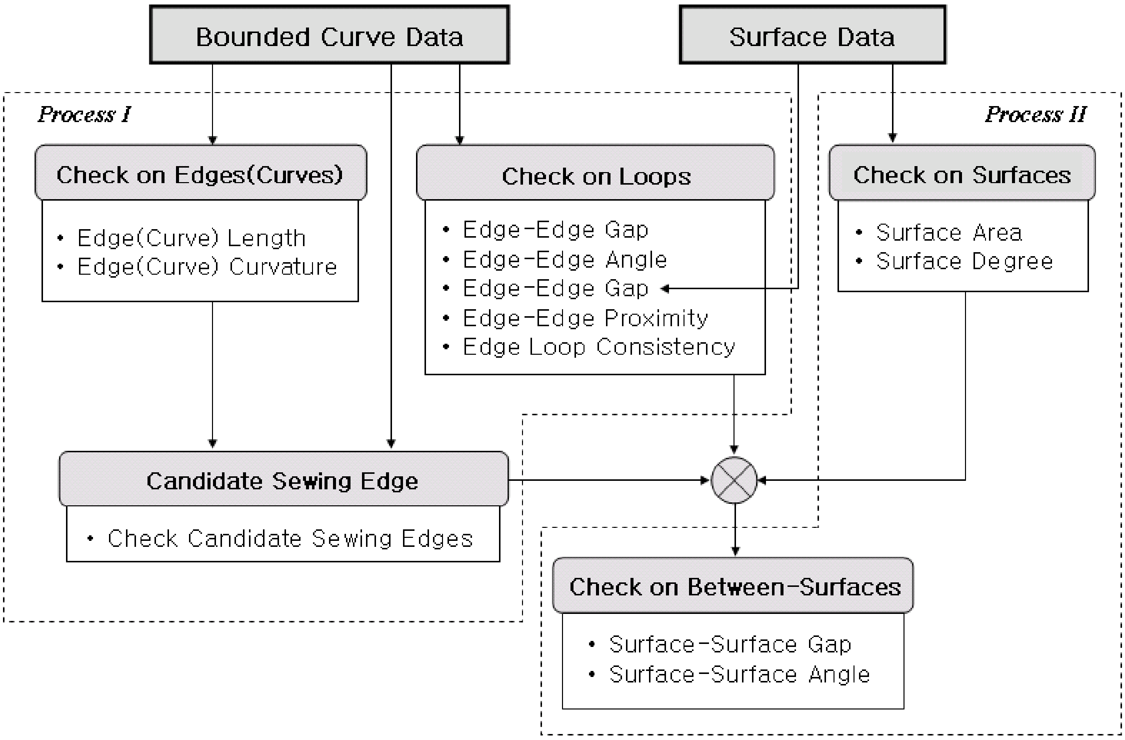


Figure 4. The procedural verification process of the IGES model

2.4 Categorization of Verification Items

Category	Verification Item	Description
Edge Errors	Large gap between vertex and edge	C^0 discontinuity. After measuring a distance between 3D points linked to a vertex and a point that is projected on a curve from that vertex point, check if its distance is longer than the user's input value.
	Small length of an edge or a curve	Tiny segment. Consider zero-length if the length of a section curve determined by each vertex is smaller than the given value or if the coordinate values of two vertices on an edge are almost equal.
	High curvature of an edge or a curve	Curled segment. After obtaining two parameters from a curve, divide them into twenty pieces and measure the curvature at the point of its curve corresponding to each parameter.
Loop Errors	High degree of faces or surfaces	Check if the excessive high degree in the U/V parameters of a B-spline surface is used.
	Large gap between an edge and a face	The boundary of the surface is generated by 3D curves in 3D space or by 2D curves in a 2D plane. For a 2D curve, no gap exists between the face and the edge because that curve is on the U/V parameter plane. However, in most cases the gap is caused by 3D curves while mapping from the 2D parametric domain to the 3D space domain—this phenomenon is due to the CAD system's tolerance.
	Edge/loop consistency	Inspect the connectivity of every loop, which in turn consists of a list of edges, according to their directional meaning. The direction of each edge, in case of a STEP file, is determined by the two vertices edge_start and edge_end. In addition, oriented_edge is used for giving a directional meaning to edges in a loop.
	Large gap or overlap between edges	Check the length of the gap between two neighboring edges by measuring the distance between every projection point of the vertex that shared two connected edges.

