Int J Adv Manuf Technol (2006) 28: 101–108
DOI 10.1007/s00170-004-2422-y

ORIGINAL ARTICLE

Yoonho Seo · Dae-Young Kim · Suk-Hwan Suh

Development of Web-based CAM system

Received: 7 July 2004 / Accepted: 20 September 2004 / Published online: 4 May 2005

Abstract In this paper, we present the methods to develop a Web-based CAM system that is capable to dynamically simulate machining processes in solid graphics form through the Internet. Methods include the virtual reality modeling language (VRML) structure to represent the machined workpieces progressively on Web, the definition and storage of workpiece information upon the request of users, the CORBA-based interpretation of NC part programs, and algorithms for efficient machining simulation. Based on these methods, an Internet-based CAM system is implemented at http://isd.korea.ac.kr. Using the Web-based CAM system, public users can simulate machining processes by for their own NC part programs.

Keywords CAD/CAM · Machining process · Virtual reality modeling language (VRML) · Web-based

1 Introduction

Virtual reality technology, which has been applied to manufacturing engineering since early 1990s, has reached the stage where virtual manufacturing system is implemented for industrial practice and applied to virtual enterprise realization [1]. Utilization of the virtual reality technology will improve the product development process and cost by combining constraints of production function, manufacturing, logistics, and many other life-cycle issues.

Specifically, with the latest advancements in Internet communication technology, virtual manufacturing system technology that is based on the network has received increased attention as a means to increase the responsiveness of manufacturing systems. In particular, technical advancement in data handling, synchronization, and animation through the Internet, along with ultra-speed ISDN, has motivated the development of Web-based software tools [2, 3].

In this paper, we develop a Web-based CAM system (WCAM). The concept of an Internet-based CAM system has been developed as a subsystem of a Web-based virtual machine tools system (WVMT) [2]. But, due to the technical hurdles such as the great amount of data communication and processing on the Internet, the WVMT system could not display the solid graphics of machining processes in real-time; it could only display tool paths or finalized machined shape. The current system is attempts to break the representational limit of dynamic 3D graphic simulation on Internet.

Recently, the modeling tools for educational purpose and simulation [4, 5], the implementation of network-based virtual machine tools [2], and the remote control and monitoring of Internet-based virtual machining systems [3, 6] have been developed using Web-based visualization tools such as virtual reality modeling language (VRML) or Java3D [7]. As far as CAD/CAM technology is concerned, numerous systems have been developed, and many are commercially available [8, 9]. Even though software systems to incorporate the motion of machine tools into the CAD/CAM platform have been developed and are commercially available, these are basically off-line software running in a stand-alone fashion, not on the Internet [10–12].

This paper discusses the implementation methodologies associated with the development of a Web-based CAM system. Specifically, we consider the representation method necessary to visualize dynamic machining process on the Web, the VRML file structures to contain the workpiece information as specified by public users, and a CORBA-based NC part program interpretation method and machining simulation algorithm. Also, the development of a prototype Web-based CAM system is demonstrated.

Y. Seo (✉)
Intelligent Systems Design Lab,
Dept of Industrial Systems and Information Engineering,
Korea University, Seoul, South Korea (135-713)
E-mail: yoonhoseo@korea.ac.kr
Tel.: +82-2-32903393
Fax: +82-2-9295888

D.-Y. Kim
Duzon DASS Corp.,
Ire Bldg. 2, Yangpyungdong 4-ga, Yongdungpogu, Seoul, South Korea

S.-H. Suh
National Research Lab. for STEP-NC Technology,
POSTECH, Pohang, South Korea

2 Design of Web-based CAM system

Conceptually, WCAM technology basically involves bringing off-line CAM functions into the Internet environment. In WCAM, users on the Internet can simulate machining processes on the workpiece by user-specified NC part programs. There may be two extreme cases to realize the WCAM concept: one is to program a client's JAVA applet to be equipped with all the functions of CAM software; the other side is to minimize the client's role, in which client's computer only transfers the user's input to the CAM server and displays the machined results. However, both cases are infeasible for WCAM implementation. The size of the JAVA applet quickly becomes too large to access easily for the first case; while the second case causes an excessive amount of communication requirements between client and server because of the real-time and graphic characteristics.

In this paper, the data processing load required in a Web-base CAM system is distributed between client, server and interpreter server by way of a middleware CORBA, as shown in Fig. 1. The applet in client side is composed of a workpiece generator to create and store information on workpiece through user input, and a simulator to calculate 3D machined shapes and to dynamically display it on the workpiece in real-time. Meanwhile, the interpreter server generates the dynamic information on the cutting process.

