

Internet-based dimensional verification system for reverse engineering processes

In-Ho Song¹, Kyung-Don Kim² and Sung-Chong Chung^{3,*}

¹*Division of Mechanical Design, Ajou University, Suwon 443-749, South Korea*

²*Gyeonggi Regional Headquarter, Small Business Corporation, Suwon 443-766, South Korea*

³*School of Mechanical Engineering, Hanyang University, Seoul 133-791, South Korea*

(Manuscript Received November 13, 2006; Revised April 1, 2008; Accepted April 11, 2008)

Abstract

This paper proposes a design methodology for a Web-based collaborative system applicable to reverse engineering processes in a distributed environment. By using the developed system, design reviewers of new products are able to confirm geometric shapes, inspect dimensional information of products through measured point data, and exchange views with other design reviewers on the Web. In addition, it is applicable to verifying accuracy of production processes by manufacturing engineers. Functional requirements for designing this Web-based dimensional verification system are described in this paper. ActiveX-server architecture and OpenGL plug-in methods using ActiveX controls realize the proposed system. In the developed system, visualization and dimensional inspection of the measured point data are done directly on the Web; conversion of the point data into a CAD file or a VRML form is unnecessary. Dimensional verification results and design modification ideas are uploaded to markups and/or XML files during collaboration processes. Collaborators review the markup results created by others to produce a good design result on the Web. The use of XML files allows information sharing on the Web to be independent of the platform of the developed system. It is possible to diversify the information sharing capability among design collaborators. Validity and effectiveness of the developed system has been confirmed by case studies.

Keywords: ActiveX; Collaborative system; Dimensional verification; Reverse engineering; Viewer; Web

1. Introduction

The development of information technology and widespread availability of the Internet are strongly urging traditional manufacturing processes to enter the e-manufacturing environment. The manufacturing industry pursues reduction in cost and time-to-market by adopting the concept of remote design and manufacture. This concept consists of usage of the Internet and digital production technologies. The Internet technology allows collaborators, designers and manufacturers to be at different places.

The digital production technologies of CAX (Computer Aided X) systems, such as CAD, CAM, CAI, etc., are already validated as powerful systems in

manufacturing. Owing to these advantages of the remote design and manufacture, the collaborators do not need to be together in one place or use the same computing environment. They are free from the restriction of space and tools. However, the closed CAX systems still seem to have restrictions affecting the collaborators. The general CAD system tacitly forces its users to use CAM/CAE/CAI systems which are subordinated in the CAD System. Directly speaking, those packages have unique characteristics dependent on their producers. Justly, the characteristics cannot be shared with other systems.

In order to overcome this disadvantage, new technologies for distributing product information have been developed by people who do not want proprietary or high-end software systems. Among them, one promising technology is Web-based collaborative visualization, which enables its users to visualize,

*Corresponding author. Tel.: +82 2 2220 0444, Fax.: +82 2 2298 4634
E-mail address: chung@hanyang.ac.kr
DOI 10.1007/s12206-008-0408-7

annotate, and control 3D design models interactively over the Internet [1]. Efforts have been devoted to construct Web-based design and manufacturing systems [2, 3]. Ahn et al. proposed an Internet-based CAD/CAM system, CyberCut, developed by using Java applet [4]. An agent-based collaborative design system on the Internet, PACT (Palo Alto Collaborative Testbed), using KQML (Knowledge Query and Manipulation Language) and LISP agents was developed by Mark et al. [5]. Pang and Wittenbrik studied a shared viewing environment called Cspray to assist distributed users [6]. Using Cspray, users see their data interactively from independent or shared camera positions. In addition, O'Grady proposed asynchronous two-way communication, interactive 3D viewing and markup methods for STEP data [7]. In this system, the shape abstraction and client-server communication techniques were studied for the Internet-based CAD systems. As a data format, this system adopted a polygonized CAD model (VRML) which is good for visualization. However, the VRML adopted system cannot inspect dimensions accurately.

Previous researches [1-5] apply the concept of remote design and manufacture to fabricate products from CAD information. But the reverse engineering process on the Internet, called Web-based reverse engineering, has not been studied so far. The reverse engineering means a process extracting a CAD model from measured point data obtained through a 3D digitizing process by using a layout machine shown in Fig. 1. In addition, AP219 of ISO 10303 [8], DMIS [9], and dmlspec (Dimensional markup language) [10] are well known as standards for inspection. These standards are good for inspection, but not applicable to expressing reverse engineered data. Com-

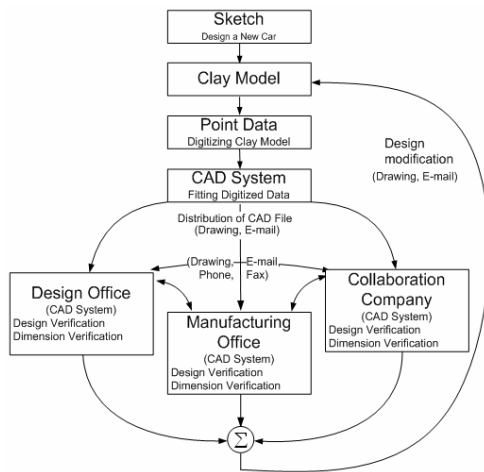


Fig. 1. Layout machine for 3D digitizing.