Category	Verification Item	Description
Loop Errors (Cont.)	Sharp angle between edges	Measure the angle between two tangent vectors from the left edge and the right edge on the basis of a vertex where the two edges meet. The verification process is used not only for the neighboring state between two edges but also for unreal anomalous shapes.
	Edge proximity	Producing distorted faces such as a long, thin shape is possible due to a designer fault or a difference in the tolerance of CAD systems. Most proximity errors happen when an edge has zero length. Check if a right edge and a left edge meet at a zero gap.
Face Errors	Edge-use-count	Basic information to find the errors connected with faces. When the type of shape in STEP is *shell_based_surface_model*, for example, the edge-use count of many edges is 1 because although a shell has internal topological information the STEP model of *shell_based_surface_model* does not include the connectivity between shells. So, it is hard to find the connectivity between two faces without the edge-use-count process.
	Large gap and overlap between faces	Check if the gap between the two faces sharing the edge is more than the input value.
	Sharp angle between faces	The angle problem occurs between two faces in two situations: when two faces make a sharp angle, and when the information of the face's normal vector is incorrectly specified. The first situation, which produces an impractical shape, is likely to make trouble in the CAM process; the other is an incorrect linkage between two faces due to the CAD translator's error or a design fault.
	Inconsistency between faces	In a solid model, one edge is used two times by two faces. Although the edge should have been used for the opposite directions of both faces, if its directions are not alternately used in each face, the normal vector could not keep the two joined faces consistent.

Category	Verification Item	Description
Connectivity of surfaces without topology	Candidate sewing edge	Although there is no topological information between edges whose edge use count was 1, we checked the geometrical connectivity using maximum distances and angles between two surfaces. For a surface, we created boundary curves because there was no definite edge to divide its border. To create the boundary curves of a surface without topology, we applied U and V parameters in the parametric domain. Furthermore, for the surfaces with boundaries such as the *geometrically_bounded_surface* of STEP, we checked their connectivity by handling their boundaries with edges as topological meaning.
	Void face	During the conversion from a CAD system to STEP or IGES, some faces occasionally disappear. If that problem happens, a CAD model no longer remains solid but is turned into face unit data or surface unit data. So, check the possibility of a void face by checking whether those edges were linked to each other and got into a loop.