In order to realize the WCAM concept, it is required to develop (1) a dynamic representation of machined processes on Web, (2) a creation of workpiece VRML file by user specification, (3) an interpretation of a NC part program through CORBA, and (4) an efficient machining simulation algorithm.

First, existing CAM systems mainly use cell decomposition methods, such as Z-map, M-map, Octree and Voxel, for the solid representation of a workpiece. This is because the calculating method and time of cutting geometry are relatively simple and fast [8, 9, 13]. Especially in the case of the Z-map method, which is widely used to visualize three-axis machining processes because of its concise size, easy implementation and fast calculations, its main weakness is the fact that it cannot express the machined shape in sidewall. In this paper, a Z-map-based representation method of workpieces is presented to dynamically visualize the cutting process using VRML on the Internet.

Second, in order to minimize the communication load between the client and server, all tasks to create and possess information on the workpiece and to display 3D machined processes in real-time are performed in the client applet. To make it feasible, a VRML file is created by client applet should have the structure that can contains the information describing the shape and mesh of workpieces.

Third, to analyze an NC part program input by user, the Web-based CAM system selects a way to use an existent CAM system through CORBA, rather than developing a brand-new interpreter in the JAVA applet [14].

And finally, a machining simulation algorithm applies a method to narrow the searching dimension by calculating the bounding box where a tool's moving area conflicts with the workpiece.

3 Development of Web-based CAM system

3.1 VRML-type Z-map representation

The ElevationGrid node of VRML's geometry nodes, which is composed of rectangular forms with XZ space and variable height on Y-axis, is similar to the Z-map structure. But the ElevationGrid node is not appropriate to express the machining process dynamically because it cannot change in height by external authoring interface (EAI) communication.

As a result, in this paper, the IndexedFaceSet geometry node is used for Internet-based dynamic representation of machining simulation for the following reasons: (1) a surface can be constructed with a combination of vertices inside an area, and (2) the height (Z-axis value) of the surface can be changed dynamically through EAI communication.

To represent the workpiece with Z-map structure using VRML's IndexedFaceSet node, a VRML-type Z-map structure is proposed. For VRML-type Z-map representation, a workpiece is decomposed into small meshes which are formed by combining four adjacent points defined by IndexedFaceSet. Examples are the meshes (0, 1, 6, 5), (1, 2, 7, 6), (2, 3, 8, 7), etc., shown