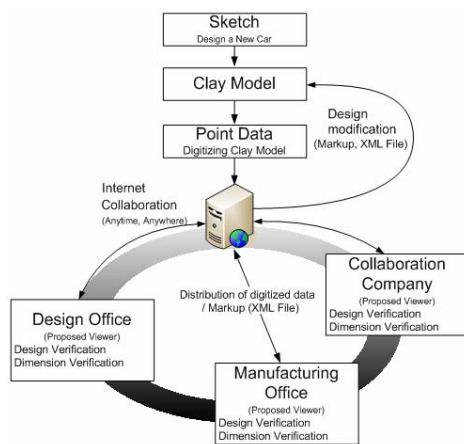
mercial viewer tools, such as AutoVue of Cimmetry Inc. [11], SpinFire of Actify Inc. [12], and Hoop3D of Techsoft Inc. [13] have been successfully deployed in various industrial applications with their rich user interfaces and functions. But those systems have insufficient quality for the reverse engineering process. Some common reverse modeling tools are as follows: Rapidform of INUS Technology Inc. [14] and Rhinoceros of McNeel Inc. [15]. Both tools provide various functions, but they are very sophisticated to use and they do not support the Internet environment.

Making a trial car or modifying the design of an existing model requires digitizing processes to obtain point data from the clay model as shown in Fig. 1. The digitized point data are converted into a CAD model for verification of the design specification as shown in Fig. 2(a). Since the CAD model does not fully consider the manufacturing or technical information necessary in the early stage of designing, it should be distributed to automobile engineers and to collaborative designers or companies to examine the dimension, geometric and aesthetic shapes, interference, manufacturability, etc. Designers collect the verified results and revise the clay model through the phone, fax, email and/or drawing. Generally, in order to confirm the interior or exterior shape of a car, the design-verification-modification process should be iterated, which requires close cooperation among developers. Long distance iteration requires a greater amount of time and effort, and a longer product development cycle in a conventional development process like Fig. 2(a).

In this paper, to reduce the lead-time and improve flexibility of the reverse engineering process, a web-based dimensional verification system is studied. Conventional measurement and inspection processes are improved as shown in Fig. 2(b). Point data measured on a layout machine is directly used to verify design specifications. As shown in Fig. 2(b), a design supervisor (or digitization supervisor) registers point data on the developed web server. Conversion of the point data into a CAD file is not required. This system enables collaborators separated by time and distance to verify geometric shapes of a product and its dimensional information on the Internet. The collaborators are also able to share and exchange their views and opinions over the Internet through the developed viewer, markup functions and XML files as shown in Fig. 2(b). The resolution of design conflicts with others in the early design stage is performed on the Web.



(a) Conventional development process



(b) Proposed development process

Fig. 2. Reverse engineering processes to develop a new car.

This reduces the product development lead-time and manufacturing cost significantly.

The next section describes functional requirements of the Web-based dimensional verification system and current technologies applicable to the system. Overall architecture of the developed system is described in section 3. Section 4 shows how to design the dimensional verification and markup modules. A case study is included in section 5 to verify usefulness of the developed system. Section 6 concludes.

2. Design concepts

In order to construct a Web-based dimensional verification system for reverse engineering processes,

five functional requirements are devised as follows:

Excellent graphic performance is required for examining geometric shapes of a product, which is described in section 3.

The visualization and dimensional verification functions of measured point data over the Internet are described in section 4.

Automatic installation of additional programs for the reverse engineering on the client PC is described in section 4.1.

In order to improve the accuracy of the dimensional verification, direct visualization of the measured data is required on the Web instead of conversion of the point data into VRML or CAD format files, which is described in section 5.2

It is required to offer convenience to design reviewers during the dimensional verification operation on the Web, which is described in section 5.3.

3. Visual realization methods for Web-based 3D application

To construct the Web-based dimensional verification system for reverse engineering processes, 3D shape of point data must be visualized and handled on the Web. There are major two programming methods to satisfy the requirements: Java applet of Sun Microsystems, Inc., or OpenGL plug-in method using ActiveX of Microsoft Corporation.