3. Implementation

3.1 System Architecture

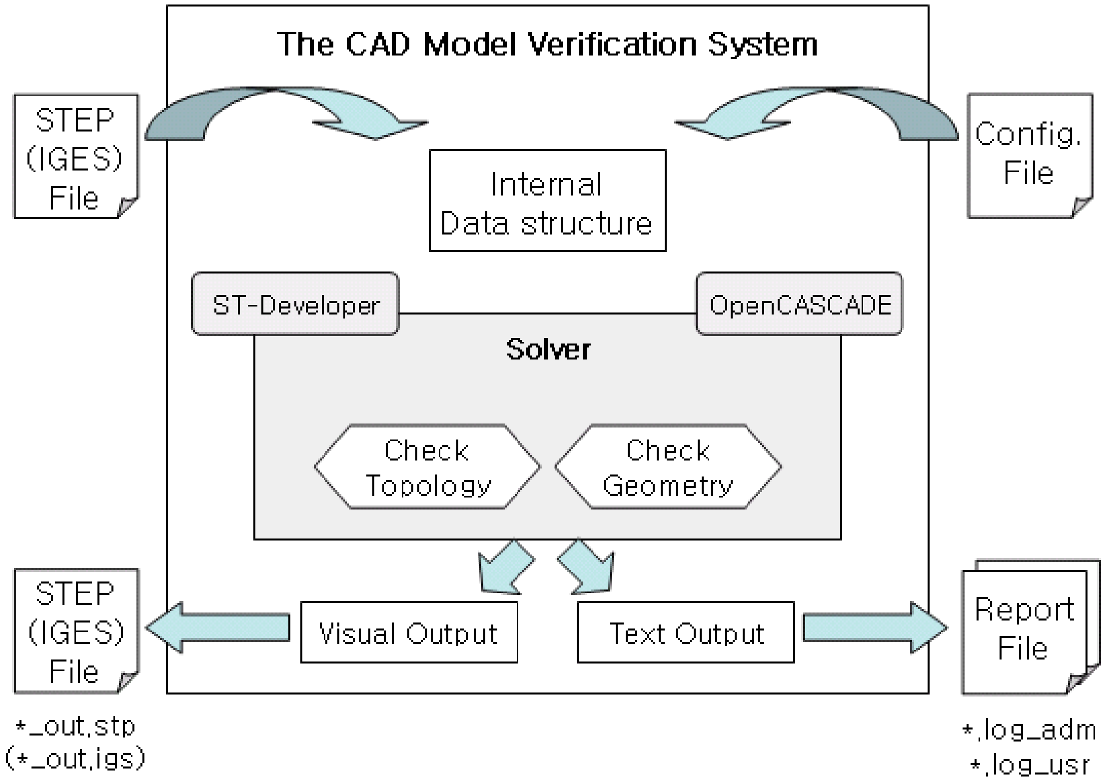


Figure 5. Architecture of the CAD model verification system

As shown in Figure 5, a configuration file that includes verification conditions and criteria, as well as a model file such as STEP or IGES, are loaded into the verification system. The STEP file is translated into the internal data structure through the ST-Developer. The verification operation for topology and geometry is performed by means of the low-level APIs of OpenCASCADE such as mathematical computations in addition to the solver functions that we have developed. The postprocessor of this system adds special entities to the geometric entities or shapes where the errors occurred. The added entity is a color, a descriptive text, or a symbol. The purpose of adding the entity is to help designers to locate the errors. The system produces two types of verification results: one type for ordinary users and one for system administrators.

3.2 Input and Output Files

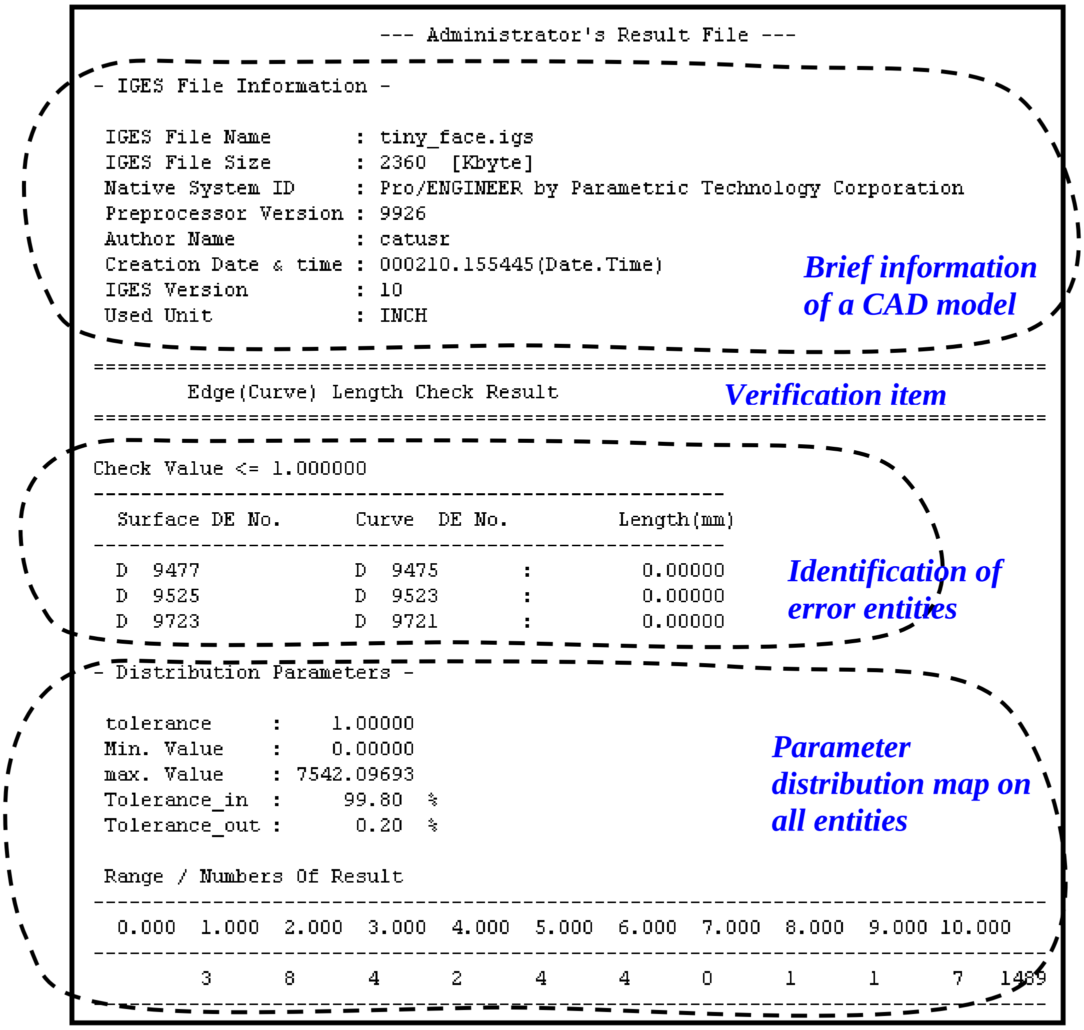
3.2.1 Input Configuration File

The configuration file is divided into three areas: the quantitative criterion area for each verification item; the area for the error indication; and the area for reporting options. The file can be customized according to the design guides prescribed by a user company.