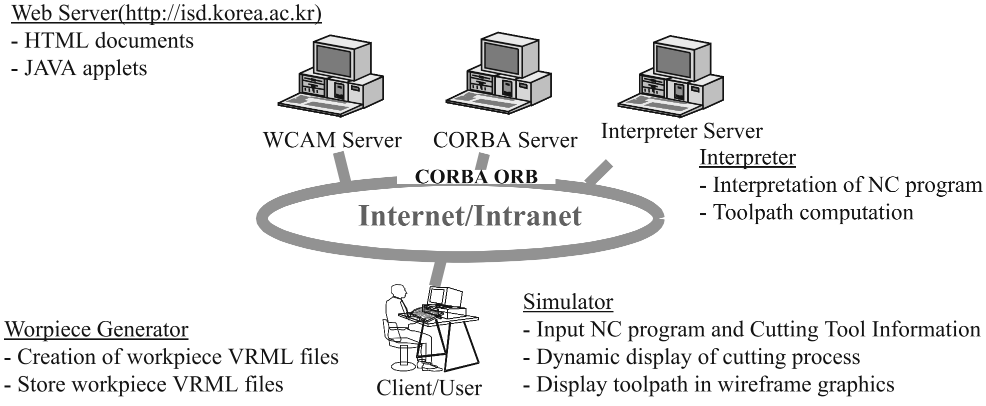


Fig. 1. Conceptual diagram of Web-based CAM system

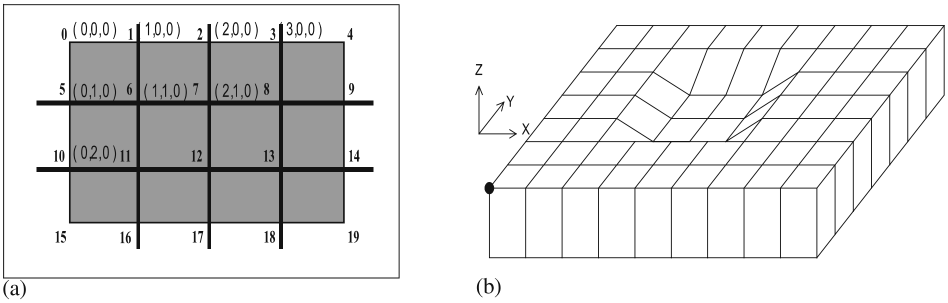


Fig. 2. a Mesh structure and **b** Machined shape of VRML-type Z-map

in Fig. 2a that are formed by combining the four adjacent points. By changing the Z values, the machined shape can be expressed as per Fig. 2b. But, as shown in Fig. 2b, VRML-type Z-map representation has a shortcoming in the fact that sidewalls between the meshes with different heights increase since height change in a mesh has an influence on neighboring meshes.

To resolve the inclined sidewall problem, a side-attached Z-map method that adds "dummy" meshes for representing the sidewall is proposed. For example, see Fig. 3a, where a dummy mesh (1, 2, 10, 9) is formed to express the sidewall between meshes (0, 1, 9, 8) and (2, 3, 11, 10). Also, dummy meshes (8, 9, 17, 16), (9, 10, 18, 17), (10, 11, 19, 18), etc., are used for sidewall representation, while meshes such as (0, 1, 9, 8), (2, 3, 11, 10), (16, 17, 25, 24), etc., are used to express the machined surface. In a side-attached Z-map representation, since the size of dummy mesh for a sidewall is zero, it is possible to express a 90-degree sidewall between adjacent meshes. However, the VRML file size in side-attached Z-map representation is four times as big as that of a VRML-type Z-map, since four coordinate values are set to one point for side-attached Z-map representation.

In case of workpiece and mesh space, the file size and simulation time between a side-attached Z-map and a VRML-type Z-map are compared, and the results are listed in Table 1. As shown in Table 1, the file size of the side-attached Z-map method is about four times larger than those of the VRML-type Z-map method. The machining simulation speed of the side-attached Z-map method decelerates exponentially as file size increases. This is because the time is influenced by the transmission speed of a file, the visualization speed of the VRML browser, the EAI communication speed of coordinate changes, etc.

Table 1. Comparison between VRML-type and side-attached Z-map

Workpiece size	VRML-type Z-map		Side-attached Z-map	
width×length×height	size (kB)	time (s)	size (kB)	time (s)
100 × 100 × 50	318	3	1376	15
200 × 200 × 50	1398	15	6003	153
300 × 300 × 50	3200	58	14143	1597

In workpiece machining simulation, it is usual to express a precisely machined shape with a high resolution. And in the case where a small mesh is used, the inclined sidewall phenomenon of the VRML-type Z-map can be diminished. Therefore, in spite of the inclined sidewall phenomenon, VRML-type Z-map representation is used in this research to visualize dynamic machining simulation in consideration of today's Internet technology. But it must be acknowledged that if the visualization speed of VRML browsers improve, then the side-attached Z-map method may become more a useful representation method for machining simulation.

3.2 Generation of workpiece VRML File

To accommodate various workpiece shapes and mesh resolutions specified by a client and to efficiently express the workpiece in a client applet, the procedure to generate and store client-specified information is as follows: (1) the generation of a workpiece VRML file with dimensions and mesh resolutions specified by the client, (2) the file is stored in the client's computer, and (3) the simulator uses the workpiece information stored in the client's hard disk, as shown in Fig. 4.

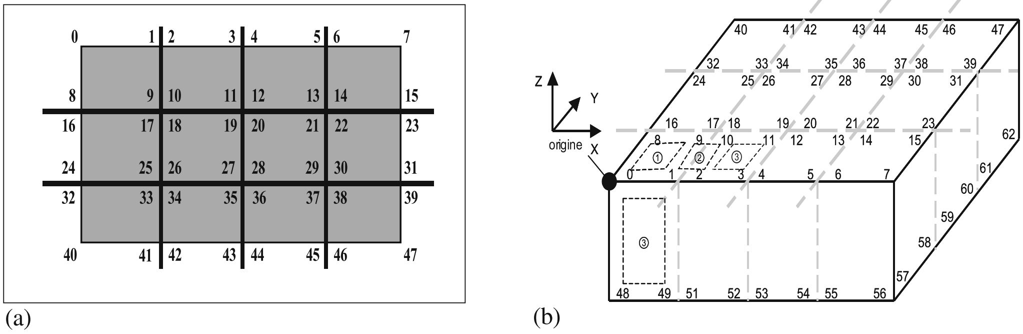


Fig. 3. a Mesh structure and **b** Workpiece representation by side-attached Z-map