Java is an object-oriented programming language based on C++. It creates neutral byte codes through the compiler. Although Java enables the development of platform-independent software, the developed software requires additional software called the Java machine. Owing to the installation process of the Java machine, execution of the developed software requires booting time as well. Thanks to the development of the Java 3D technology, this yields similar performance compared to a program developed through OpenGL and C++. However, for developing the 3D configuration program including numerical computation, C++ yields better performance than Java [16]. In this paper, C++ is applied for developing the Internet-based dimensional verification system for reverse engineering processes.

Microsoft Corporation combined the Internet and the OLE technology to create Active X. The creation has enabled the use of existing OLE operations and technology on the Internet. Although the OpenGL plug-in method using the ActiveX control is per-

formed only on the Windows platform, the ActiveX is executed directly on the Internet Explorer without additional installation processes that are necessary for programs such as Java. In addition, the OpenGL plug-in method using the ActiveX is adequate to represent CAD models through abundant graphic libraries. It is better than the method using Java applet for the Web-based 3D data verification. Active X control technology is applied to process 3D graphic data in this paper.

4. System architecture

Fig. 3 shows the overall architecture of the Web-based dimensional verification system. The system consists of a digitization client, an integrated server and web clients. The digitization client registers measured data files through web pages by using the viewer and dimensional verification modules. The integrated server consists of the web, database and application servers. The web server is constructed by using the Structured Query Language (SQL) and the Active Server Page (ASP) of Microsoft Corporation. The web server notifies web pages and performs user certifications, file upload/download, file search, etc. The database server manages measurement files in connection with the SQL. The application server distributes the ActiveX control as a client program through the Internet. The program downloaded from the server is installed on a client's PC automatically. Web clients perform searching, viewing and verifying digitized files by using the distributed program when they are connected to the integrated server on the Internet or the Intranet environment. Table 1 describes hardware and software specifications for the

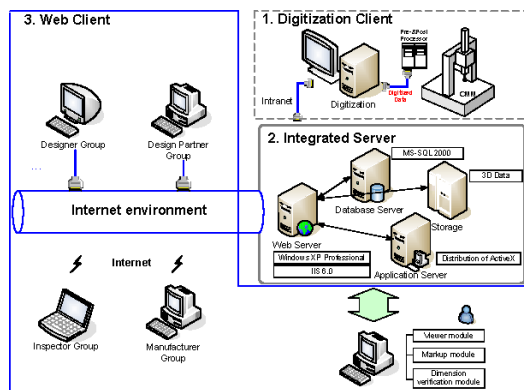


Fig. 3. Architecture of the Web-based dimensional verification system.

server system environment.

4.1 Web client architecture

Fig. 4 shows the architecture of a Web client system. The ActiveX-server architecture is applied to the dimensional verification system. This architecture has the following advantages over the client-server architecture:

1) The Active X-server architecture does not require a manual installation process. On the client-server architecture, users perform manual installation of client programs. However, ActiveX control is a client program and is installed automatically when a client is connected to the server through the Internet.

2) Updating of the dimensional verification system becomes simpler; system developers update only the server programs and the Active X controls stored in the server. It is required to take care of ActiveX controls stored in a client's PC. When a user is connected to the server, versions of the distributed ActiveX controls are checked, and the updated ActiveX controls are automatically downloaded to client computers.

3) Adopting the ActiveX-server architecture saves the construction cost of a high performance server system [17]. Although current client-server architecture experiences a falloff of network performance when multiple users are connected to one server simultaneously, the ActiveX-server architecture performs calculation on the client's PC and relieves the problem.

The distributed client software, the ActiveX control, is executed on the platforms of Windows and Internet Explorer. HTML is used to notify the ActiveX controls on the Web. VBScript is utilized for the ActiveX plug-in. The ActiveX control is realized by using OpenGL and Visual C++ over the Visual Studio 2003. OpenGL is a 3D graphic library produced by Silicon Graphics Company that is used for handling 3D point data.

Table 1. Server system environment.

Item	Specification
Hardware	
CPU	Pentium 4 3.0
RAM	2 Giga RAM
Software	
OS	Windows XP Professional
Database	MS-SQL 2000
Web server	IIS(Internet Information Services) 6.0

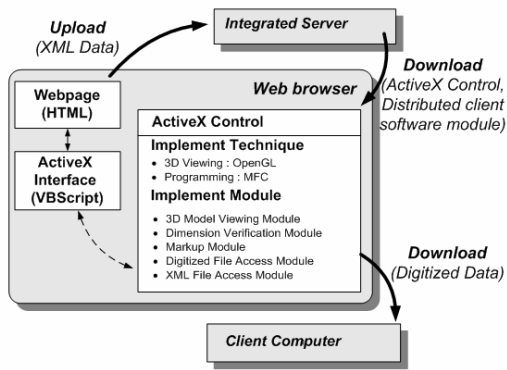


Fig. 4. Architecture of the client system.