3.2.2 Output Log File

After we have completed the error verification of 19 STEP items and 12 IGES items, the results are expressed as a log file and a modified physical STEP or IGES file to help designers making decisions. In ASCII form, as shown in Figure 6, the log file includes a general description of the checked model, the verification results, and the identities of the erroneous entities.

Furthermore, to help handling erroneous entities, the system generates a modified STEP or IGES model with special signs and symbols such as color or indicators as requested by the configuration file. For instance, in the IGES output log file, which is shown in Figure 6, the first block includes a general description of the IGES model such as the file name, size, and version. The second block shows criteria values that the designer inputted, and the location of error entities, inclusive of their parents and values. In the last block, maximum and minimum values for the verification and a deviation ratio on error entities compared to criteria are included. In addition, we have used the reference table to learn, on the basis of criteria, the number of entities for verification in any range. As a result, we can determine from the table whether an entity is good or bad, or just issue a warning.



```
                    --- Administrator's Result File ---
- IGES File Information -

 IGES File Name        : tiny_face.igs
 IGES File Size        : 2360  [Kbyte]
 Native System ID      : Pro/ENGINEER by Parametric Technology Corporation
 Preprocessor Version : 9926
 Author Name           : catusr
 Creation Date & time : 000210.155445(Date.Time)
 IGES Version          : 10
 Used Unit             : INCH
================================================================================
        Edge(Curve) Length Check Result
================================================================================
Check Value <= 1.000000
-----------------------------------------------------
  Surface DE No.       Curve  DE No.         Length(mm)
-----------------------------------------------------
  D  9477              D  9475      :          0.00000
  D  9525              D  9523      :          0.00000
  D  9723              D  9721      :          0.00000
- Distribution Parameters -

 tolerance     :    1.00000
 Min. Value    :    0.00000
 max. Value    : 7542.09693
 Tolerance_in  :      99.80  %
 Tolerance_out :       0.20  %

 Range / Numbers Of Result
--------------------------------------------------------------------------------
  0.000  1.000  2.000  3.000  4.000  5.000  6.000  7.000  8.000  9.000 10.000
--------------------------------------------------------------------------------
         3      8      4      2      4      4      0      1      1      7   1489
--------------------------------------------------------------------------------
```


Figure 6. An output file for the administrator

3.3 Implementation Environment

The verification system has been implemented into two types; a batch processing program for UNIX machines and an interactive application with a visualization module and a graphical user interface (GUI) for Windows NT. Its implemented environment is as follows:

- Programming language: ANSI C/C++;
- Graphics library and user interface: OpenGL, Microsoft Foundation Class;
- Geometric modeling kernel: OpenCASCADE v3.0
- STEP interface: ST-Developer 8.0
- Operating system: MS Windows 2000, Silicon Graphics IRIX 6.5

Figure 7 shows an example of the verification system with MS Windows 2000. The user interface consists of four sections: viewing tools, visualization area, main control panel, and status window.