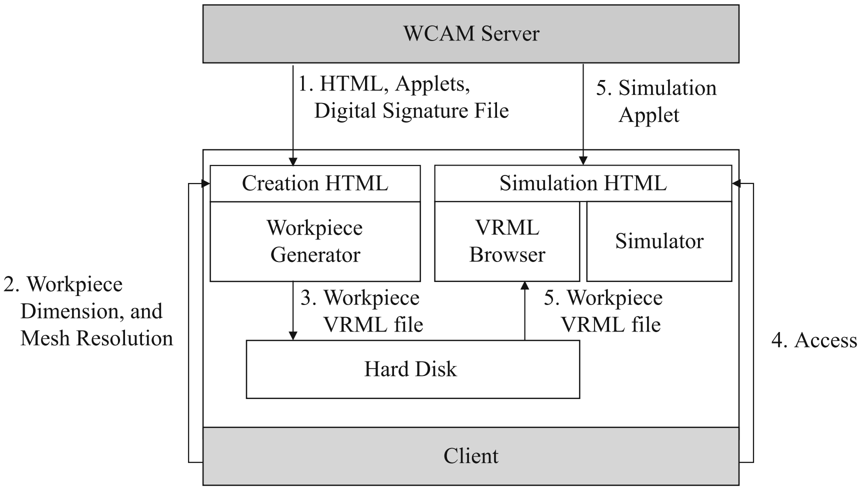


Fig. 4. Structure of workpiece generation

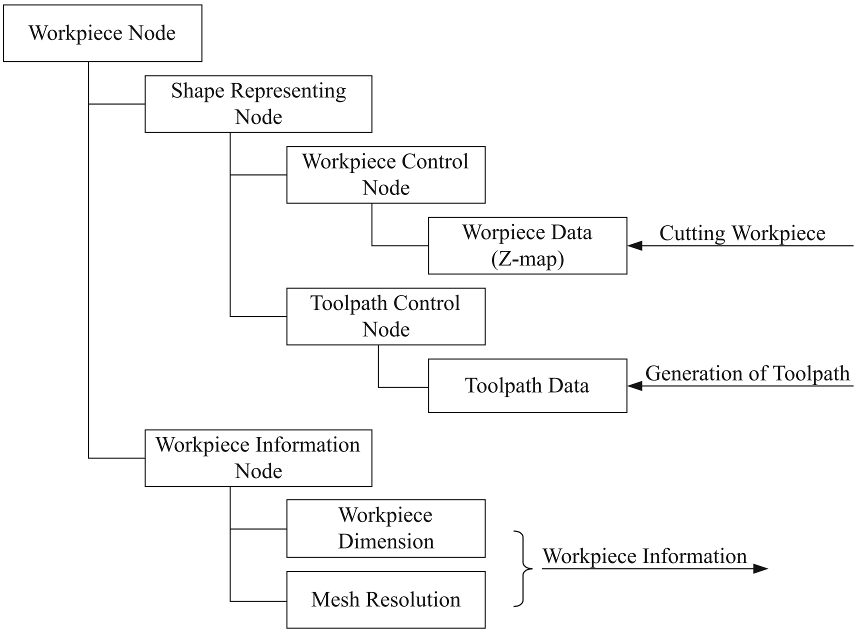


Fig. 5. VRML tree structure for workpiece information

The workpiece VRML file should contain information describing dimensions and mesh resolutions, a workpiece's representation in the VRML-type Z-map format, and tool path. Also, a digital signature tool is used to store the workpiece VRML file on the client's computer [15, 16]. The workpiece VRML file presents information in a tree structure, as shown in Fig. 5. This workpiece information is passed to the simulator to generate the cutting trace and tool path.