Developed ActiveX controls consist of various functional modules: i) Digitized file accessing module reads or writes point data, ii) XML file accessing module reads or writes XML files including verification results related to the digitized file, iii) 3D model viewing module visualizes point data on the Web, iv) Dimensional verification module offers various tools for examining dimensions of the visualized point data, and v) Markup module enables users to exchange clients' opinions with other clients by using various tools.

4.2 Interface between the integrated server and Web clients

The following are the principal functions of the developed system:

- 1) The user authentication function verifies users and allows them to use the modules.
- 2) The registration function assists the registration of digitization and markup files including supplementary information through the web page.
- 3) The search function assists clients to search digitized files or XML files.
- 4) The file transmission function transmits searched files from the integrated server to web clients to make the dimensional inspection processes possible.
- 5) The dimensional inspection and markup functions assist dimensional verifiers to inspect geometric shapes and dimensions, and to express their opinions through the markup functions.

In order to register, search, upload and download digitized files or XML files, the integrated server exchanges messages with digitization clients or Web clients to transmit the file information. Message exchange is performed by using the socket communication. Fig. 5 shows the message flow between the inte-

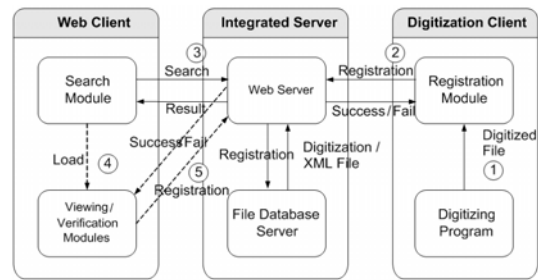


Fig. 5. Message flows between the integrated server and Web clients.

grated server and the clients as follows:

- ① When the digitization processes are completed, inspectors save the digitized point data as a digitized file and transmit the file to the registration module.
- ② The registration module transmits the file to the Web server, and it returns success or failure messages.
- ③ The search module of the Web client transmits search conditions to the Web server and then receives the Web address of digitized files that satisfy the conditions.
- ④ The viewing and Verification module receives the Web address from the search module, downloads digitized files through the HTTP protocol, and performs dimensional verification processes.
- ⑤ After the dimensional verification and markup processes, results are stored as an XML file and are uploaded to the database server to have the information shared with other clients.

5. Design methods of the dimensional verification and markup modules

5.1 User interface

Fig. 6 shows the user interface of the dimensional verification and markup modules. The screen consists of the toolbar [A], administrative tree-view [B] and graphic property toolbar [C]. The toolbar [A] is designed for selecting the dimensional verification and markup functions. The administrative tree-view [B] shows the dimensional verification results and user viewports. The user viewports contain his/her views on specific geometric shapes or dimensions. A client can have access to opinions of others by selecting user viewports. The toolbar [C] is displayed on the screen to offer more convenient dimensional verification processes.

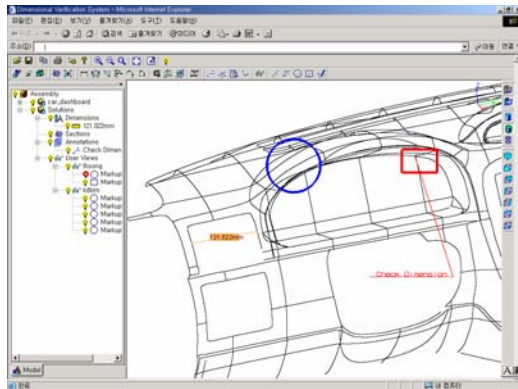
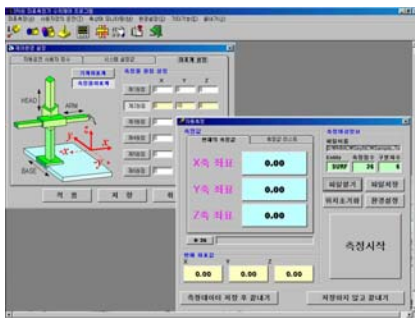


Fig. 6. User interface of the dimensional verification and markup modules.



(a) Digitization of an object



(b) Processing of digitized data

LINE	234.78	0.00	-129.73
	168.67	0.00	-59.03
/			
CIR	234.75	42.84	-133.56
	162.91	42.84	-61.08
/			
CRV	234.75	220.00	110.00
/	162.91	332.00	211.00
∴			

(c) Format of digitized data

Fig. 7. Digitization processes of the reverse engineering.