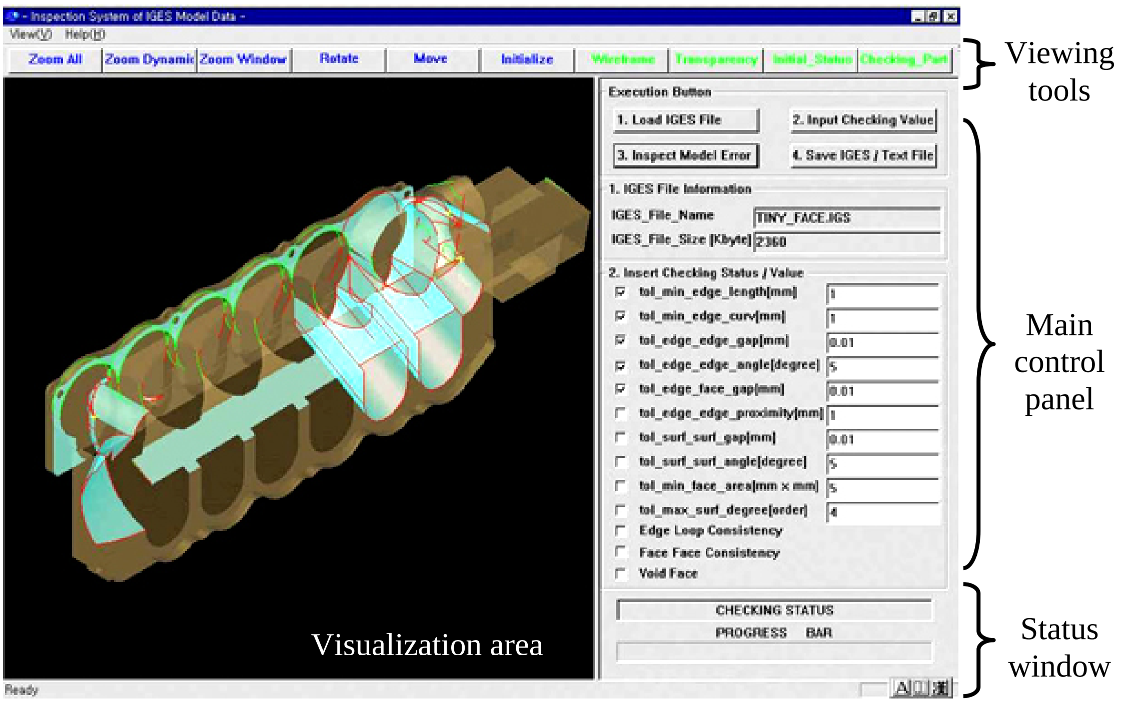
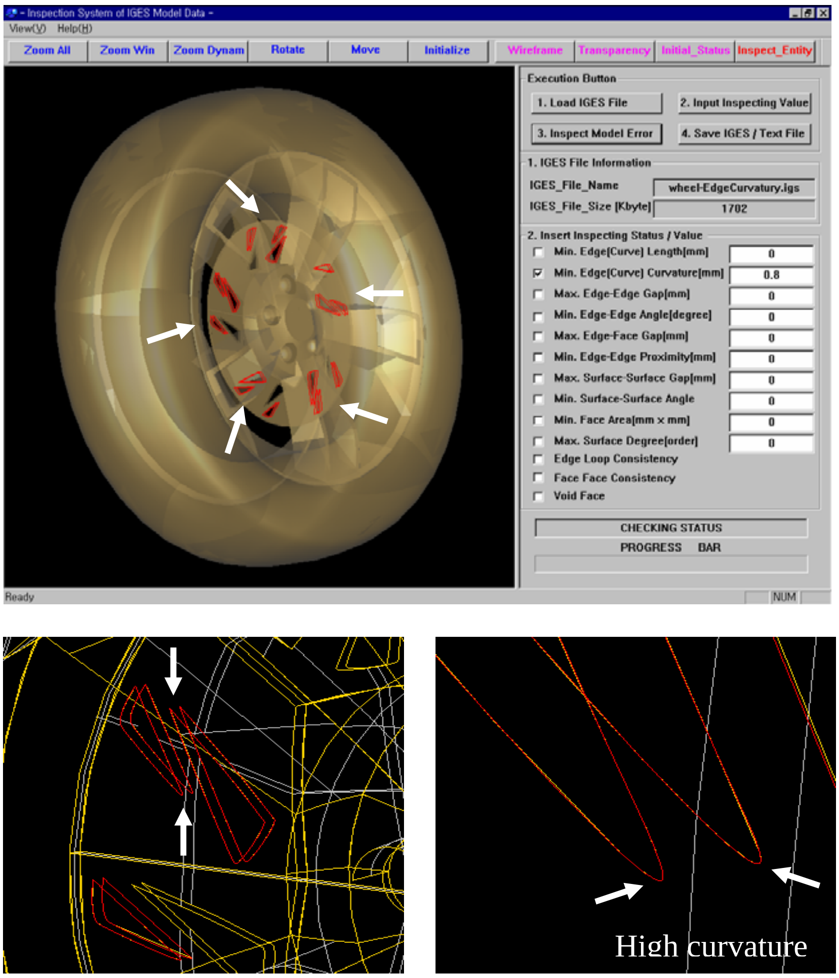


Figure 7. User interface of the interactive application in Windows 2000

4. Case Study

Case 1: Edge (Curve) Curvature

An engineering designer occasionally prefers to use a higher radius of curvature for a curved shape. However, it can be a poor design for the style design, dissociation when being offset, and excessive unsatisfactory cuts in line processing of CAM. Figure 8 shows the verification of edge(curve) curvature of an automobile rim. The model with the file size of 8.3 mega bytes is an IGES format converted from Pro/CDRS V2000i. We found 5 sets of modeling errors along the radius of curvature below 0.8.