3.3 CORBA-based NC code interpretation

To interpret the user-specified NC code, the Web-based CAM system uses an existing CAM system rather than developing a new NC code interpreter in the JAVA applet. As shown in Fig. 6, the ,machining simulator on client side sends the user's

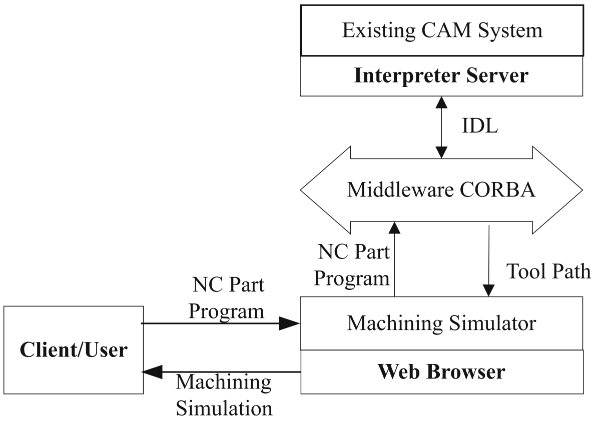
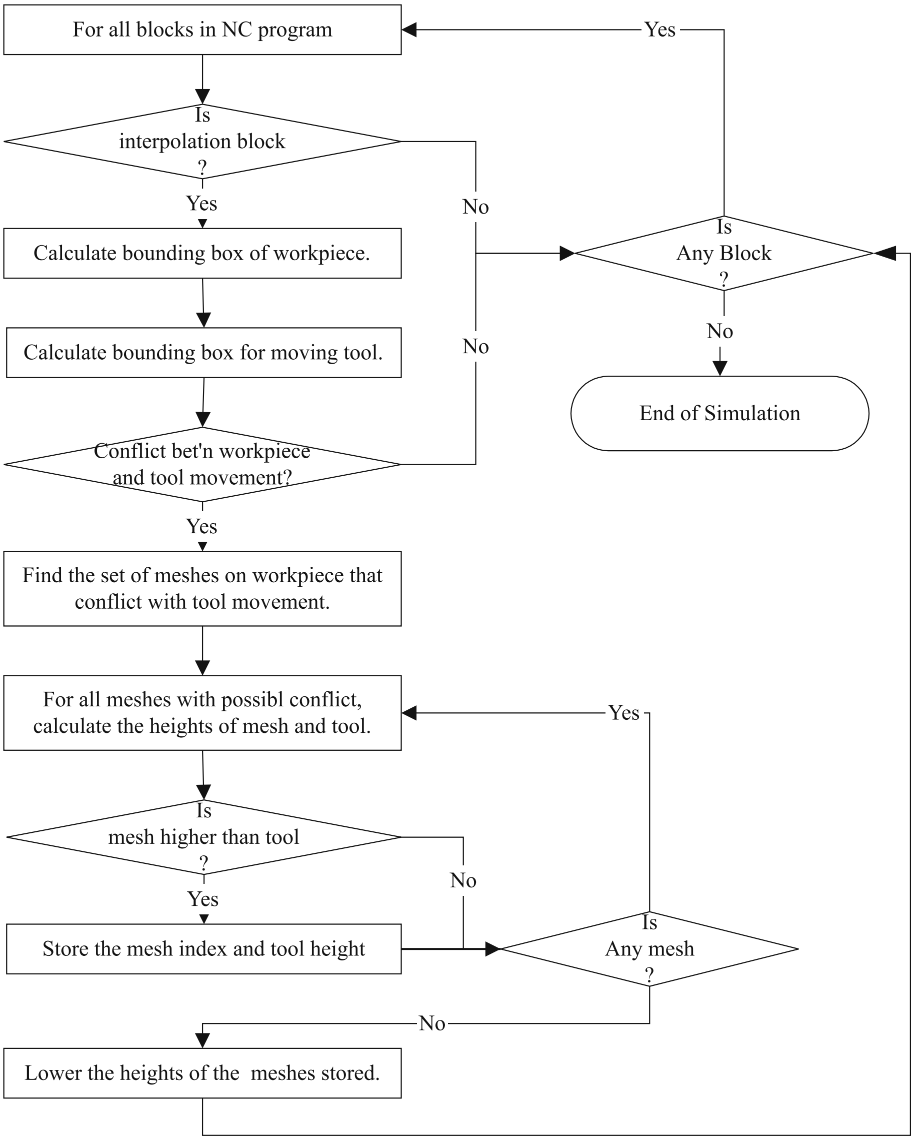


Fig. 6. NC code interpretation through CORBA

Fig. 7. Machining simulation algorithm



NC part program and receives tool path information from an existing CAM system through CORBA.

For example, in this implementation, PosCAM, a CAD/CAM system developed by Suh is used [17]. The PosCAM interprets the user's NC part program and sends the results, including a series of segmented tool paths indicating the starting and ending points. Then, the machining simulator displays the tool path on the workpieces.