5.2 Creation method of digitized files

Most measuring machines have their own data formats that may not be used in other machines. This paper proposes a new digitized file format to complement such a drawback and to satisfy 1.2 and 1.3. The new file format recognizes all types of formats. It includes a format composed of geometric forms and three-dimensional coordinate values, even a format just composed of three-dimensional coordinate information. In general, operators of measuring machines in the styling room know the shape of objects when they digitize clay models of cars (Fig. 7(a)). In order to sort digitizing data, they put tick marks between 3 dimensional coordinate values during digitizing processes (Fig. 7(b)). They also put tick marks intentionally to distinguish geometric features during the digitizing process. Fig. 7(c) shows a sample of the digitized file. The tick marks “/” as shown in Fig. 7(c). By using the above digitizing procedures, the designer’s intention is included in the digitized data. Users inspect measurement values on a monitor by clicking a displayed geometric feature composed of digitized point data. Then the users confirm the geometric feature and verify dimensions according to the designer’s intention.

5.3 Dimensional verification functions

In order to verify dimensions, the client ActiveX reads digitized files and loads them on memories in the form of classes as shown in Fig. 8. One class is created for each geometric feature that is recorded in the digitized file. Feature_Type is an argument for classification of geometric features, such as lines, circles, curves, etc. Feature_Order represents an order of geometric features recorded in a digitized file. Feature_Order equals the order of a class. Point data belonging to each geometric feature is stored as a pointer array in the class, and each class is connected in the binary tree structure using a self-reference structure pointer. The tree structure using the self-reference structure pointer is simple and performs modification, insertion and deletion of documents easily. Since the tree structure searches data quickly and allocates memories dynamically, it is useful for handling a huge data file.

In the class shown in Fig. 8, *Prev_class is a pointer that connects the current class to the previous class. *Next_class is a pointer that connects the cur-

```

class Digitized_Data{
  Int GLID /* Identification number of the OpenGL selection*/
  Type Feature_Type; /* Type of the current geometric feature */
  Int Feature_Order; /* Order of the current geometric feature */
  Int Point_Number; /* Number of digitized points belonging to the current geometric feature */
  Point *point; /* Pointer to digitized points belonging to the current geometric feature */
  Digitized_Data *Prev_class; /* Pointer to the previous digitized data class */
  Digitized_Data *Next_class; /* Pointer to the next digitized_data class */
  ...
}
    
```

Fig. 8. Class form for saving digitized point data.

rent class to the next class. Classes created by reading a digitized file are loaded through OpenGL libraries as objects for 3D visualization and dimensional verification. OpenGL enables selection of an object by putting the GLID separating every visualized object. In this paper, Feature_Order is used as the GLID.

Selecting a point on a monitor by using a mouse, a client ActiveX examines which class is selected when the object selection function of OpenGL returns Feature_Order of the class containing the selected point data. Moreover, the returned Feature_Order is used to determine Feature_Type of the selected class and to extract 3D coordinate values of a point belonging to the class. Then, the acquired information is utilized for dimensional verification processes.

Fig. 9 shows details of the dimensional verification and markup functions of the client ActiveX. Measurement of coordinates is a function that verifies 3D coordinate values of a point on the 2D screen with a simple mouse click. Measurement of line length extracts the length of line composed of points. The point-to-point distance calculates the shortest 3D distance between two selected points.

Fig. 10 represents a pseudo code to extract a distance between two lines as follows:

1) First, a client ActiveX calculates a 3D coordinate value by inserting a selected point in the 2Dto3D_Coordinate function. The 3D coordinate value is returned and stored at the 3D_Select_Point1.

2) The 3D_Select_Point1 is inserted to the glSelectBuffer function that is an OpenGL function. The glSelectBuffer returns the Feature_Order of an object including the selected point data.

3) The acquired Feature_Order is inserted to the Least_Square_Line function. The function extracts point data belonging to a selected class and calculates a line equation through the least square method.

4) The shortest 3D distance between two lines is extracted by inputting two line equations in the distance function. Then, the extracted value is trans-