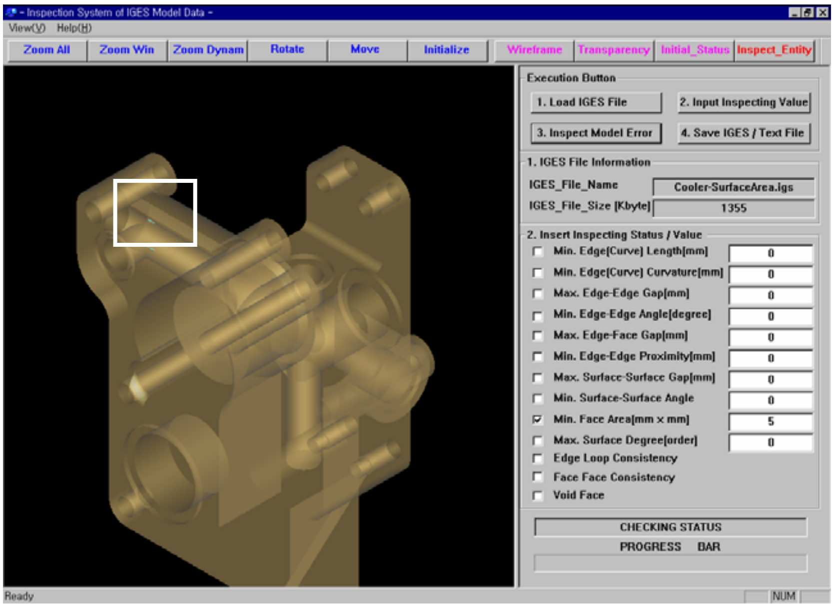
```
=================================================
=              Edge(Curve) Curvature Check Result
=================================================
=
Check Value <= 0.800000
------------------------------------------------------------
 Surface DE No.    Curve  DE No.   Radius of Curvature(mm)
------------------------------------------------------------
 D  2257        D  2083     :      0.29489
 D  2261        D  2077     :      0.03658
 D  2267        D  2097     :      0.14589
 D  2271        D  2143     :      0.26036
 D  2273        D  2145     :      0.26036
(diminished)

- Distribution Parameters -
 tolerance    :  0.80000
 Min. Value   :  0.03658
 max. Value   : 10000000000000000.00000
 Tolerance_in :    95.93  %
 Tolerance_out :    4.07  %

 Range / Numbers Of Result
------------------------------------------------------------------------------------------
 0.000  0.800  1.600  2.400  3.200  4.000  4.800  5.600  6.400  7.200  8.000
------------------------------------------------------------------------------------------
    35     0     30     0      0      0      0      0      0      0    794
------------------------------------------------------------------------------------------
```

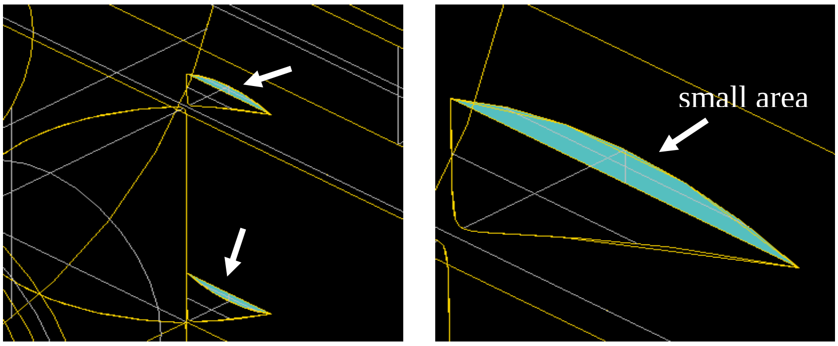
Figure 8. Verification result on the edge (curve) curvature

Case 2: Void-Face (Surface)

A model shape with tiny face(surface) cannot define cross-sections and cannot be projected. No offset surface can be achieved. It creates unsatisfactory mesh quality in CAE. Moreover, an excessive amount of time is required for analytical calculation. In CAM, it requires unsuccessful cutting location (CL) calculations and cannot be manufactured.

Figure 9 shows the verification result of an automobile toolbox surface area (Void-Surface). The automotive toolbox with file size of 11.7 mega bytes is an IGES model converted from Pro/Engineer. Since the minimum surface area specified in the configuration file is 5 mm^2, the system paints cyan color on the surface areas below 5 mm^2 and displays them on the monitor. In the wireframe window on the lower right, we zoom in the part to see the error flake that is generated when the cross curves are created at the intersection of the two pipes. The error flake is so small that a designer cannot visually perceive it during modeling or verification steps. Finally, we insert color or indication symbols into the erroneous entities, and we save the model as another IGES model so that it can be corrected in downstream CAD systems.





```
==================================================
              Surface  Area Check Result
==================================================
Check Value <= 5.000000
----------------------------------------
 Surface DE No.        Area(mm*mm)
----------------------------------------
     D   93       :      0.00797
     D  119       :      0.04689
     D  143       :      2.66061
     D  163       :      2.66061
     D  767       :      0.00797
     D 1713       :      0.04689
     D 2217       :      2.66061
     D 2237       :      2.66061

- Distribution Parameters -
 tolerance     :   5.00000
 Min. Value    :   0.00797
 max. Value    : 26063.76716
 Tolerance_in  :    94.37 %
 Tolerance_out :     5.63 %

 Range / Numbers Of Result
------------------------------------------------------------------------------
 0.000  5.000 10.000 15.000 20.000 25.000 30.000 35.000 40.000 45.000 50.000
------------------------------------------------------------------------------
   8      0      2      0      3      2      0      0      0      3     124
------------------------------------------------------------------------------
```