CORBA-based interpretation to obtain tool path information has the following advantages: (1) the system can be constructed easily, 2) it can reduce the heavy communication load between client and server which could occur when an interpreter in an Applet is developed, (3) both efficiency in system development and reliability in code interpretation are obtained because an existing verified CAM system is used, and (4) the future extension and maintenance will be easy because of the interdependency of the interpreter.

3.4 Machining simulation algorithm

The machining simulation algorithm is carried out by (1) searching the machining area on a workpiece by calculating bounding boxes in which tool and workpiece conflicts occur, and (2) changing the height of meshes at the machining area, which is composed of Indexed FaceSet geometry nodes [18]. A detailed algorithm is depicted as a flowchart in Fig. 7.

4 Implementation of the Internet-based CAM system

The WCAM proposed in this paper is constructed using a client/server structure, as shown in Fig. 8. The WCAM server contains the various HTML documents, JAVA applets, and digital signature files required for clients. As depicted in Fig. 8, the

Fig. 8. Architecture of Internet-based CAM system

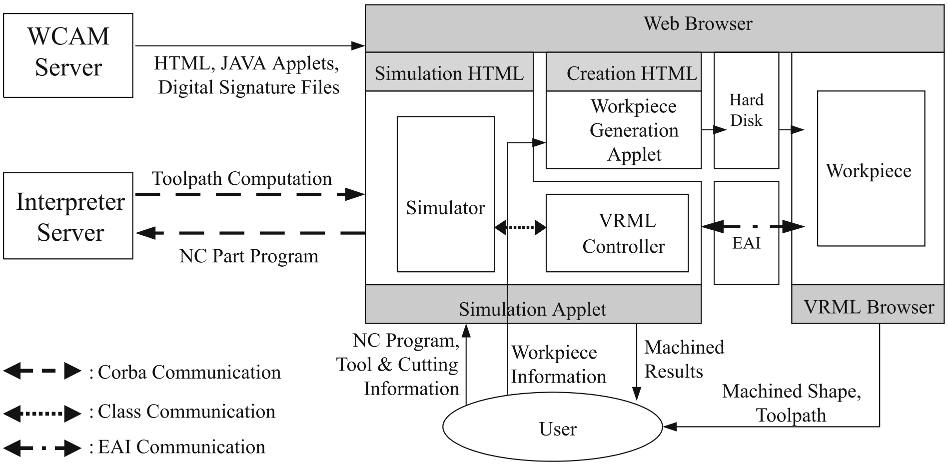
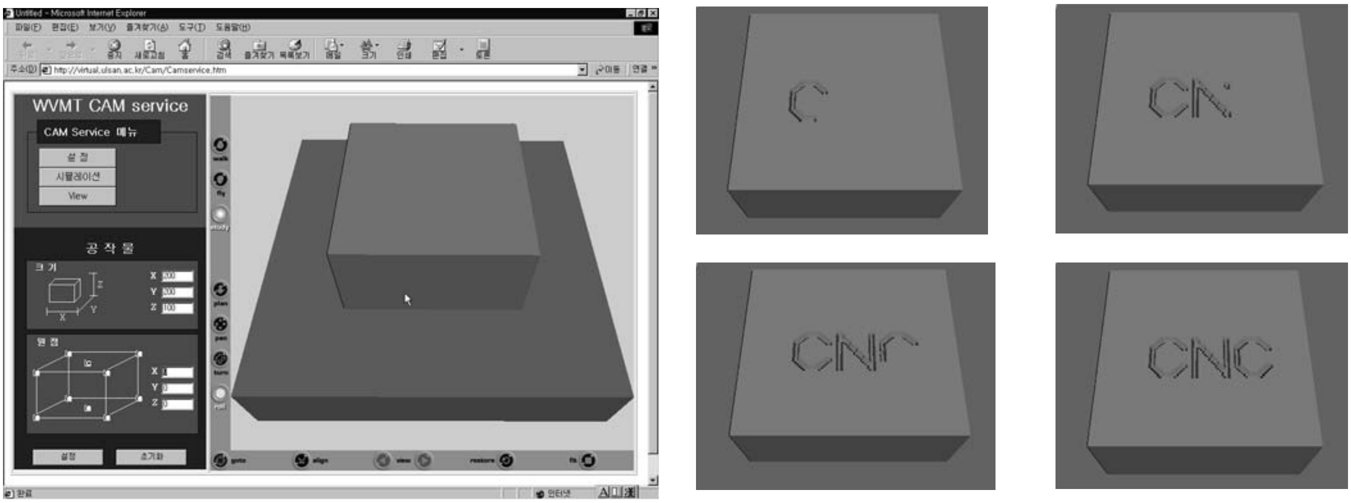