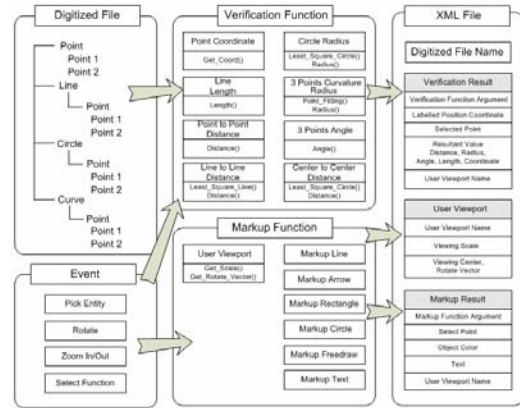


Fig. 9. Functions and file structure of the developed ActiveX.

```

3D_Select_Point1 = 2Dto3D_Coordinate(2D_Selected_point1);
3D_Select_Point2 = 2Dto3D_Coordinate(2D_Selected_point2);

If (DimensionType == LNL_Distance && MousePickNumber == 2)
{
  Feature_Order1=glSelectBuffer(3D_Select_Point1);
  Feature_Order2=glSelectBuffer(3D_Select_Point2);
  Line_Equation1=Least_square_Line(Feature_Order1);
  Line_Equation2=Least_square_Line(Feature_Order2);
  ResultDim=Distance(Digitized_Line_Equation1, Digitized_Line_Equation2)
  Display_Dimension(ResultDim);
  Initial_Picking_Order();
}
    
```

Fig. 10. Pseudo code to extract distance between two lines.

ferred to the ResultDim.

5) The shortest 3D distance between two lines is displayed on the monitor by using the Display_Dimension function.

Whenever users move a cursor around point data, geometric features including the point data are searched and highlighted on the monitor. A geometric feature including the point selected by using a mouse is extracted.

Radius of a circle is computed from a least square fitting of points belonging to the selected circle. An angle and a curvature radius composed of three points are calculated from the three points selected by a user. A distance between two circles calculates the shortest 3D distance between two circle centers. The least square method is also used for calculating the distance between the interpolated circles by using point data that belong to the two selected circles.

5.4 Markup functions

The markup function enables distributed clients to

collaborate with each other by expressing and sharing their opinions on visualized geometric features. As shown in Fig. 9, clients are able to select the markup types such as a line, an arrow, a rectangle and so on. They leave markups on a visualized feature and express their thoughts in writing. In addition, a viewport of a client, which is represented by a viewing scale, viewing center and rotational vector mathematically, is stored together with dimensional verification and markup results through the user viewport function. Other clients are not only able to understand another client’s opinion, but are also able to look at the viewport when they select the user viewport name registered on the administrative tree-view. The selected viewport screen shows the markup and dimensional verification results from the previous verifiers. Using the markup function, collaborations between designers, manufacturers and buyers are available on the Internet.

Fig. 11 shows an XML file including the digitized file name, dimensional verification results, user viewports and markup results. If dimensional verification results are saved together with the point data, the file size will be very large. Clients should download the same point data iteratively in order to examine dimensional verification results of other clients. However, since the dimensional verification and markup results are saved as XML files and not as the digitized files mentioned in this paper, the iterative downloading processes of a large-size data file are eliminated.

Verification results, composed of the labeled position coordinates, selected points, resultant values and user viewport name, have different information according to verification functions such as coordinates,

line length, point-to-point distance, etc. Therefore, the verification results are classified and saved according to the verification function argument that identifies verification functions. A label, including a result value, is generated when the verification process is finished. The label position coordinate is a coordinate value shown on a monitor.

Markup results are classified according to the markup function argument identifying markup types. Each result contains the selection of the point, color, text and user viewport name.

If new dimensional verification or markup functions are required, the developed system is updated by assigning verification and markup function arguments and by implementing the added functions [18].

6. Case studies

The clay model of the dashboard shown in Fig. 12 is measured by using a layout machine shown in Fig. 1. The efficiency of the web-based dimensional verification system for reverse engineering is investigated through the application of the measured data. After connecting to a web page and confirming a user authentication, users are able to start the dimensional verification processes. Users search for specific digitized files by inputting an inspector name or the name of measuring data, which is performed through the search window. If a user selects a digitized file from the search window, the selected file is transferred to the user’s PC. Then, the ActiveX control reads the digitized files and shows their geometric features to the user.

The length of the ventilating element on the dashboard is examined through the developed system.

```

<?xml version="1.0" encoding="UTF-8" ?>
<BaseDocument topNode="Drawing">
  <UserViewport>
    <UserViewport_Name>insong</UserViewport_Name>
    <Viewing_Scale>106.736</Viewing_Scale>
    <Viewing_Cen><Vector x="1.776" y="1.065" z="14.689" /> </Viewing_Cen >
    <Rotate>< Vector x="153.948" y="104.914" z="148.414" /> </Rotate>
  </UserViewport>
  <Verification_Result>
    <Verification_Function_Argument>3</Verification_Function_Argument>
    <Style styles="Color:255,255,255; Show:1" /></Style>
    <Label_Position><Point3d x="1165.620" y="749.439" z="1493.762" /></Label_Position >
    <Select_Point><Point3d x="840.145" y="745.807" z="1472.750" /></Select_Point>
    <Select_Point><Point3d x="1491.096" y="753.071" z="1494.773" /></Select_Point>
    <Point_to_Point_Distance unit="mm">651.539</Point_to_Point_Distance>
    <UserViewport_Name>insong</UserViewport_Name>
  </Verification_Result>
  <Markup_Result>
    <Markup_Function_Arguments>3</Markup_Function_Arguments>
    <Select_Point><Point3d x="43.784" y="21.401" z="54.300" /></Select_Point>
    <Select_Point><Point3d x="23.362" y="25.094" z="65.956" /></Select_Point>
    <Style styles="Color:255,0,0; Show:1" /></Style>
    <Text>None</Text>
    <UserViewport_Name>insong</UserViewport_Name>
  </Markup_Result>
  </BaseDocument>
  