Figure 9. Verification result on the void-face (surface)

Case 3: Face consistency

We have checked the face inconsistency on a STEP pipe model of an automobile manifold with the file size of 19.5 mega bytes, which is converted from CATIA. This case is invisible to ordinary sight. Upon closer examination, we can find that normal vectors where two faces met in the opposite direction to one another. A directional mismatch may result from the offset operation. Moreover, it can abort a tool path generation process in CAM.

Identifying the angular error between faces by an external visual inspection is difficult. Most angular errors are due to the incorrect normal vector of a face that indicates the opposite direction to each other. If switching on '*back culling*' on the visualization option in the CAD system, we noticed that a face with incorrect normal vectors disappears as shown in the bottom right of Figure 10.

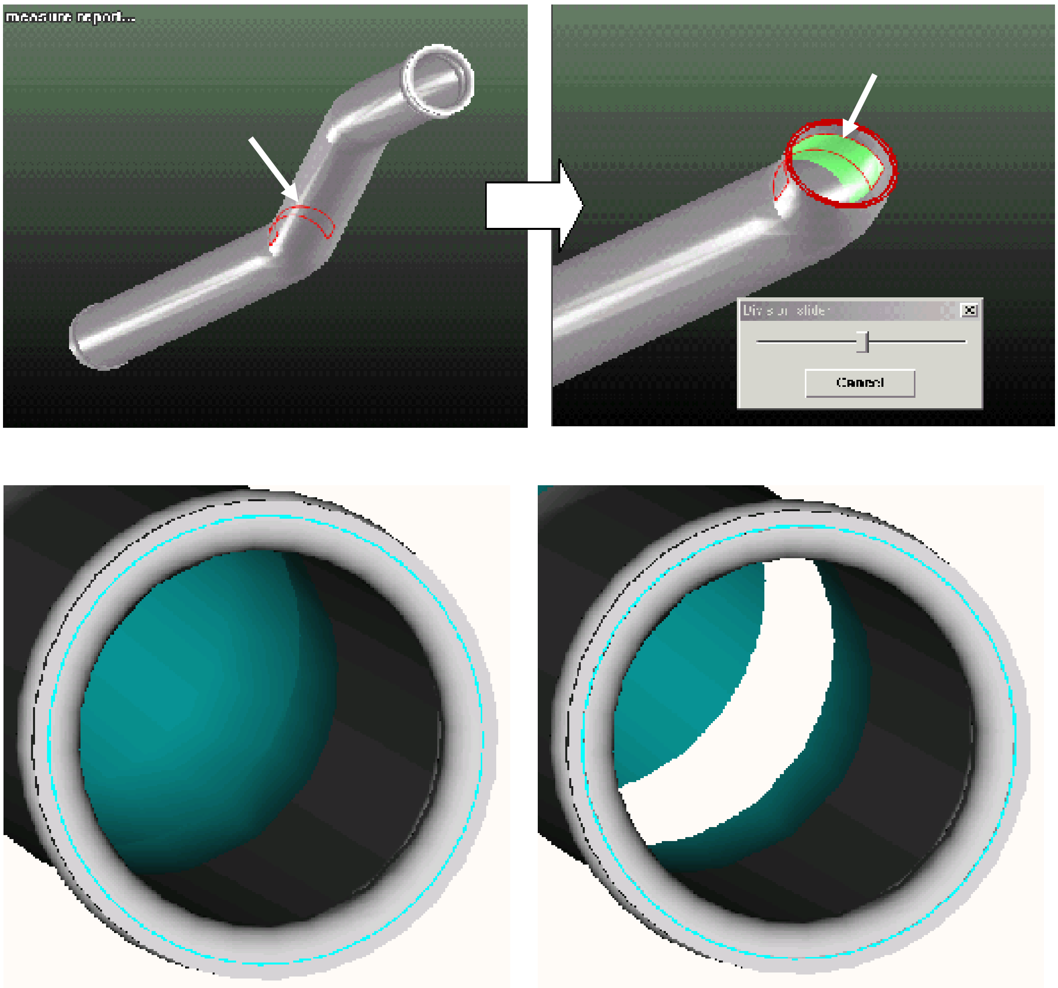


Figure 10. Verification result on the face consistency

5. Conclusion

Although commercial CAD systems have recently shown stronger power and various functions, one problem that distresses designers is the CAD model errors. There are three scenarios for the source of these errors: designers who use the wrong modeling practices but do not obey the rules required by the CAD system vendor; bugs or technological faults in the CAD system itself; and incorrect translations that cause data loss and anomalies in the CAD model while exchanging data between heterogeneous CAD systems.