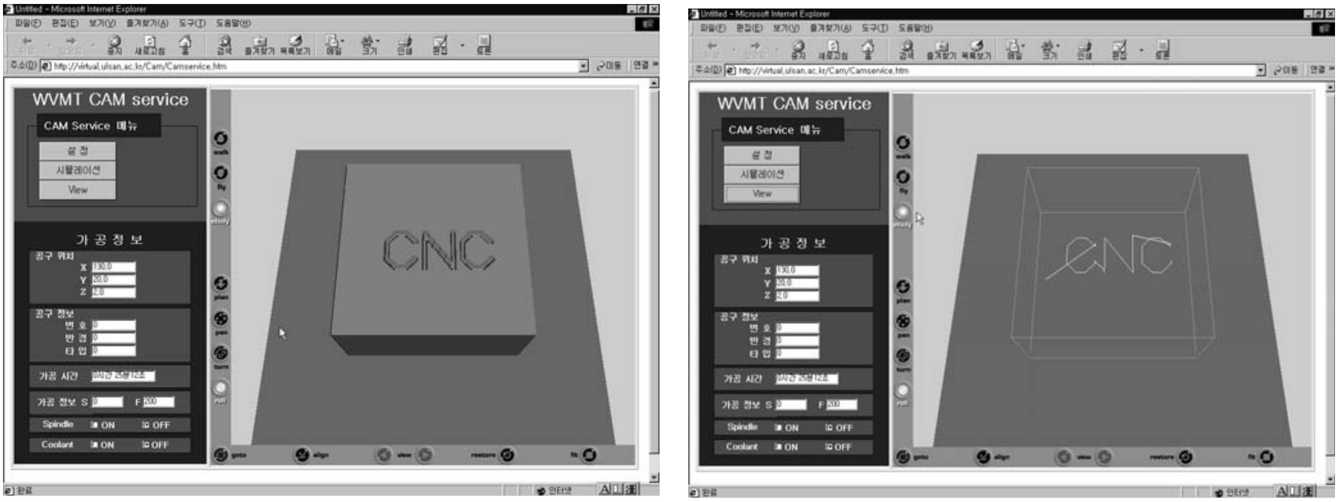
Fig. 9. Sample IDL program

```
//WVMT.idl
module WVMT
{
  //
  typedef sequence<writing> StringSeq;

  // structure for storing the result of Interpreter
  struct IPOutput
  {
    short numBlock;      // block number of interpreted NC codes
    short gCode;         // type of interpolation, 0: G00, 1: G01, 2: G02
    float startPos[3];      // starting point (mm)
    float endPos[3];       // ending point (mm)
    float circleCenter[3];   // center coordinate of circular interpolation (mm)
    float circleRadius;     // radius of circular interpolation (mm)
    float feed;            // feed rate (mm/min)
  };

  // structure sequence for storing the result of Interpreter
  typedef sequence<IPOutput> IPOutputSeq;
  struct PLCOutput
  {
    short numBlock;      // block number
    char PLCCode;       // type of PLC code
    short mCode;         // M: M-code
    short sCode;          // S: S-code
    short tCode;          // T: T-code
  };

  struct ToolInfo
  {
    float toolRadius;
    float toolRWear;
  };
  typedef sequence<ToolInfo> ToolInfoSeq;

  interface Interpreter    // interface for Interpreter Class
  {
    // method returning the results of interpretation of the given codes
    IPOutSeq interpret (inout StringSeq inoutCodes, in ToolInfoSeq inToolInfo,
           out PLCoutputSeq outPLCOutputSeq, out StringSeq outMessages);
  };
```



(a) (b)

Fig. 10. a Initial screen and **b** Progressive cutting processes



(a) (b)

Fig. 11. a Machined result and **b** Path simulation

proposed Web-based CAM system performs the graphic simulation of a dynamic machining process through the following procedure: once a user accesses the Web server, the Web-based CAM system takes workpiece dimensions and mesh resolution as inputs, automatically generates the workpiece VRML file and stores it in client's hard disk. Then, with after accepting user inputs such as part program and cutting tool information, the simulator sends this information to the interpreter server and receives tool path information through CORBA. To make it possible to exchange information between the interpreter and simulator, an interface in an interface definition language (IDL) should be defined in CORBA, as shown in Fig. 9. Using tool path information, the simulator finds the machining area and sends the instructions to the VRML browser to visualize the cutting shape.