```

Fig. 11. XML file format for saving dimensional verification and markup results.

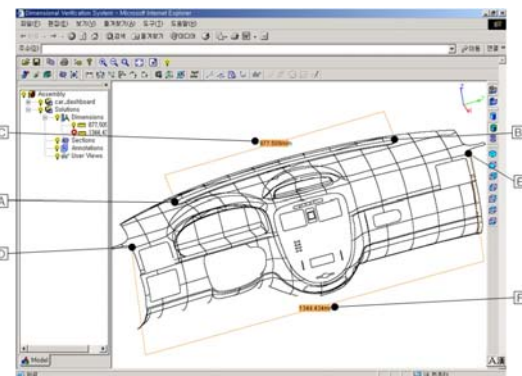


Fig. 12. Case study: Dashboard of a car.

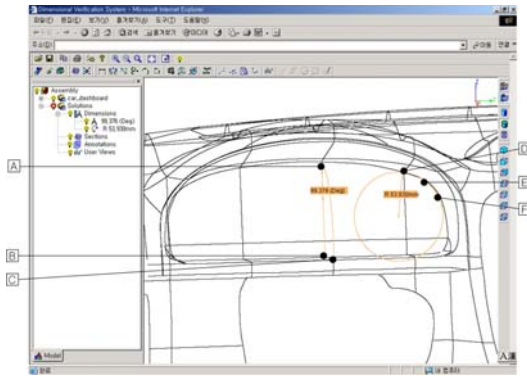


Fig. 13. Case study: Dimensional verification for an instrument panel.

After the icon to calculate the distance between two points, point **A** and point **B** as shown in Fig. 12 is selected, the length of the ventilating element is calculated as 877.509mm and is displayed on the monitor as shown in **C** of Fig. 12. The outer length of the dashboard is examined to be 1344.434mm by selecting points **D** and **E**. By using functions to calculate the distance between two points or two lines, a user is able to inspect whether the dashboard has been installed on a car frame correctly.

The viewing angle of a car instrument panel, as shown in Fig. 13, is important for human engineering. This design factor can be examined by using the function of three-point angles. After selecting the icons for three-point angles, starting point **A**, mid point **B** and end point **C**, the angle among the three points is measured as 99.376° . The calculated angle may be compared with design specifications or examined to verify whether it truly is the comfortable angle for car drivers.

In order to verify the outward shape of the instrument panel, users may use the function of three-point curvature radius. After the icon for the three-point curvature radius is selected, the curvature radius among the selected points **D**, **E** and **F** is measured as shown in Fig. 13. Users are able to confirm the curvature radius as 53.930mm and compare the result with the design specifications.

User viewports are managed by user viewport names listed in the administrative tree-view, as shown in **A** of Fig. 14. If a user selects a user viewport name in the tree structure, the current screen changes to the specific screen containing markup and dimensional verification results of the previous users. **B** of Fig. 14 shows a dimension value and markups repre-

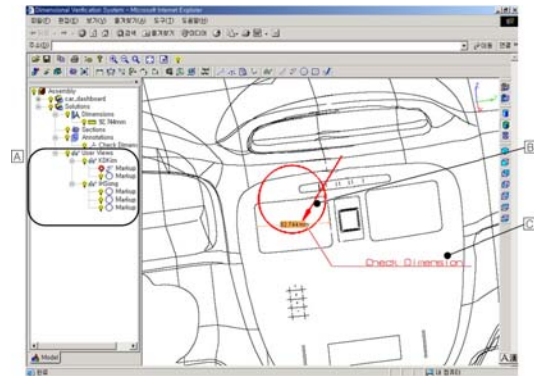


Fig. 14. Case study: User viewport and markup operations for a centerfacia.

sented as a circle and an arrow. Opinions of the previous user also appear as a text shown in **C** of Fig. 13. Since the text is designed to support a hyperlink function, it is to be linked with a document, drawing, or a picture. Using the hyperlink function, users are able to explain their opinions in detail.