This paper introduces a verification method for CAD model errors, especially the errors that occur while converting a commercial CAD model into a neutral model. Until now, most methods of coping with the problem have depended entirely on the CAD system developer and vendor, who know the internal data structure. Likewise, related studies on the errors of the neutral model, which is frequently used in the field, are inactive.

A brief summary of this paper is as follows:

- After considering their structural differences, we analyzed the topological and geometrical errors that can occur on a CAD model, especially neutral models such as STEP and IGES.
- We proposed a procedural method for verification of CAD model errors to save system resources such as memory and CPU sharing. The method has an advantage for huge CAD models.
- To check errors in IGES or STEP models which do not have topological information, we applied a method that extracts topological entities form the geometry.

- Based on the analysis, we developed a verification system of CAD model errors; for Windows2000 with visualization and GUI, and for UNIX with a batch-processing mode. Using this system, we have tested automotive part models.

Acknowledgements

We are grateful to Hyundai Motors for their financial support of this research.

Reference

Barequet, G., 1997, “Using geometric hashing to repair CAD objects,” IEEE Computational Science & Engineering: October-December 1997, pp. 22~28.

Barequet, G., Shapiro, D. and Tal, A., 1996, “History-driven reconstruction of polyhedral surfaces from parallel slices,” Proceedings of the IEEE Visualization, San Fransisco, pp. 149~156.

Deshpande, V., Fornasier, L., Gerteisen, E. A., Hilbrink, N., Mezentsev, A., Merazzi, S. and Wöhler, T., 2000, “Virtual engineering of multi-disciplinary applications and the significance of seamless accessibility of geometry data,” Future Generation Computer Systems, Vol. 16, pp. 435~444.

Gu, H., Chase, T. R., Cheney, D. C., Bailey, T. and Johnson, D., 2001, “Identifying, Correcting, and Avoiding Errors in Computer-Aided Design Models Which Affect Interoperability,” Journal of Computing and Information Science in Engineering, Vol. 1, pp. 156~166.

Hoffman, C. M. and Robert, J. A., 1998, “CAD and the product master model,” Computer-Aided Design, Vol. 30, pp. 905~918.

ISO Committee, 1997, “ISO Committee Draft 10303-214: Industrial automation systems and integration - Product data representation and exchange - Part214: Application Protocol: Core data for automotive mechanical design process,” ISO TC184/SC4 N577.

Japan Automobile Manufacturers Association, 2002, “JAMA/JAPIA PDQ Verification Model - Data Specifications,” JAMA Electronic Information Exchange Committee, JAMAEIC005.

Choi, K., Mun, D. and Han, S., 2002, “Exchange of CAD Part Models Based on the Macro-Parametric Approach,” International Journal of CAD/CAM (www.ijcc.org), Vol. 2, pp. 23~31.

Reed, K., Harrod, D. and Conroy, W., 1990, “The Initial Graphics exchange Specification(IGES) Version 5.0,” U.S. Department of Commerce.

Steinbrenner, J. P., Wynman, N. J. and Chawner, J. R., 2001, “Procedural CAD Model Edge Tolerance Negotiation for Surface Meshing,” Engineering with Computers, Vol. 17, pp. 315~325.

Tassey, G., 1999, “Interoperability cost analysis of the U.S. Automotive supply chain – Final report,” RTI Project Number 7007-03, Research Triangle Institute.

Volpin, O., Sheffer, A., Bercovier, M. and Joskowicz, L., 1998, “Mesh simplification with smooth surface reconstruction,” Computer-Aided Design, Vol. 30, pp. 875~882.

Yang, J., Goltz, M. and Han, S., 2004, “Parameter-Based Engineering Changes for a Distributed Engineering Environment,” Concurrent Engineering: Research and Applications, Vol. 12, No. 4, pp. 275~286.

Yang, J., Han, S., Cho, J., Kim, B. and Lee, H., 2004, “An XML-Based Macro Data Representation for a Parametric CAD Model Exchange,” Computer-Aided Design and Applications, Vol. 1, pp. 153~162.

EDS.com: Products, http://www.eds.com/products/plm/teamcenter/vis/view/

OpenCASCADE Homepage, http://www.opencascade.com/

STEP Tools Ltd., http://www.steptools.com/products/stdev/index.html