For the system implementation, Internet Information Server 4.0 (IIS4.0) is used for the Web server, while the interpreter server is programmed with VC++6.0, and wrapped up with Visibroker 3.3 for C++ and Visibroker 3.4 for Java for CORBA. The digital signature file uses SdkJava40 for IE, while the client must have Cortona 3.0 and a Web browser.

During simulation, information on various statuses can be provided, such as tool position, tool information, machining time, etc., An initial screen display in which workpiece information is specified is illustrated in Fig. 10a, while progressive cutting shapes are shown in Fig. 10b.

Figure 11a illustrates the result of a cutting simulation in solid graphics, while Fig. 11b shows a path simulation in wire graphics. Also, a 2D cross-sectional view of the cutting shape can be provided.

5 Conclusions

In this paper, Web-based CAM technology is presented, including a VRML-type Z-map structure for the dynamic expression of cutting process on Web, a VRML tree structure to contain workpiece information in client side, a CORBA-based NC code interpretation, and a machine simulation algorithm. Based on these methods, a prototype Web-based CAM system is developed and is available on-line at http://isd.korea.ac.kr.

While existing CAD/CAM systems are of the stand-alone, off-line types, our system is based on Internet. Therefore, public users can use this system and CAM technology on the Internet. Also, it is useful as a means of education, since multiple copies of expensive CAM/CAM software are not necessary. Furthermore, the WCAM technology can be used for a Web-based, on-line monitoring and control system for machine tools, a process that is going to be developed.

Acknowledgement This research was in part supported by a Korea University grant (R01-2002-000-00232-0) and by a grant from the Basic Research Program of the Korea Science and Engineering Foundation.

References

1. Mehler BH, Reinhart G (1998) Building the virtual enterprise-manufacturing in decentralized networks. In: Proceedings of the 31st CIRP International Seminar on Manufacturing Systems, 26–29 May 1998
2. Suh SH, Seo YH, Lee SM, Choi TH, Jeong GS, Kim DY (2003) Modeling and implementation of Internet-based virtual machine tools. Int J Adv Manuf Technol 21:516–522
3. Lee KI, Noh SD (1997) Virtual manufacturing system: a test bed of engineering activities. CIRP Ann 46:347–350
4. Kim KC, Park JY (2000) Development of Web-based operating simulator for virtual lathe system. In: Proceedings of the 2000 Fall Conference, The Korean Institute of Industrial Engineers, Section A04–1
5. Rohrmeier M (2000) Web based robot simulation using VRML. In: Proceedings of the Simulation Conference 2:1525–1528
6. Jung YS (1999) A study on the Internet-based monitoring and teleoperation of a machining center. Dissertation, Seoul National University
7. Marrin C (1997) Proposal for a VRML 2.0 informative annex, external authoring interface reference. Silicon Graphics. Accessed in February 2000: http://www.graphcomp.com/info/specs/eai.html
8. Hook TV (1986) Real-time shaded NC milling display. In: Proceedings of SIGGRAPH in Computer Graphics 20(4):15–20
9. Hsu PL, Yang WT (1993) Real-time 3D simulation of 3-axis milling using isometric projection. Comput Aided Des 25(4):215–224
10. Delmia Virtual NC. Accessed in February 2000: http://www.delmia.com/gallery/pdf/DELMIA_VirtualNC.pdf
11. Flow software technology Workspace5. Accessed in February 2000: http://www.workspace4.com
12. Technomatix ROBCAD. Accessed in February 2000: http://www.tecnomatix.com
13. Choi BK, Chung YC (1995) Machining simulation and NC-code verification using. IE Interface 8(3):155–169
14. Evans E, Rogers, D (1997) Using Java applets and CORBA for multiuser distributed applications. IEEE Internet Comput 1:43–55
15. Microsoft Corp (2000) Microsoft SDK for Java, version4.0. http://www.microsoft. com/Java/sdk/40/
16. Netscape Corp (1999) Object-signing tools. http://developer.netscape.com/docs /manuals /signedobj/
17. Suh SH, et al. (1997) Development of optimal operational system for CNC and CAM (2). PosTech Technical Report. Pohang University of Science and Technology, Korea
18. Jerard RB, Drysdale RL (1991) Method for geometric modeling, simulation and spatial verification of NC machining programs. In: Proceedings of the IFIP, Product Modeling for Computer-Aided Design and Manufacturing, pp 39–52