The creation method of a markup is as follows: when a user selects the icon on the user viewport and inputs a name, a user viewport is created and registered in the administrative tree-view as **A** of Fig. 14. After this step, all markup operations are added to the created user viewport as a file substructure.

7. Conclusions

(1) The paper proposes a web-based collaborative system that examines geometric shapes of measured point data, inspects dimension information and exchanges views with other clients in the reverse engineering process. As the developed system handles measured point data directly, a conversion process of measured point data into a CAD file or VRML form is not required to inspect and verify dimensions.

(2) The functional requirements are defined to construct the web-based dimensional verification system, and the current programming technologies have been investigated to select more adequate methods to implement the functional requirements.

(3) As the ActiveX-server architecture is applied to the developed system, installation of the developed system is conducted automatically through the Internet. Automatic updating of the ActiveX reduces the maintenance and upgrade cost of the system on the distributed environment. It is better than the conventional client-server architecture.

(4) The digitized file format is devised to analyze geometric features as well as three-dimensional coordinate values of point data. The viewing module of the developed system is able to visualize point data and geometric features on the Web by using OpenGL and ActiveX technologies. Whenever users move a cursor around point data, the developed system searches and highlights geometric features around the point data. Design reviewers can confirm geometric features and verify dimensions according to the designer's intention.

(5) The dimensional verification and markup functions are designed to inspect dimensional information, mark-up an important area, and make a statement of the design reviewer's opinion through the Web. And the functions are classified according to the verification and markup function arguments, respectively. If new dimensional verification or markup functions are required, the developed system is simply updated by assigning arguments and implementing required functions. As the confirmation of the verification results is conducted through XML files, iterative downloading of a large-size data file is eliminated and it is quick.

(6) The validity and effectiveness of the developed system have been verified through case studies. Various parts of a dashboard are examined to determine whether geometric shapes and dimensions coincide with design specifications or human factors.

References

- [1] C. H. Chu, C. Y. Cheng and C. W. Wu, Applications of the Web-based collaborative visualization in distributed product development, *Computers in Industry*, 57 (2006) 272-282.
- [2] S. H. Ahn, et al., Design consultant: a network-based concurrent design environment, Proceedings of ASME IMEC & E, USA. (1999) 23-30.
- [3] G. Q. Huang, Web-based support for collaborative product design review, *Computers in Industry*, 48 (2002) 71-88.
- [4] S. H. Ahn, et al., CyberCut: an Internet-based CAD/CAM system, *Transactions of ASME*, 1 (4) (2001) 52-59.
- [5] J. M. Mark, et al., PACT: an experiment in integrated concurrent engineering systems, *IEEE Computer*, 26 (1993) 28-37.
- [6] A. Pang and C. Wittenbrink, Collaborative 3D visualization with Cspray, *IEEE Computer Graphics Application*, 17 (1997) 32-41.
- [7] O. O'Grady, et al., Distributed concurrent engineering: Internet-based interactive 3-D dynamic browsing and markup of STEP data, *Concurrent Engineering Research and Applications*, 6 (1998) 53-70.
- [8] International Organization Standardization, ISO/DIS 10303-219, Industrial Automation Systems and Integration-Product Data Representation and Exchange-Part 219: Application Protocol: Dimensional Inspection, (2004).
- [9] DMIS, Dimensional Measuring Interface Standard, ANSI/CAM-I 101, version 2.1, Computer Aided Manufacturing - International Inc., Arlington, USA, (1990).
- [10] Dimensional Markup Language, Available online at: www.dmlspec.org (accessed 30-4-2006).
- [11] Cimmetry Inc., Available online at: www.cimmetry.com (accessed 30-9-2006).
- [12] Actify Inc. Available online at: www.actify.com (accessed 30-9-2006).
- [13] Techsoft Inc., Available online at: www.techsoft3d.com (accessed 30-9-2006).
- [14] INUS Inc., Available online at: www.inustech.com (accessed 30-9-2006).
- [15] McNeel Inc., Available online at: www.rhino3d.com (accessed 30-4-2006).
- [16] J. E. Moreira, S. P. Midkiff and M. Gupta, A comparison for Java, C/C++, and FORTRAN for numerical computing, *IEEE Antennas and Propagation Magazine*, 40 (5) (1998) 102-105.
- [17] I. H. Song and S. C. Chung, Web-based precision dimensional verification system for rapid design and manufacture, Proc. of American Society for Precision Engineering, Portland, Oregon, USA. (2003) 359-362.
- [18] I. H. Song, K. D. Kim and S. C. Chung, An Internet-based dimensional verification system for reverse engineering, *Transactions of the KSME (A)*, 27 (2004) 1409-1417.